

KECERDASAN BUATAN DAN PENGEMBANGAN SISTEM CERDAS

Goldie Gunadi, S.Kom., M.Kom., Ahmad Pudoli, S.Kom., M.Kom.,
Dewi Kusumaningsih, S.Kom., M.Kom., Sirwan, S.Pd., M.Pd.,
Shinta Widyaningtyas, S.TP., M.T., Julfika Mawansyah, S.Kom., MT.,
Machrus Ali, ST., MT., IPM., dan Eka Wulansari Fridayanthie, M.Kom.



Kecerdasan Buatan dan Pengembangan Sistem Cerdas

Goldie Gunadi, S.Kom., M.Kom.

Ahmad Pudoli, S.Kom., M.Kom.

Dewi Kusumaningsih, S.Kom., M.Kom.

Sirwan, S.Pd., M.Pd.

Shinta Widyaningtyas, S.TP., M.T.

Julfikar Mawansyah, S.Kom., MT.

Machrus Ali, ST., MT., IPM.

Eka Wulansari Fridayanthie, M.Kom.



Kecerdasan Buatan dan Pengembangan Sistem Cerdas

Penulis : Goldie Gunadi, S.Kom., M.Kom., Ahmad Pudoli, S.Kom., M.Kom., Dewi Kusumaningsih, S.Kom., M.Kom., Sirwan, S.Pd., M.Pd., Shinta Widyaningtyas, S.TP., M.T., Julfikar Mawansyah, S.Kom., MT., Machrus Ali, ST., MT., IPM., dan Eka Wulansari Fridayanthie, M.Kom.
ISBN : 978-634-7701-33-6 (PDF)
Penyunting Naskah : Muhammad Hidayat.L, S.Pd., M.Pd.
Tata Letak : Endah Romadhon
Desain Sampul : Novikean Keysah Sanisri

Penerbit

PT Cipta Digital Edukasi

Jl. Wijaya Ii/Wijaya Grand Centre Blok G-15, Desa/Kelurahan Pulo, Kec. Kebayoran Baru, Kota Adm. Jakarta Selatan, Provinsi DKI Jakarta, Kode Pos: 1216

E-Mail : penerbitcde@gmail.com

Website : cideka.com

© Hak cipta dilindungi oleh undang-undang

Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan pengumuman.

Dilarang mengutip atau memperbanyak sebagian atau seluruh isi buku ini tanpa izin tertulis dari penerbit. Ketentuan Pidana Sanksi Pelanggaran Pasal 2 UU Nomor 19 Tahun 2002 Tentang Hak Cipta.

1. Barang siapa dengan sengaja dan tanpa hak melakukan perbuatan sebagaimana dimaksud dalam Pasal 2 ayat (1) atau Pasal 49 ayat (1) dan ayat (2) dipidana dengan pidana penjara masing-masing paling singkat 1 (satu) bulan dan/atau denda paling sedikit Rp1.000.000,00 (satu juta rupiah), atau pidana penjara paling lama 7 (Tujuh) tahun dan/atau denda paling banyak Rp5.000.000.000,00 (lima miliar rupiah).
2. Barang siapa dengan sengaja menyerahkan, menyiarkan, memamerkan, mengedarkan atau menjual kepada umum suatu Ciptaan atau barang hasil pelanggaran Hak Cipta atau Hak Terkait sebagaimana dimaksud pada ayat (1) dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah)

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga buku ajar yang berjudul *Kecerdasan Buatan dan Pengembangan Sistem Cerdas* ini dapat disusun dengan baik. Perkembangan teknologi digital yang sangat pesat telah membawa kecerdasan buatan menjadi salah satu bidang yang paling berpengaruh dalam berbagai aspek kehidupan. Oleh karena itu, pemahaman mengenai konsep, prinsip, dan penerapan sistem cerdas menjadi semakin penting untuk mendukung kemajuan ilmu pengetahuan dan teknologi.

Buku ini disusun untuk memberikan pemahaman yang menyeluruh mengenai kecerdasan buatan, mulai dari konsep dasar, metode, hingga implementasi dalam berbagai sistem cerdas seperti Internet of Things (IoT), robotika, dan otomasi industri. Dengan penyajian yang sistematis dan mudah dipahami, diharapkan buku ini dapat menjadi referensi bagi masyarakat umum.

Jakarta, April 2026

Tim Penyusun

DAFTAR ISI

KATA PENGANTAR.....	iii
DAFTAR ISI.....	iv
Bab 1: Jangkauan dan Peran Kecerdasan Buatan	1
1.1 Definisi dan Konsep Dasar Kecerdasan Buatan	1
1.2 Teknologi dan Teknik Inti dalam Kecerdasan Buatan	4
1.3 Peran dan Dampak Kecerdasan Buatan di Berbagai Sektor	6
1.4 Tantangan dan Pertimbangan Etika dalam Kecerdasan Buatan	9
1.5 Latihan Soal.....	12
Bab 2: Representasi Pengetahuan	14
2.1 Fondasi Teori dan Konsep Dasar Representasi Pengetahuan.....	14
2.2 Ontologi dan Pemodelan Pengetahuan Berbasis Ontologi	17
2.3 <i>Knowledge Graph</i> dan Representasi Semantik	19
2.4 Ekstraksi Pengetahuan dan Teknologi NLP	21
2.5 Representasi Pengetahuan Neuro-Simbolik	23
2.6 Evaluasi dan Aplikasi Praktis Representasi Pengetahuan	25
2.7 Latihan Soal.....	27
Bab 3: Metode Pencarian dan Penalaran	29
3.1 Fondasi Teoritis Metode Pencarian dan Penalaran.....	29
3.2 Metode Pencarian dalam Sistem Informasi	34
3.3 Strategi dan Algoritma Pemecahan Masalah.....	40
3.4 Penalaran dalam Konteks Data dan <i>Knowledge</i>	48
3.5 Aplikasi Praktis dan Studi Kasus.....	57
3.6 Latihan Soal.....	60
Bab 4: Pembelajaran Mesin	62
4.1 Konsep Fundamental Pembelajaran Mesin	62
4.2 Jenis-Jenis Pembelajaran Mesin	72

4.3 Algoritma-Algoritma Pembelajaran Mesin	79
4.4 Pembelajaran Mesin Lanjutan dan Aplikasi Praktis	86
4.5 Tantangan dan Masa Depan Pembelajaran Mesin.....	89
4.6 Latihan Soal.....	92
Bab 5: Algoritma Klasifikasi dan Prediksi.....	94
5.1 Pengenalan Algoritma Klasifikasi	94
5.2 Algoritma Klasifikasi Berbasis Pohon Keputusan	96
5.3 Algoritma <i>Ensemble: Random Forest</i> dan <i>Gradient Boosting</i>	98
5.4 <i>Support Vector Machine</i> dan <i>Naive Bayes</i>	100
5.5 <i>K-Nearest Neighbors</i> dan <i>Logistic Regression</i>	104
5.6 <i>Deep Learning</i> untuk Klasifikasi.....	106
5.7 Latihan Soal.....	107
Bab 6: Jaringan Syaraf Tiruan.....	109
6.1 Pengertian dan Konsep Dasar Jaringan Syaraf Tiruan	109
6.2 Proses Pembelajaran dan Pelatihan Jaringan Syaraf Tiruan.....	112
6.3 Arsitektur Jaringan Syaraf Tiruan yang Lanjut	114
6.4 Aplikasi Jaringan Syaraf Tiruan	117
6.5 Tantangan dan Penelitian Terkini dalam Jaringan Syaraf Tiruan	120
6.6 Latihan Soal.....	123
Bab 7: Deep Learning	124
7.1 Fondasi dan Konsep Dasar <i>Deep Learning</i>	124
7.2 <i>Convolutional Neural Networks (CNN)</i>	126
7.3 <i>Recurrent Neural Networks (RNN)</i> dan <i>LSTM</i>	128
7.4 <i>Transfer Learning</i> dan <i>Pre-trained Models</i>	130
7.5 <i>Generative Models</i> dan Aplikasinya.....	131
7.6 Optimasi dan <i>Training Deep Neural Networks</i>	133
7.7 Aplikasi <i>Deep Learning</i> dalam Domain-Domain Praktis.....	134
7.8 Tantangan dan Arah Masa Depan dalam Deep Learning	136
7.9 Latihan Soal.....	137

Bab 8: Sistem Cerdas dan Multi Agen	139
8.1 Konsep Dasar Sistem Cerdas dan Arsitektur Multi Agen	139
8.2 Mekanisme Komunikasi dan Koordinasi Antar Agen.....	141
8.3 Pembelajaran Penguatan Multi Agen dan Kolaborasi	144
8.4 Aplikasi Praktis Sistem Multi Agen dalam Berbagai Domain	146
8.5 Tantangan dan Arah Masa Depan dalam Sistem Multi Agen	149
8.6 Latihan Soal.....	151
Bab 9: Integrasi Artificial Intelligence dalam Sistem Komputasi	153
9.1 Fundamentasi Arsitektur Sistem Komputasi Berbasis AI	153
9.2 Manajemen Sumber Daya dan Optimisasi Komputasi	156
9.3 Integrasi <i>Edge AI</i> dan <i>Cloud Intelligence</i>	159
9.4 Optimisasi Energi dan Keberlanjutan.....	161
9.5 Keamanan dan Privasi dalam Sistem AI Terdistribusi	164
9.6 Latihan Soal.....	166
Bab 10: Integrasi Metode Kecerdasan Buatan dalam Sistem	168
10.1 Fondasi Konseptual dan Strategi Integrasi AI	168
10.2 Metodologi dan Pendekatan Implementasi.....	172
10.3 Aplikasi AI dalam Proses Bisnis Utama.....	176
10.4 Tantangan dan Strategi Mitigasi.....	180
10.5 Studi Kasus dan Best Practices.....	184
10.6 Latihan Soal.....	188
PROFILE PENULIS.....	190
DAFTAR PUSTAKA.....	198

Bab 1: Jangkauan dan Peran Kecerdasan Buatan

1.1 Definisi dan Konsep Dasar Kecerdasan Buatan

1.1.1 Pengertian Kecerdasan Buatan

Kecerdasan Buatan (*Artificial Intelligence/AI*) adalah cabang ilmu komputer yang berfokus pada pengembangan sistem komputer yang dapat meniru perilaku dan kemampuan kognitif manusia (Todera, 2025). *AI* dirancang untuk memproses informasi, belajar dari pengalaman, mengenali pola, dan membuat keputusan dengan cara yang mirip dengan cara berpikir manusia. Istilah "*Artificial Intelligence*" pertama kali dikemukakan pada tahun 1956 oleh sekelompok peneliti dalam konferensi Dartmouth, yang menandai dimulainya era baru penelitian dalam bidang ini (Panda & Bhatia, 2018). Sejak saat itu, *AI* telah berkembang menjadi salah satu teknologi paling transformatif dalam sejarah manusia, dengan aplikasi yang tersebar luas di berbagai sektor industri dan kehidupan sehari-hari.

1.1.2 Perbedaan antara AI Umum dan AI Sempit

Dalam spektrum kecerdasan buatan, terdapat dua kategori utama yang membedakan tingkat kemampuan sistem: AI umum (*General AI*) dan AI sempit (*Narrow AI*) (Dameruppula, 2025). AI sempit merujuk pada sistem yang dirancang untuk menyelesaikan

tugas-tugas spesifik dengan tingkat keunggulan yang dapat melampaui kemampuan manusia dalam domain tersebut, namun tidak memiliki kemampuan untuk mentransfer pengetahuan ke area lain. Sebagian besar aplikasi AI yang ada saat ini termasuk dalam kategori ini, seperti sistem pengenalan wajah, mesin penerjemah otomatis, dan asisten digital. Sebaliknya, AI umum atau *Artificial General Intelligence* (AGI) adalah konsep hipotetis yang merujuk pada sistem AI yang mampu memahami dan melaksanakan berbagai tugas intelektual dengan kemampuan yang setara atau melebihi manusia, dengan sifat adaptif dan fleksibel seperti kecerdasan manusia. Meskipun penelitian mengenai AGI sedang berlangsung, realisasi praktisnya masih merupakan tantangan yang kompleks dan belum tercapai sepenuhnya (Dameruppula, 2025).

1.1.3 Fondasi Teoritis Kecerdasan Buatan

Fondasi teoritis AI dibangun atas beberapa pilar utama, termasuk logika matematika, teori probabilitas, dan teori pengambilan keputusan (Alkhudhayr, 2024). *Machine Learning* (ML), sebagai subset dari AI, merupakan pendekatan yang memungkinkan komputer untuk belajar dari data tanpa diprogram secara eksplisit untuk setiap skenario. Teori ini dikembangkan berdasarkan pemahaman tentang bagaimana sistem dapat mengidentifikasi pola dan hubungan dalam data besar. *Deep Learning* (DL), sebagai bagian dari ML, menggunakan jaringan saraf buatan dengan banyak lapisan untuk mengekstrak fitur-fitur hierarki dari data mentah. Fondasi matematis AI mencakup konsep seperti optimisasi konveks, kalkulus variasi, dan aljabar linear, yang

semuanya berkontribusi pada pengembangan algoritma pembelajaran mesin yang efisien (Azizov et al., 2024).

1.1.4 Evolusi Historis Kecerdasan Buatan

Perjalanan *AI* dimulai pada abad pertengahan, ketika para filsuf mulai membayangkan mesin yang dapat meniru pemikiran manusia, tetapi era modern dimulai dengan konferensi Dartmouth pada tahun 1956 (Dameruppula, 2025). Pada dekade pertama, peneliti optimis tentang kemungkinan menciptakan mesin yang benar-benar cerdas, namun ekspektasi yang tidak realistis ini menyebabkan "*AI winter*" pada tahun 1970-an dan 1980-an, ketika pendanaan dan minat menurun drastis karena hasil yang mengecewakan. Kebangkitan kembali terjadi pada akhir 1980-an dan awal 1990-an dengan sistem *expert* dan peningkatan daya komputasi. Namun, transformasi sejati terjadi pada dekade 2010-an dengan munculnya *deep learning*, didorong oleh ketersediaan data besar, peningkatan kekuatan komputasi, dan algoritma pembelajaran yang lebih baik (Alkudhayr, 2024). Saat ini, *AI* telah menjadi bagian integral dari berbagai industri, dari kesehatan hingga keuangan, membawa perubahan fundamental dalam cara kita bekerja dan hidup.

1.2 Teknologi dan Teknik Inti dalam Kecerdasan Buatan

1.2.1 Machine Learning dan Paradigma Pembelajaran

Machine Learning (ML) adalah teknologi inti dalam *AI* yang memungkinkan sistem untuk belajar dan meningkat dari pengalaman tanpa pemrograman eksplisit untuk setiap tugas (Kumar, 2024). Dalam *ML*, terdapat tiga paradigma pembelajaran utama: pembelajaran tersurvisi (*supervised learning*), pembelajaran tanpa supervisi (*unsupervised learning*), dan pembelajaran dengan penguatan (*reinforcement learning*). Pembelajaran tersurvisi melibatkan pelatihan model dengan data berlabel, di mana algoritma belajar memetakan input ke output yang diketahui. Pembelajaran tanpa supervisi mengidentifikasi pola tersembunyi dalam data tanpa label, sementara pembelajaran dengan penguatan menggunakan sistem reward untuk melatih agen untuk membuat keputusan optimal (Dameruppula, 2025). Setiap paradigma memiliki aplikasi unik, dan pemilihan yang tepat tergantung pada sifat masalah yang akan diselesaikan dan jenis data yang tersedia.

1.2.2 Deep Learning dan Jaringan Saraf Buatan

Deep Learning (DL) merevolusi bidang *AI* dengan kemampuannya untuk secara otomatis mempelajari representasi fitur dari data mentah melalui jaringan saraf buatan berlapis dalam (Alkhudhayr, 2024). *Artificial Neural Networks (ANN)* terinspirasi oleh struktur dan fungsi otak biologis, terdiri dari neuron buatan yang saling terhubung dalam lapisan-lapisan. Setiap neuron

menerima input, melakukan perhitungan matematis, dan menghasilkan output yang diteruskan ke neuron berikutnya. Convolutional Neural Networks (CNN) khususnya dirancang untuk pemrosesan citra, menggunakan operasi konvolusi untuk mengekstrak fitur lokal secara efisien (Tejeda & Mayer, 2024). *Recurrent Neural Networks* (RNN) dan variannya seperti Long *Short-Term Memory* (LSTM) dirancang untuk memproses data sekuensial, memungkinkan sistem untuk mempertahankan informasi dari input sebelumnya (Hsieh et al., 2024). Transformers, yang diperkenalkan dalam dekade terakhir, telah menjadi arsitektur dominan dalam pemrosesan bahasa alami dan kini juga diterapkan dalam visi komputer (Azizov et al., 2024).

1.2.3 Natural Language Processing dan Pemrosesan Bahasa Alami

Natural Language Processing (NLP) adalah subfield *AI* yang berfokus pada kemampuan mesin untuk memahami, menafsirkan, dan menghasilkan bahasa manusia dengan cara yang bermakna dan berguna (Dameruppula, 2025). Tugas-tugas fundamental dalam *NLP* mencakup analisis sentimen, penerjemahan mesin, pengenalan entitas bernama, dan pemahaman pertanyaan. Teknologi *NLP* modern menggunakan model pembelajaran dalam skala besar, seperti *Transformer-based models* dan *Large Language Models (LLM)*, yang dilatih pada miliaran token teks dari berbagai sumber (Azizov et al., 2024). Aplikasi *NLP* telah merevolusi cara manusia berinteraksi dengan mesin, memungkinkan percakapan

alami dengan asisten digital, terjemahan real-time antar bahasa, dan ekstraksi informasi otomatis dari dokumen teks.

1.2.4 Computer Vision dan Penglihatan Komputer

Computer Vision adalah cabang *AI* yang mengajarkan mesin untuk menafsirkan dan memahami dunia visual melalui citra digital dan video (Alkhudhayr, 2024). Tugas-tugas utama dalam *computer vision* termasuk klasifikasi citra (*image classification*), deteksi objek (*object detection*), segmentasi semantik (*semantic segmentation*), dan analisis gerakan (*motion analysis*). Jaringan saraf dalam, terutama *CNN*, telah membuat terobosan signifikan dalam *computer vision*, mencapai akurasi superhuman dalam banyak tugas. Aplikasi praktis mencakup pengenalan wajah untuk keamanan, diagnosis medis berbantuan komputer, kendaraan otonom, dan pengawasan otomatis (Tejeda & Mayer, 2024). Integrasi *computer vision* dengan teknologi lain seperti *Augmented Reality (AR)* dan *Internet of Things (IoT)* membuka kemungkinan baru untuk interaksi manusia-mesin yang lebih kaya dan intuitif.

1.3 Peran dan Dampak Kecerdasan Buatan di Berbagai Sektor

1.3.1 Kecerdasan Buatan dalam Bidang Kesehatan

Kecerdasan buatan telah membawa transformasi signifikan dalam sektor kesehatan, mulai dari diagnosis hingga pengembangan obat dan manajemen pasien (Bahyat et al., 2025). Sistem *AI* dapat menganalisis citra medis seperti sinar-X dan MRI dengan akurasi

yang sebanding atau bahkan melampaui radiolog manusia, membantu mendeteksi penyakit kanker, penyakit jantung, dan kondisi medis lainnya pada tahap awal. *Machine learning* digunakan untuk memprediksi kemungkinan pasien mengembangkan penyakit kronis berdasarkan riwayat kesehatan dan faktor genetik mereka. *NLP* diterapkan untuk mengekstrak informasi klinis dari catatan medis elektronik, mengidentifikasi pola dalam data pasien yang dapat menginformasikan keputusan pengobatan. *AI* juga memainkan peran penting dalam penemuan obat, mempercepat identifikasi senyawa calon yang mungkin efektif melawan penyakit tertentu (Marques et al., 2024). Aplikasi-aplikasi ini tidak hanya meningkatkan hasil pasien tetapi juga mengurangi beban kerja profesional kesehatan dan menurunkan biaya perawatan kesehatan secara keseluruhan.

1.3.2 Kecerdasan Buatan dalam Sektor Keuangan dan Perbankan

Industri keuangan telah dengan cepat mengadopsi teknologi *AI* untuk meningkatkan efisiensi operasional, mengelola risiko, dan melayani pelanggan dengan lebih baik (Mienye et al., 2024). Algoritma pembelajaran mesin digunakan untuk deteksi penipuan, mengidentifikasi transaksi mencurigakan secara real-time untuk melindungi pelanggan dan institusi keuangan. Sistem *AI* membantu dalam penilaian kredit, menganalisis ribuan variabel untuk memprediksi risiko default peminjam dengan lebih akurat daripada metode tradisional. Dalam perdagangan saham, algoritma pembelajaran dalam digunakan untuk analisis sentimen

pasar dan prediksi pergerakan harga. *Natural Language Processing* diterapkan untuk analisis laporan keuangan dan analisis risiko teks. Manajemen portofolio yang dipersonalisasi menggunakan *AI* untuk memberikan rekomendasi investasi yang disesuaikan dengan profil risiko dan tujuan keuangan individual klien (Qatawneh et al., 2024).

1.3.3 Kecerdasan Buatan dalam Pendidikan dan Pembelajaran

Integrasi *AI* dalam pendidikan membawa potensi untuk personalisasi pembelajaran, meningkatkan aksesibilitas, dan meningkatkan hasil pendidikan (Sonawane, 2024). Sistem pembelajaran adaptif menggunakan *machine learning* untuk menyesuaikan kecepatan, konten, dan gaya pengajaran berdasarkan kebutuhan individu siswa, memungkinkan setiap siswa untuk belajar dengan cara yang paling efektif bagi mereka. *Natural Language Processing* memungkinkan sistem untuk memberikan umpan balik tertulis otomatis atas esai siswa, mengidentifikasi area yang perlu ditingkatkan sambil memberikan penjelasan edukatif. *Computer vision* digunakan untuk menganalisis interaksi di ruang kelas dan mendeteksi ketika siswa kesulitan, memungkinkan guru untuk intervensi yang tepat waktu. Tutor virtual bertenaga *AI* dapat menyediakan bantuan pembelajaran 24/7, membantu siswa yang mungkin tidak memiliki akses ke guru manusia yang berkualitas (Gordon et al., 2024).

1.3.4 Kecerdasan Buatan dalam Manufaktur dan Industri

Dalam konteks Industri 4.0, *AI* memainkan peran sentral dalam mengoptimalkan proses manufaktur, meningkatkan kualitas

produk, dan mengurangi biaya operasional (Chen et al., 2024). Sistem *computer vision* digunakan untuk inspeksi kualitas otomatis, mendeteksi cacat permukaan dan ketidaksempurnaan dengan presisi tinggi, jauh lebih cepat daripada inspeksi manual. Pemeliharaan prediktif menggunakan analitik data dan pembelajaran mesin untuk memprediksi kegagalan peralatan sebelum terjadi, memungkinkan penggantian proaktif dan meminimalkan waktu henti yang tidak terduga. Robot bertenaga *AI* dapat beradaptasi dengan perubahan dalam tugas produksi tanpa reprograman manual yang ekstensif, meningkatkan fleksibilitas lini produksi. *Natural Language Processing* dan teknologi *AI* lainnya meningkatkan komunikasi dan kolaborasi antara sistem manufaktur yang berbeda, mengoptimalkan aliran produksi dari bahan baku hingga produk jadi (Chen et al., 2024).

1.4 Tantangan dan Pertimbangan Etika dalam Kecerdasan Buatan

1.4.1 Tantangan Teknis dan Keterbatasan Sistem AI

Meskipun kemajuan yang signifikan, sistem *AI* masih menghadapi berbagai tantangan teknis yang penting (Todera, 2025). Salah satu tantangan utama adalah data scarcity, ketika data pelatihan yang tersedia tidak cukup untuk melatih model yang robust. Overfitting adalah masalah lain, di mana model mempelajari noise dan anomali dalam data pelatihan daripada pola umum, mengakibatkan kinerja buruk pada data baru yang belum pernah

dilihat (*generalization* yang lemah). Computational complexity atau kompleksitas komputasi menjadi hambatan signifikan, khususnya untuk *deep learning models* yang memerlukan sumber daya komputasi yang sangat besar. Interpretability atau kemampuan untuk menjelaskan keputusan yang dibuat oleh model *AI*, terutama dalam kasus *deep neural networks*, tetap menjadi tantangan yang sulit, karena model-model ini sering dianggap sebagai "*black box*" yang keputusannya tidak mudah dipahami (Samek et al., 2020).

1.4.2 Bias Algoritma dan Keadilan dalam Sistem AI

Salah satu tantangan etika paling kritis dalam *AI* adalah bias algoritma, di mana sistem *AI* dapat mereproduksi atau bahkan memperkuat bias yang ada dalam data pelatihan atau dalam desain algoritma itu sendiri (Agrawal et al., 2025). Jika data pelatihan mencerminkan bias historis atau kontemporer terhadap kelompok-kelompok tertentu, model yang dilatih pada data tersebut akan belajar dan mereplikasi bias-bias tersebut pada skala yang lebih besar. Sebagai contoh, sistem pengenalan wajah telah ditunjukkan memiliki akurasi yang jauh lebih rendah untuk wajah dengan kulit gelap, mencerminkan bias dalam data pelatihan dan desain awal sistem. Ini menjadi masalah serius ketika sistem *AI* digunakan dalam pengambilan keputusan yang mempengaruhi kehidupan manusia, seperti dalam sistem peradilan kriminal atau kelayakan kredit. Memastikan keadilan dan menghilangkan bias algoritma memerlukan usaha berkelanjutan dalam pengumpulan data yang representatif, desain algoritma yang adil, dan pengujian menyeluruh dari sistem *AI* sebelum penerapan (Todera, 2025).

1.4.3 Privasi Data dan Keamanan Sistem AI

Privasi data merupakan pertimbangan kritis dalam pengembangan dan penerapan sistem *AI*, terutama mengingat bahwa model pembelajaran mesin sering kali memerlukan data personal dalam jumlah besar untuk pelatihan (Dniz-Garca et al., 2023). Risiko ekstraksi data, di mana informasi pribadi dapat dipulihkan dari model yang telah dilatih, merupakan ancaman nyata untuk privasi individu. Serangan adversarial, di mana penyerang secara sengaja menginput data yang dirancang khusus untuk membingungkan sistem *AI* dan menyebabkan kesalahan, adalah kekhawatiran keamanan yang signifikan, terutama untuk aplikasi *AI* yang kritis (Ren et al., 2020). Untuk mengatasi tantangan ini, teknik seperti *differential privacy*, enkripsi data, dan pembelajaran terdistribusi sedang dikembangkan untuk melindungi privasi sambil tetap memungkinkan pelatihan model yang efektif.

1.4.4 Dampak Sosial dan Pertimbangan Ketenagakerjaan

Adopsi luas teknologi *AI* membawa implikasi sosial yang mendalam, termasuk potensi perpindahan tenaga kerja dalam berbagai sektor (Damioli et al., 2021). Otomasi yang didorong oleh *AI* dapat menghilangkan pekerjaan yang repetitif dan dapat diprediksi, yang mungkin mempengaruhi secara tidak proporsional pekerja dengan keterampilan rendah. Namun, sejarah inovasi teknologi menunjukkan bahwa sementara beberapa pekerjaan hilang, yang lain diciptakan, dan komposisi keahlian yang dibutuhkan berubah. Tanggung jawab masyarakat terletak pada berinvestasi dalam pelatihan kembali dan pendidikan untuk

mempersiapkan tenaga kerja untuk ekonomi berbasis *AI*. Selain itu, pertimbangan etis mengenai akuntabilitas dalam sistem *AI* menjadi semakin penting. Ketika sistem *AI* membuat keputusan yang mempengaruhi individu atau masyarakat, pertanyaan tentang siapa yang bertanggung jawab ketika hal-hal salah harus dijawab (Todera, 2025).

1.5 Latihan Soal

1. Jelaskan perbedaan mendasar antara *Artificial Intelligence* sempit (*Narrow AI*) dan *Artificial General Intelligence* (*AGI*). Berikan dua contoh konkret dari masing-masing jenis dan diskusikan mengapa realisasi *AGI* masih merupakan tantangan yang kompleks hingga saat ini.
2. Describe tiga paradigma pembelajaran utama dalam *Machine Learning* - pembelajaran tersupervisi, pembelajaran tanpa supervisi, dan pembelajaran dengan penguatan. Untuk setiap paradigma, berikan satu aplikasi praktis dan jelaskan bagaimana paradigma tersebut cocok untuk aplikasi yang Anda sebutkan.
3. *Deep Learning* telah merevolusi bidang *Artificial Intelligence*. Jelaskan bagaimana jaringan saraf dalam berbeda dari pendekatan tradisional dalam *Machine Learning*, dan mengapa ketersediaan data besar dan peningkatan kekuatan komputasi memungkinkan *Deep Learning* untuk berkembang pesat dalam dekade terakhir.

4. Analisis tantangan utama yang dihadapi sistem *AI* dalam hal interpretability atau explainability. Mengapa penting bagi sistem *AI* yang digunakan dalam aplikasi kritis (seperti diagnostik medis atau sistem peradilan) untuk dapat menjelaskan keputusan mereka? Apa konsekuensi dari kurangnya transparansi ini?
5. Diskusikan bagaimana bias algoritma dapat muncul dalam sistem *AI* dan dampak potensialnya pada keadilan dan kesetaraan dalam masyarakat. Berikan contoh spesifik dari bias dalam sistem *AI* yang ada dan sarankan strategi untuk mitigasi bias dalam pengembangan sistem *AI* di masa depan.

Bab 2: Representasi

Pengetahuan

Representasi pengetahuan merupakan salah satu aspek fundamental dalam pengembangan sistem kecerdasan buatan (*artificial intelligence*) modern. Konsep ini mengacu pada cara-cara untuk mengkodekan, menyimpan, dan mengorganisir informasi sehingga dapat diproses, diakses, dan dimanfaatkan oleh sistem komputer dengan efisien dan akurat. Dalam era digital yang terus berkembang ini, kemampuan untuk merepresentasikan pengetahuan dengan baik menjadi semakin penting mengingat volume data yang terus meningkat dan kompleksitas masalah yang perlu diselesaikan. Representasi pengetahuan tidak hanya berhubungan dengan bagaimana data disimpan dalam sistem komputer, tetapi juga mencakup cara bagaimana makna semantik dan hubungan antar konsep dapat diekspresikan dan diproses untuk mendukung penalaran otomatis dan pengambilan keputusan yang cerdas (Prathi et al., 2025).

2.1 Fondasi Teori dan Konsep Dasar Representasi Pengetahuan

Representasi pengetahuan dalam sistem kecerdasan buatan berangkat dari pemahaman fundamental tentang bagaimana

pengetahuan dapat dibuat dapat dipahami oleh mesin. Pengetahuan sendiri merupakan informasi yang telah diorganisir dan diproses sedemikian rupa sehingga dapat digunakan untuk memecahkan masalah, membuat keputusan, atau memberikan wawasan yang berguna. Sistem representasi pengetahuan dirancang untuk menangkap aspek-aspek penting dari dunia nyata dan mengekspresikannya dalam bentuk yang dapat dimanipulasi oleh program komputer (Serratore & Clunis, 2025).

2.1.1 Definisi dan Tujuan Representasi Pengetahuan

Representasi pengetahuan didefinisikan sebagai cara formal untuk mengekspresikan informasi tentang dunia, termasuk fakta, aturan, dan hubungan antar konsep dalam bentuk yang dapat dipahami dan diproses oleh mesin. Tujuan utama dari representasi pengetahuan adalah untuk memungkinkan sistem komputer melakukan penalaran (*reasoning*), pengambilan keputusan, dan pemecahan masalah dengan cara yang mirip dengan bagaimana manusia melakukan aktivitas-aktivitas tersebut. Dengan merepresentasikan pengetahuan secara eksplisit, sistem dapat mengakses pengetahuan ini kapan saja dan menerapkannya dalam konteks yang berbeda-beda (DeBellis & Neches, 2022).

2.1.2 Jenis-Jenis Representasi Pengetahuan

Terdapat berbagai jenis representasi pengetahuan yang telah dikembangkan dalam bidang kecerdasan buatan, masing-masing dengan keunggulan dan keterbatasan yang berbeda. Representasi deklaratif adalah jenis representasi yang menyatakan fakta dan informasi secara eksplisit, sedangkan representasi prosedural

menyatakan pengetahuan dalam bentuk prosedur atau algoritma untuk menyelesaikan tugas tertentu. Representasi logis menggunakan notasi logika formal seperti logika predikat untuk mengekspresikan pengetahuan, sementara representasi berbasis graf menggunakan struktur graf untuk menunjukkan hubungan antar entitas (Abase et al., 2025).

2.1.3 Integrasi dengan Teknologi Semantic Web

Teknologi *Semantic Web* telah membawa revolusi dalam cara representasi pengetahuan dilakukan dalam sistem modern. *Semantic Web* menyediakan infrastruktur standar untuk merepresentasikan, berbagi, dan mengolah pengetahuan di tingkat web. Standar-standar seperti RDF (*Resource Description Framework*), OWL (*Web Ontology Language*), dan SPARQL menjadi fondasi untuk membangun sistem representasi pengetahuan yang interoperable dan dapat diolah oleh berbagai aplikasi (Martinez & Sharma, 2025).

2.1.4 Peran Representasi Pengetahuan dalam Sistem AI Modern

Dalam sistem AI modern, representasi pengetahuan memainkan peran yang sangat penting dalam menghubungkan pembelajaran berbasis data dengan penalaran simbolik. Sistem modern sering mengintegrasikan pendekatan berbasis data (*neural networks*) dengan representasi pengetahuan tradisional untuk mencapai tingkat kecerdasan yang lebih tinggi. Integrasi ini memungkinkan sistem untuk tidak hanya belajar dari data tetapi juga menerapkan pengetahuan yang ada secara eksplisit untuk

meningkatkan akurasi dan interpretabilitas dari keputusan yang dibuat (Gadde, 2025).

2.2 Ontologi dan Pemodelan Pengetahuan Berbasis Ontologi

Ontologi merupakan salah satu pendekatan paling penting dalam representasi pengetahuan modern, terutama dalam konteks pengembangan sistem informasi yang kompleks. Ontologi secara formal didefinisikan sebagai spesifikasi eksplisit dari konseptualisasi suatu domain, yang mencakup definisi konsep-konsep (*classes*), properti-properti (*properties*), dan hubungan antar konsep dalam domain tersebut (Parsanasab et al., 2025).

2.2.1 Struktur dan Komponen Ontologi

Struktur ontologi umumnya terdiri dari beberapa komponen utama yang saling terhubung. Kelas (*classes*) merepresentasikan konsep atau kategori dalam domain, misalnya dalam domain medis, "penyakit" dan "obat" merupakan kelas yang berbeda. Properti (*properties*) menggambarkan karakteristik atau atribut dari kelas-kelas tersebut, seperti "nama penyakit", "gejala", atau "efek samping obat". Hubungan (*relationships*) menunjukkan koneksi antar kelas, misalnya hubungan "menyebabkan" antara "penyakit" dan "gejala". Individu atau *instances* adalah anggota spesifik dari kelas-kelas tersebut, misalnya penyakit influenza merupakan instance dari kelas "penyakit" (Lobo-Santos & Borrego-Daz, 2025).

2.2.2 Pengembangan dan Validasi Ontologi

Proses pengembangan ontologi melibatkan serangkaian tahap yang terstruktur untuk memastikan bahwa ontologi yang dihasilkan akurat dan relevan dengan domain yang ditargetkan. Tahap pertama adalah identifikasi ruang lingkup dan tujuan ontologi. Tahap kedua melibatkan pengumpulan dan analisis sumber daya yang ada dalam domain tersebut, termasuk literatur, standar yang ada, dan pengetahuan dari para ahli. Tahap ketiga adalah konseptualisasi, di mana konsep-konsep utama dan hubungan mereka diidentifikasi dan didokumentasikan. Tahap keempat adalah formalisasi, di mana konsep dan hubungan ini diekspresikan dalam bahasa formal seperti OWL atau RDF (Sahadevan et al., 2024).

2.2.3 Penalaran Otomatis Berbasis Ontologi

Salah satu keuntungan utama menggunakan ontologi adalah kemampuannya untuk mendukung penalaran otomatis. Sistem penalaran berbasis ontologi dapat menyimpulkan pengetahuan baru dari pengetahuan yang ada menggunakan aturan logika. Misalnya, jika ontologi mendefinisikan bahwa "semua influenza adalah penyakit menular" dan kita mengetahui "pasien X menderita influenza", sistem dapat secara otomatis menyimpulkan bahwa "pasien X menderita penyakit menular". Penalaran ini menggunakan mekanisme seperti *description logic* dan *rule-based reasoning* yang memungkinkan sistem untuk melakukan inferensi yang valid secara logis (Pan et al., 2025).

2.2.4 Aplikasi Ontologi dalam Berbagai Domain

Penggunaan ontologi telah menyebar ke berbagai domain aplikasi. Dalam domain medis, ontologi digunakan untuk merepresentasikan pengetahuan tentang penyakit, obat, dan prosedur medis, mendukung sistem diagnosis otomatis dan pengambilan keputusan klinis. Dalam domain e-commerce, ontologi membantu dalam kategori produk, atribut, dan hubungan antar produk, meningkatkan kemampuan pencarian dan rekomendasi. Dalam domain manajemen risiko, ontologi membantu memetakan risiko, penyebab, dan dampak potensial, mendukung identifikasi dan mitigasi risiko yang lebih efektif (Li et al., 2025).

2.3 Knowledge Graph dan Representasi Semantik

Knowledge graph merupakan evolusi penting dari konsep representasi pengetahuan yang menggabungkan elemen-elemen ontologi dengan data berskala besar. *Knowledge graph* adalah struktur data yang merepresentasikan entitas dan hubungan antar entitas dalam bentuk graf yang dapat diproses oleh mesin (Bonatti et al., 2019).

2.3.1 Struktur dan Konstruksi Knowledge Graph

Struktur dasar *knowledge graph* terdiri dari node dan edge. Node merepresentasikan entitas (misalnya orang, tempat, konsep), sementara edge merepresentasikan hubungan antar entitas. Setiap node dan edge dapat memiliki atribut atau properti yang memberikan informasi tambahan. Proses konstruksi *knowledge graph* dimulai

dengan ekstraksi entitas dan hubungan dari sumber data mentah, baik yang terstruktur maupun tidak terstruktur. Teknik *Named Entity Recognition* (NER) dan *Relation Extraction* (RE) digunakan untuk mengidentifikasi entitas dan hubungan dari teks. Setelah ekstraksi, entitas dan hubungan ini diintegrasikan ke dalam struktur graf, sering kali dengan penyesuaian terhadap ontologi yang sudah ada (Luo, 2025).

2.3.2 Penggalian Pengetahuan dari Data Tidak Terstruktur

Salah satu tantangan dalam pembangunan *knowledge graph* adalah menggali pengetahuan dari data tidak terstruktur seperti teks. Teknik pemrosesan bahasa alami (*natural language processing* atau NLP) memainkan peran penting dalam tugas ini. *Language models* berbasis transformer seperti BERT telah meningkatkan akurasi ekstraksi entitas dan hubungan secara signifikan. Sistem dapat sekarang tidak hanya mengidentifikasi entitas dan hubungan eksplisit dalam teks tetapi juga menyimpulkan hubungan implisit berdasarkan konteks semantik (Sure et al., 2025).

2.3.3 Penalaran pada Knowledge Graph

Penalaran pada *knowledge graph* melibatkan penggunaan teknik seperti *knowledge graph embedding* dan *graph neural networks* untuk melakukan prediksi dan inferensi. *Knowledge graph embedding* mengubah node dan edge menjadi vektor dalam ruang vektor berdimensi rendah, memungkinkan operasi matematis untuk mendiskusikan hubungan semantik. Teknik-teknik ini dapat digunakan untuk memprediksi hubungan yang belum ada (*link*

prediction) atau mengidentifikasi entitas yang hilang dalam triplet pengetahuan (Shang et al., 2025).

2.3.4 Aplikasi Knowledge Graph

Knowledge graph telah menemukan aplikasi luas dalam berbagai bidang. *Google Knowledge Graph* membantu meningkatkan relevansi hasil pencarian dengan memahami maksud pengguna secara lebih baik. Dalam domain medis, knowledge graph digunakan untuk mengintegrasikan pengetahuan tentang penyakit, obat, dan gejala, mendukung diagnosis dan perencanaan pengobatan. Dalam domain pendidikan, knowledge graph membantu memetakan hubungan antar konsep pembelajaran, mendukung pembelajaran yang dipersonalisasi dan rekomendasi kurikulum yang lebih baik (Fareedi et al., 2025).

2.4 Ekstraksi Pengetahuan dan Teknologi NLP

Ekstraksi pengetahuan dari data, terutama dari teks tidak terstruktur, merupakan proses kritis dalam membangun sistem representasi pengetahuan yang komprehensif. Teknologi pemrosesan bahasa alami (*natural language processing*) menyediakan alat dan teknik untuk melakukan ekstraksi ini secara otomatis (S et al., 2025).

2.4.1 Named Entity Recognition (NER)

Named Entity Recognition adalah tugas mengidentifikasi dan mengklasifikasikan entitas bernama dalam teks ke dalam kategori yang telah ditentukan, seperti nama orang, organisasi, lokasi, atau

istilah teknis. Teknologi NER modern menggunakan model pembelajaran mendalam yang telah dilatih pada kumpulan data besar. Model-model ini dapat mencapai akurasi tinggi bahkan pada teks dari domain baru, berkat kemampuan generalisasi dari model berbasis transformer seperti BERT. NER merupakan langkah pertama yang penting dalam proses ekstraksi pengetahuan, karena mengidentifikasi entitas yang akan menjadi node dalam *knowledge graph* (Luo, 2025).

2.4.2 Relation Extraction (RE)

Relation Extraction melibatkan identifikasi dan klasifikasi hubungan antar entitas dalam teks. Misalnya, dalam kalimat "Albert Einstein lahir di Jerman", sistem harus mengidentifikasi bahwa hubungan "lahir_di" menghubungkan entitas "Albert Einstein" dengan "Jerman". Teknik *relation extraction* modern menggunakan pendekatan pembelajaran mendalam yang dapat menangani hubungan kompleks dan konteks yang beragam. Integrasi antara NER dan RE memungkinkan sistem untuk mengekstrak triplet pengetahuan lengkap (subyek-hubungan-obyek) dari teks (Zhao et al., 2024).

2.4.3 Penggabungan dengan Large Language Models

Kemunculan *Large Language Models* (LLMs) seperti GPT dan Claude telah mengubah lanskap ekstraksi pengetahuan. LLM dapat memahami konteks dan nuansa bahasa dengan cara yang lebih baik daripada model yang lebih kecil, memungkinkan ekstraksi pengetahuan yang lebih akurat dan fleksibel. Teknik *in-context learning* memungkinkan LLM untuk melakukan ekstraksi

pengetahuan tanpa memerlukan pelatihan khusus. Pendekatan ini sangat berguna untuk domain spesifik di mana data pelatihan yang berlabel mungkin terbatas (Jiang et al., 2025).

2.4.4 Tantangan dan Solusi dalam Ekstraksi Pengetahuan

Meskipun kemajuan signifikan telah dicapai, masih terdapat tantangan dalam ekstraksi pengetahuan. Salah satu tantangan adalah menangani ambiguitas linguistik, di mana teks yang sama dapat memiliki makna yang berbeda dalam konteks yang berbeda. Tantangan lainnya adalah menangani pengetahuan implisit yang tidak secara eksplisit dinyatakan dalam teks. Solusi untuk tantangan-tantangan ini melibatkan penggunaan model yang lebih canggih, integrasi pengetahuan latar belakang, dan penggunaan teknik pembelajaran multimodal yang menggabungkan informasi dari berbagai sumber (Hassolthine et al., 2024).

2.5 Representasi Pengetahuan Neuro-Simbolik

Representasi pengetahuan neuro-simbolik merupakan pendekatan yang menggabungkan kekuatan pembelajaran berbasis data (*neural approaches*) dengan representasi pengetahuan simbolik. Pendekatan ini berusaha mengatasi keterbatasan masing-masing paradigma dengan mengintegrasikan keduanya (Serratore & Clunis, 2025).

2.5.1 Integrasi Pembelajaran Neural dengan Pengetahuan Simbolik

Sistem neuro-simbolik mengintegrasikan jaringan saraf dengan komponen-komponen simbolik yang dapat melakukan penalaran logis. Model-model seperti *Answer Set Programming* (ASP) dapat digabungkan dengan Large Language Models untuk menghasilkan representasi pengetahuan yang lebih terstruktur dan dapat diverifikasi. Integrasi ini memungkinkan sistem untuk tidak hanya belajar pola dari data tetapi juga menerapkan aturan logika yang ketat untuk memastikan konsistensi dan validitas logis (Alviano et al., 2025).

2.5.2 Embedding dan Representasi Vektor Pengetahuan

Salah satu teknik penting dalam representasi neuro-simbolik adalah *embedding*, yang mengubah konsep, entitas, dan hubungan ke dalam representasi vektor dalam ruang berdimensi rendah. Embedding ini mempertahankan informasi semantik dari konsep asli sambil memungkinkan operasi matematis yang efisien. Teknik *knowledge graph embedding* seperti TransE, DistMult, dan ComplEx telah menunjukkan keberhasilan dalam merepresentasikan pengetahuan dengan cara yang dapat diproses oleh model pembelajaran mesin (Mahure & Khobragade, 2024).

2.5.3 Penalaran Berbasis Logika pada Representasi Vektor

Salah satu inovasi menarik dalam representasi neuro-simbolik adalah pengembangan teknik untuk melakukan penalaran berbasis logika pada representasi vektor. Metode seperti *stl2vec* mengembangkan representasi vektor dari formula logika yang

mempertahankan semantik logika. Ini memungkinkan sistem untuk melakukan penalaran logis sambil memanfaatkan efisiensi operasi numerik pada vektor. Pendekatan ini membuka kemungkinan untuk mengintegrasikan temporal logic dan bentuk-bentuk penalaran kompleks lainnya ke dalam sistem pembelajaran mesin (Saveri et al., 2024).

2.5.4 Aplikasi dan Tantangan Representasi Neuro-Simbolik

Representasi neuro-simbolik telah diterapkan dalam berbagai aplikasi, dari sistem tanya-jawab hingga sistem diagnosis medis. Dalam domain medis, sistem neuro-simbolik dapat menggabungkan pembelajaran dari data elektronik rekam medis pasien dengan pengetahuan klinis yang diekspresikan secara eksplisit. Tantangan utama dalam pengembangan sistem neuro-simbolik adalah kompleksitas dalam merancang arsitektur yang seimbang antara komponen neural dan simbolik, serta kesulitan dalam melatih sistem yang menggabungkan kedua paradigma (Hamilton et al., 2022).

2.6 Evaluasi dan Aplikasi Praktis Representasi Pengetahuan

Evaluasi keefektifan representasi pengetahuan merupakan aspek penting dari pengembangan sistem kecerdasan buatan. Penilaian harus mencakup berbagai dimensi, dari akurasi representasi hingga keefektifan dalam mendukung aplikasi praktis (Bisenbaev, 2026).

2.6.1 Metrik Evaluasi untuk Representasi Pengetahuan

Evaluasi representasi pengetahuan dapat dilakukan melalui berbagai cara. Untuk *knowledge graph*, metrik seperti *F1-score* untuk ekstraksi entitas dan hubungan mengukur akurasi dari komponen-komponen dasar graf. Untuk penalaran, metrik seperti akurasi prediksi hubungan yang hilang mengukur kualitas dari mekanisme penalaran. Untuk ontologi, *competency questions* digunakan untuk memverifikasi bahwa ontologi dapat menjawab pertanyaan yang relevan dengan domain yang dimodelkan (Reddy et al., 2025).

2.6.2 Aplikasi dalam Sistem Rekomendasi dan Pencarian

Sistem pencarian dan rekomendasi merupakan salah satu aplikasi paling penting dari representasi pengetahuan. Dengan menggunakan *knowledge graph*, sistem dapat memahami hubungan semantik antar item, pengguna, dan fitur, memungkinkan rekomendasi yang lebih personal dan relevan. Pencarian berbasis pengetahuan dapat melampaui pencocokan kata kunci sederhana dan memberikan hasil yang lebih relevan dengan memahami maksud pengguna melalui entitas dan hubungan yang dikodekan dalam *knowledge graph* (Tran et al., 2025).

2.6.3 Penggunaan dalam Pengambilan Keputusan dan Diagnosis

Dalam domain medis dan bisnis, representasi pengetahuan memainkan peran krusial dalam mendukung pengambilan keputusan. Sistem diagnosis berbasis pengetahuan menggunakan ontologi medis dan *knowledge graph* untuk menyarankan diagnosis berdasarkan gejala yang diinputkan. Dalam manajemen risiko,

representasi pengetahuan tentang aset, ancaman, dan kerentanan membantu organisasi mengidentifikasi dan mengurangi risiko. Ketika representasi pengetahuan dirancang dengan baik, sistem ini dapat memberikan penjelasan untuk keputusan yang dibuat, meningkatkan kepercayaan pengguna terhadap sistem (Masa et al., 2022).

2.6.4 Perspektif Masa Depan dan Tantangan

Masa depan representasi pengetahuan akan didominasi oleh integrasi yang lebih dalam antara pembelajaran berbasis data dan pengetahuan simbolik. Tantangan utama adalah mengembangkan metode untuk mengekstrak, merepresentasikan, dan mengintegrasikan pengetahuan dengan cara yang scalable, maintainable, dan dapat dijelaskan. Isu keberlanjutan representasi pengetahuan juga penting, karena dunia yang direpresentasikan terus berubah dan pengetahuan perlu diperbarui secara dinamis. Penelitian dalam area ini terus berkembang, dengan fokus pada pengembangan teknik yang lebih efisien dan lebih mudah dipahami manusia (Tupayachi et al., 2024).

2.7 Latihan Soal

1. Jelaskan perbedaan mendasar antara representasi deklaratif dan representasi prosedural dalam konteks sistem kecerdasan buatan, dan berikan masing-masing satu contoh aplikasi praktis dari kedua jenis representasi tersebut.

2. Bagaimana ontologi dapat membantu dalam proses penalaran otomatis dalam sistem kecerdasan buatan? Jelaskan mekanisme inferensi dan berikan contoh bagaimana ontologi dapat menyimpulkan pengetahuan baru.
3. Analisis keuntungan dan keterbatasan penggunaan *knowledge graph* dibandingkan dengan basis data tradisional dalam merepresentasikan pengetahuan domain yang kompleks.
4. Jelaskan bagaimana teknik pemrosesan bahasa alami seperti *Named Entity Recognition* dan *Relation Extraction* berkontribusi pada pembangunan *knowledge graph* dari teks tidak terstruktur.
5. Diskusikan pendekatan neuro-simbolik dalam representasi pengetahuan. Bagaimana integrasi antara pembelajaran berbasis neural dan pengetahuan simbolik dapat mengatasi keterbatasan masing-masing paradigma?

Bab 3: Metode Pencarian dan Penalaran

Metode pencarian dan penalaran merupakan fondasi penting dalam upaya manusia untuk memahami, menganalisis, dan menyelesaikan berbagai masalah yang kompleks. Dalam era informasi digital saat ini, kemampuan untuk mencari, mengidentifikasi, dan menalar informasi dengan efektif telah menjadi keterampilan kritis yang diperlukan dalam berbagai disiplin ilmu, dari sains dan teknologi hingga pengambilan keputusan bisnis dan penelitian akademik. Tulisan ini akan membahas secara komprehensif tentang berbagai metode pencarian dan penalaran, prinsip-prinsip dasar yang mendasarinya, aplikasi praktisnya, serta tantangan dan peluang yang terkait dengan implementasi metode-metode tersebut.

3.1 Fondasi Teoritis Metode Pencarian dan Penalaran

3.1.1 Konsep Dasar Penalaran dan Logika

Penalaran adalah proses kognitif fundamental yang memungkinkan individu untuk membuat inferensi logis, memecahkan masalah, dan membuat keputusan berdasarkan informasi yang tersedia (Ke et al., 2025). Dalam konteks penelitian

dan analisis data, penalaran melibatkan kemampuan untuk menghubungkan berbagai informasi, mengidentifikasi pola, dan menarik kesimpulan yang dapat dipertanggungjawabkan. Logika, sebagai disiplin ilmu yang mempelajari prinsip-prinsip penalaran yang valid, memberikan kerangka kerja untuk mengevaluasi kebenaran dan kevalidan dari sebuah argumen.

Terdapat beberapa bentuk penalaran yang umum digunakan dalam penelitian dan analisis, termasuk penalaran deduktif, penalaran induktif, dan penalaran abduktif. Penalaran deduktif dimulai dari premis umum dan menarik kesimpulan spesifik, sedangkan penalaran induktif melakukan kebalikannya dengan mengobservasi kasus-kasus spesifik untuk menarik kesimpulan umum. Penalaran abduktif, juga dikenal sebagai penalaran hipotesis, melibatkan pengembangan penjelasan yang paling baik atau paling mungkin untuk satu set observasi (Arakelyan et al., 2024). Pemahaman yang mendalam tentang bentuk-bentuk penalaran ini sangat penting untuk menerapkan metode pencarian dan penalaran secara efektif.

3.1.2 Algoritma dan Struktur Data Pencarian

Algoritma pencarian adalah prosedur sistematis yang dirancang untuk menemukan solusi atau informasi spesifik dalam sebuah ruang masalah yang luas. Pemilihan algoritma pencarian yang tepat sangat bergantung pada karakteristik masalah yang dihadapi, termasuk ukuran ruang pencarian, struktur masalah, dan persyaratan efisiensi komputasional (Yao et al., 2023). Beberapa algoritma pencarian dasar yang sering digunakan mencakup

pencarian linear (*linear search*), pencarian biner (*binary search*), dan berbagai varian pencarian berdasarkan kedalaman (*depth-first search*) dan keluasan (*breadth-first search*).

Struktur data yang tepat memainkan peran penting dalam mengoptimalkan kinerja algoritma pencarian. Pohon pencarian (*search tree*), grafik, dan struktur data hierarki lainnya memungkinkan organisasi informasi yang efisien dan mempercepat proses pencarian. Dalam konteks modern, metode seperti *Monte Carlo Tree Search* (MCTS) telah menunjukkan efektivitas yang luar biasa dalam mengatasi ruang pencarian yang sangat besar dengan ketidakpastian yang tinggi (Zhou et al., 2023). Kemajuan dalam teknologi komputasi dan algoritma pembelajaran mesin telah membuka peluang baru untuk mengembangkan metode pencarian yang lebih cerdas dan adaptif.

3.1.3 Penalaran Logis dan Sistem Formal

Sistem penalaran logis formal merupakan kerangka kerja yang digunakan untuk merepresentasikan pengetahuan dan melakukan inferensi (penarikan kesimpulan) menggunakan aturan logika yang terdefinisi secara matematis. Sistem ini memastikan bahwa setiap kesimpulan yang dihasilkan mengikuti aturan yang benar secara logika. Beberapa karakteristik utama dari sistem penalaran logis formal antara lain (Harrison, 2009):

1. Menggunakan simbol formal: Pengetahuan direpresentasikan dalam bentuk simbol atau ekspresi logika.

2. Memiliki aturan sintaks yang jelas: Terdapat aturan yang menentukan bagaimana simbol dapat disusun menjadi pernyataan yang valid.
3. Memiliki semantik yang jelas: Setiap simbol dan ekspresi memiliki makna yang terdefinisi.
4. Menggunakan aturan inferensi: Terdapat aturan untuk menghasilkan kesimpulan dari premis yang ada.
5. Dapat diproses oleh komputer: Karena strukturnya formal dan terdefinisi dengan baik, sistem ini dapat diimplementasikan dalam program komputer.

Logika predikat orde pertama (*first-order predicate logic*) memungkinkan representasi pengetahuan yang lebih ekspresif dibandingkan dengan logika proposisional sederhana. Sistem-sistem seperti Prolog, yang didasarkan pada logika predikat, telah banyak digunakan dalam aplikasi kecerdasan buatan dan sistem penunjang keputusan (Li et al., 2025). Penggabungan antara penalaran logis formal dengan metode pembelajaran mesin telah menghasilkan pendekatan neuro-simbolik (*neuro-symbolic*) yang menjanjikan, menggabungkan kekuatan representasi logis yang transparan dengan kemampuan pembelajaran dari data (Subramanian et al., 2024).

Penelitian terkini menunjukkan bahwa kombinasi antara penalaran logis dan pembelajaran berbasis data menghasilkan sistem yang lebih robust dan dapat dijelaskan. Pendekatan yang mengintegrasikan simbolik dengan neural networks memungkinkan sistem untuk belajar dari data sambil mempertahankan transparansi dan interpretabilitas yang penting dalam aplikasi-aplikasi kritis

seperti diagnosis medis dan pengambilan keputusan strategis (Peng et al., 2023).

3.1.4 Penalaran Probabilistik dan Pengambilan Keputusan

Penalaran probabilistik merupakan metode penalaran yang digunakan untuk menangani ketidakpastian dalam proses pengambilan keputusan. Berbeda dengan penalaran logis yang bersifat deterministik (benar atau salah), penalaran probabilistik menggunakan konsep probabilitas untuk menyatakan tingkat kemungkinan suatu peristiwa atau hipotesis terjadi. Dalam banyak permasalahan dunia nyata, informasi yang tersedia sering kali tidak lengkap, tidak pasti, atau mengandung ambiguitas. Oleh karena itu, sistem kecerdasan buatan membutuhkan pendekatan yang mampu memperhitungkan ketidakpastian tersebut. Penalaran probabilistik memungkinkan sistem untuk membuat keputusan yang paling mungkin benar berdasarkan data yang tersedia (Vyas, 2023).

Teori probabilitas dan statistika Bayesian memungkinkan penggabungan pengetahuan sebelumnya (*prior knowledge*) dengan bukti baru untuk menghasilkan estimasi probabilitas yang lebih akurat. Model-model grafis probabilistik seperti *Bayesian Networks* dan *Markov Random Fields* telah terbukti sangat efektif dalam berbagai aplikasi, dari diagnosis penyakit hingga rekomendasi produk (Ali et al., 2023).

Pengambilan keputusan dalam kondisi ketidakpastian memerlukan pendekatan yang menggabungkan analisis probabilistik dengan evaluasi preferensi dan nilai. Teori utilitas yang diharapkan (*expected utility theory*) menyediakan kerangka kerja untuk

membandingkan berbagai pilihan dengan mempertimbangkan probabilitas hasil yang berbeda dan nilai yang mereka berikan (Kumar & Pamucar, 2025). Dengan berkembangnya metode pembelajaran mesin, sistem pengambilan keputusan otomatis dapat sekarang belajar fungsi utilitas secara langsung dari data atau umpan balik pengguna, membuka kemungkinan baru untuk pengambilan keputusan yang lebih adaptif dan optimal.

3.2 Metode Pencarian dalam Sistem Informasi

3.2.1 Pencarian Informasi (*Information Retrieval*) Tradisional

Dalam konteks sistem komputer, pencarian informasi biasanya digunakan untuk mengakses dokumen teks, artikel ilmiah, halaman web, atau basis data dokumen lainnya. Sistem ini tidak hanya mencari data secara langsung seperti pada basis data relasional, tetapi juga melakukan proses pencocokan antara kata kunci yang dimasukkan pengguna dengan isi dokumen yang tersedia. Sistem pencarian informasi tradisional telah berkembang dari metode berbasis kata kunci sederhana menjadi sistem yang canggih dengan kemampuan pemahaman semantik. Metode pencarian berbasis boolean, yang menggunakan operator logis AND, OR, dan NOT, memberikan kontrol yang presisi namun memerlukan keahlian pengguna yang tinggi. Metode pencarian berbasis ranking, seperti vector space model, memungkinkan peringkat hasil pencarian berdasarkan relevansi dengan query, memberikan fleksibilitas yang lebih besar untuk pengguna (Zhu et al., 2023).

Algoritma ranking yang lebih canggih seperti PageRank telah merevolusi cara mesin pencari mengorganisir dan mempresentasikan informasi dari web (Kleinberg, 1999). Algoritma ini memanfaatkan struktur link dalam web untuk mengidentifikasi halaman yang paling otoritatif dan relevan. Perkembangan selanjutnya melibatkan pengintegrasian sinyal-sinyal lain seperti engagement pengguna, waktu kunjungan, dan interaksi sosial untuk meningkatkan kualitas hasil pencarian. Dengan meningkatnya volume dan kompleksitas data, metode pencarian informasi terus berkembang untuk menangani tantangan-tantangan baru seperti pencarian multimodal, pencarian dalam bahasa alami, dan pencarian yang kontekstual (Upadhye, 2021).

Proses pencarian informasi dalam sistem Information Retrieval (IR) tradisional umumnya melibatkan beberapa tahapan yang dilakukan secara berurutan untuk memperoleh dokumen yang relevan dengan kebutuhan pengguna. Tahap pertama adalah pengumpulan dokumen, yaitu proses menghimpun berbagai dokumen dari sumber data yang tersedia seperti artikel, laporan, atau halaman web. Setelah dokumen dikumpulkan, dilakukan tahap preprocessing teks untuk mempersiapkan data agar lebih mudah diproses oleh sistem. Tahap ini biasanya mencakup tokenization untuk memecah teks menjadi unit kata, stopword removal untuk menghilangkan kata-kata umum yang tidak memiliki makna penting, serta stemming untuk mengubah kata ke bentuk dasarnya. Selanjutnya dilakukan pembuatan indeks, yaitu proses menyimpan kata-kata penting dari dokumen dalam struktur indeks sehingga

pencarian dapat dilakukan dengan lebih cepat dan efisien. Setelah indeks tersedia, sistem melakukan pemrosesan query, yaitu menganalisis kata kunci yang dimasukkan oleh pengguna dan mengubahnya ke dalam bentuk yang dapat dibandingkan dengan indeks dokumen. Tahap terakhir adalah proses pencarian dan perankingan dokumen, di mana sistem mencocokkan query dengan dokumen yang tersedia, kemudian menentukan dan menampilkan dokumen yang paling relevan kepada pengguna berdasarkan tingkat kesesuaiannya (Manning et al., 2008).

3.2.2 Pencarian Berbasis Semantic Web dan Ontologi

Pencarian berbasis Semantic Web dan ontologi merupakan pendekatan dalam sistem pencarian informasi yang bertujuan untuk memahami makna atau konteks dari informasi yang dicari oleh pengguna. Berbeda dengan sistem pencarian tradisional yang hanya mengandalkan pencocokan kata kunci, pendekatan ini memanfaatkan representasi pengetahuan dan hubungan antar konsep sehingga sistem mampu memahami maksud dari query yang diberikan. Dengan menggunakan ontologi yang terdefinisi dengan baik, sistem pencarian dapat memahami tidak hanya kecocokan kata tetapi juga makna konseptual yang mendasarinya. Pencarian berbasis ontologi memungkinkan pengguna untuk menemukan informasi yang relevan secara semantik, bahkan ketika informasi tersebut tidak menggunakan kata kunci yang sama persis (Ezhilarasi & Kalavathy, 2018).

Teknologi yang sering digunakan dalam implementasi *Semantic Web* antara lain *Resource Description Framework (RDF)*,

Web Ontology Language (OWL), dan SPARQL. RDF digunakan untuk merepresentasikan data dalam bentuk pernyataan berbasis subjek, predikat, dan objek, sedangkan OWL digunakan untuk mendefinisikan ontologi secara lebih kompleks. Sementara itu, SPARQL merupakan bahasa query yang digunakan untuk mengambil informasi dari basis data berbasis RDF. Dengan mengintegrasikan pengetahuan domain yang dikodekan dalam bentuk ontologi, sistem pencarian dapat memberikan hasil yang lebih relevan dan akurat. Perkembangan terbaru menunjukkan bahwa pengintegrasian ontologi dengan learning models berbasis data menghasilkan sistem pencarian yang kombinasi keunggulan transparansi logis dengan akurasi pembelajaran mesin (Venkatasubramanian, 2018).

3.2.3 Pencarian Menggunakan Large Language Models

Pencarian informasi mengalami perkembangan yang signifikan seiring dengan kemajuan teknologi kecerdasan buatan, khususnya melalui penggunaan *Large Language Models* (LLMs). LLM merupakan model pembelajaran mesin berbasis jaringan saraf yang dilatih menggunakan sejumlah besar data teks untuk memahami, menghasilkan, dan menganalisis bahasa manusia. Model ini mampu memahami konteks kalimat, hubungan antar konsep, serta makna semantik yang lebih dalam dibandingkan metode pencarian tradisional yang hanya bergantung pada pencocokan kata kunci. Beberapa contoh LLM yang banyak digunakan dalam berbagai aplikasi kecerdasan buatan antara lain GPT-4, BERT, dan LLaMA. Model-model tersebut dilatih

menggunakan teknik deep learning dan arsitektur transformer sehingga mampu memahami hubungan antar kata dalam suatu teks secara kontekstual. LLMs memiliki kemampuan untuk memahami pertanyaan dalam bahasa alami yang kompleks, menginterpretasikan konteks, dan menghasilkan respons yang informatif dan relevan (Ghane et al., 2024). Salah satu pendekatan yang populer dalam penggunaan LLM untuk pencarian informasi adalah *Retrieval-Augmented Generation* (RAG). Pada pendekatan ini, sistem terlebih dahulu melakukan proses pencarian dokumen dari basis data atau indeks informasi yang tersedia, kemudian LLM digunakan untuk menganalisis dan merangkum informasi tersebut sebelum menyajikannya kepada pengguna dalam bentuk jawaban yang lebih terstruktur. Dengan cara ini, sistem dapat menghasilkan jawaban yang lebih informatif dan kontekstual dibandingkan sistem pencarian tradisional. (Cao et al., 2025).

Meskipun LLMs menunjukkan kemampuan yang mengesankan, masalah seperti "hallucination" (menghasilkan informasi yang tidak akurat namun terdengar meyakinkan) tetap menjadi tantangan penting. Penelitian terkini fokus pada pengembangan mekanisme verifikasi dan filtrasi untuk memastikan bahwa informasi yang dihasilkan akurat dan dapat dipercaya (Belikova et al., 2025). Integrasi LLMs dengan sistem pencarian tradisional menunjukkan hasil yang menjanjikan, dengan RAG systems mencapai akurasi hingga 87.3% dalam tugas-tugas question-answering dibandingkan dengan 71.5% untuk pure generative models (Cao et al., 2025).

Meskipun memiliki banyak keunggulan, penggunaan LLM dalam sistem pencarian juga memiliki beberapa tantangan. Model LLM umumnya membutuhkan sumber daya komputasi yang besar serta data pelatihan yang sangat banyak. Selain itu, terdapat kemungkinan model menghasilkan informasi yang tidak sepenuhnya akurat jika sumber data yang digunakan tidak memadai. Oleh karena itu, integrasi antara LLM dengan sistem pencarian berbasis indeks atau basis data terstruktur menjadi penting untuk memastikan keakuratan informasi yang dihasilkan.

3.2.4 Clustering dan Kategorisasi dalam Pencarian

Clustering merupakan teknik pengelompokan data secara otomatis berdasarkan tingkat kemiripan antar dokumen. Dalam pendekatan ini, sistem akan menganalisis karakteristik dokumen, seperti kata-kata yang muncul di dalamnya, kemudian mengelompokkan dokumen yang memiliki pola atau topik yang serupa ke dalam satu kelompok. Berbeda dengan klasifikasi, clustering bersifat *unsupervised learning*, yang berarti proses pengelompokan dilakukan tanpa label kategori yang telah ditentukan sebelumnya. Teknik ini sering digunakan untuk mengidentifikasi struktur atau pola tersembunyi dalam kumpulan dokumen yang besar. Algoritma clustering seperti *K-means*, *hierarchical clustering*, dan *density-based clustering* memiliki aplikasi luas dalam organisasi koleksi dokumen, segmentasi gambar, dan analisis pola dalam data (Jain et al., 1999). Setiap algoritma memiliki kekuatan dan kelemahan yang berbeda tergantung pada struktur data dan tujuan analisis.

Text clustering merupakan teknik pengelompokan dokumen teks secara otomatis berdasarkan kesamaan isi atau topik yang terkandung di dalamnya. Dalam proses ini, dokumen terlebih dahulu direpresentasikan dalam bentuk fitur numerik, misalnya menggunakan metode *term frequency-inverse document frequency* (TF-IDF) atau representasi vektor kata (*word embeddings*). Setelah itu, algoritma clustering akan menghitung tingkat kemiripan antar dokumen untuk membentuk kelompok-kelompok dokumen yang memiliki topik serupa. Teknik seperti topic modeling menggunakan algoritma seperti *Latent Dirichlet Allocation* (LDA) untuk mengidentifikasi topik-topik tersembunyi dalam koleksi dokumen (Jordan, 2004). Dengan berkembangnya teknologi embedding dan neural networks, clustering modern dapat menangkap relasi semantik yang lebih dalam dan menghasilkan pengelompokan yang lebih bermakna dibandingkan dengan metode berbasis fitur tradisional (Subramaniam et al., 2021).

3.3 Strategi dan Algoritma Pemecahan Masalah

3.3.1 Pencarian Berbasis Pohon dan Graph

Dalam pengembangan sistem kecerdasan buatan dan pencarian informasi, struktur data memainkan peran penting dalam menentukan bagaimana informasi disimpan, diorganisasi, dan ditemukan kembali. Dua struktur data yang sering digunakan dalam proses pencarian adalah struktur pohon (*tree*) dan graf (*graph*). Kedua struktur ini memungkinkan representasi hubungan antar

elemen data secara lebih fleksibel dibandingkan struktur data linier, sehingga sangat cocok digunakan dalam berbagai algoritma pencarian.

Algoritma pencarian berbasis pohon seperti *Breadth-First Search* (BFS) dan *Depth-First Search* (DFS) merupakan fondasi dari banyak algoritma pencarian yang lebih kompleks. BFS memberikan jaminan untuk menemukan solusi dengan jumlah langkah minimal, sementara DFS menggunakan memori yang lebih efisien untuk masalah-masalah tertentu. Dalam permasalahan yang memiliki ruang pencarian sangat besar dan kompleks, diperlukan algoritma pencarian yang lebih canggih dan efisien untuk menemukan solusi secara optimal (Yao et al., 2023).

Alpha-Beta Pruning merupakan teknik optimisasi penting dalam pencarian adversarial, seperti pada permainan catur atau Go, yang memungkinkan penghapusan cabang-cabang pencarian yang tidak perlu tanpa mengurangi optimalitas solusi yang dihasilkan. Teknik ini bekerja dengan mengurangi jumlah node yang harus dievaluasi dalam algoritma minimax, sehingga proses pencarian menjadi lebih efisien. Sementara itu, *Beam Search* merupakan variasi dari algoritma *Breadth-First Search* (BFS) yang hanya mempertahankan sejumlah terbatas kandidat solusi terbaik pada setiap tingkat pencarian. Dengan membatasi jumlah node yang dieksplorasi, metode ini membuat proses pencarian menjadi lebih efisien secara komputasi, terutama pada permasalahan dengan ruang pencarian yang sangat besar. Kombinasi teknik-teknik tersebut dengan penggunaan heuristik yang baik memungkinkan algoritma

pencarian menemukan solusi yang memadai dalam waktu komputasi yang relatif efisien. (Jian et al., 2024).

3.3.2 Heuristik dan Optimisasi Lokal

Heuristik merupakan aturan praktis atau prosedur yang dirancang untuk meningkatkan efisiensi proses pencarian dengan mengarahkan eksplorasi menuju bagian ruang masalah yang dianggap lebih menjanjikan. Dalam banyak permasalahan kecerdasan buatan, ruang pencarian dapat menjadi sangat besar sehingga pencarian secara menyeluruh menjadi tidak efisien atau bahkan tidak mungkin dilakukan. Oleh karena itu, fungsi heuristik digunakan untuk memperkirakan kedekatan suatu keadaan (*state*) terhadap tujuan (*goal state*). Dengan memanfaatkan estimasi tersebut, algoritma pencarian dapat memprioritaskan eksplorasi pada jalur yang diperkirakan lebih dekat dengan solusi. Fungsi heuristik yang baik dapat mengurangi branching factor secara signifikan sehingga jumlah node yang harus dievaluasi menjadi lebih sedikit. Salah satu algoritma yang memanfaatkan fungsi heuristik secara efektif adalah algoritma A^* , yang menggabungkan biaya aktual untuk mencapai suatu keadaan dengan estimasi biaya untuk mencapai tujuan dari keadaan tersebut. Dengan pendekatan ini, A^* mampu menghasilkan pencarian yang optimal dan efisien selama fungsi heuristik yang digunakan bersifat *admissible*, yaitu tidak pernah melebihi-lebihkan estimasi biaya sebenarnya menuju tujuan (Zhang et al., 2024).

Pendekatan optimisasi lokal (*local optimization*) berfokus pada proses perbaikan solusi secara bertahap dengan mengevaluasi

solusi-solusi yang berada di sekitar solusi saat ini, yang disebut sebagai *neighboring solutions*. Algoritma kemudian memilih solusi tetangga yang memberikan peningkatan nilai fungsi evaluasi terbaik. Proses ini dilakukan secara iteratif hingga tidak ditemukan lagi solusi yang lebih baik atau hingga mencapai kondisi konvergensi. Pendekatan optimisasi lokal sangat efektif untuk menangani permasalahan dengan ruang solusi yang sangat besar karena tidak memerlukan eksplorasi seluruh ruang pencarian. Beberapa algoritma yang termasuk dalam kategori optimisasi lokal antara lain *Hill Climbing*, *Simulated Annealing*, dan *Genetic Algorithms*. *Hill Climbing* merupakan metode sederhana yang secara iteratif memilih solusi tetangga yang memiliki nilai evaluasi lebih baik dibandingkan solusi saat ini. Namun, metode ini memiliki kelemahan karena mudah terjebak pada kondisi *local optimum*, yaitu keadaan di mana solusi yang ditemukan lebih baik daripada solusi di sekitarnya tetapi bukan merupakan solusi terbaik secara global. Untuk mengatasi keterbatasan tersebut, metode seperti *Simulated Annealing* memperkenalkan mekanisme probabilistik yang memungkinkan sistem sesekali menerima solusi yang lebih buruk pada tahap awal pencarian. Pendekatan ini membantu sistem keluar dari jebakan *local optimum* dan meningkatkan peluang menemukan solusi yang lebih baik secara global. Sementara itu, *Genetic Algorithms* menggunakan prinsip evolusi biologis seperti seleksi, mutasi, dan rekombinasi untuk menghasilkan populasi solusi yang semakin baik dari generasi ke generasi.

Penelitian terbaru menunjukkan bahwa kombinasi beberapa metode local search, seperti penggunaan *multiple local search methods* dan *variable neighborhood search*, dapat meningkatkan kualitas solusi yang dihasilkan dibandingkan dengan penggunaan satu metode saja. Pendekatan ini memungkinkan algoritma mengeksplorasi berbagai struktur lingkungan solusi sehingga peluang menemukan solusi yang lebih optimal menjadi lebih besar (Manso et al., 2024).

3.3.3 Metode Monte Carlo Tree Search dan Aplikasinya

Monte Carlo Tree Search (MCTS) merupakan metode pencarian berbasis simulasi yang digunakan untuk mengeksplorasi ruang keputusan yang sangat besar secara efisien. Metode ini banyak diterapkan dalam berbagai aplikasi kecerdasan buatan, terutama pada permainan strategi dan permasalahan perencanaan dengan ruang kemungkinan yang luas. Berbeda dengan metode pencarian deterministik tradisional, MCTS menggunakan pendekatan probabilistik dengan melakukan sejumlah simulasi acak (*Monte Carlo simulations*) untuk memperkirakan kualitas setiap kemungkinan langkah. Proses MCTS membangun pohon pencarian secara bertahap melalui serangkaian simulasi yang terdiri dari empat tahap utama, yaitu *selection*, *expansion*, *simulation* (atau *rollout*), dan *backpropagation*. Pada tahap *selection*, algoritma memilih jalur dari akar pohon menuju node tertentu dengan mempertimbangkan keseimbangan antara eksplorasi dan eksploitasi, kemudian pada tahap *expansion* node baru ditambahkan untuk merepresentasikan kemungkinan langkah yang belum dieksplorasi. Tahap *simulation*

menjalankan simulasi acak hingga mencapai kondisi terminal, sedangkan hasilnya digunakan pada tahap *backpropagation* untuk memperbarui nilai pada node yang dilalui selama pencarian. Salah satu strategi seleksi yang sering digunakan adalah *Upper Confidence Bound for Trees* (UCT), yang menggabungkan nilai rata-rata node dan jumlah kunjungan untuk menyeimbangkan eksplorasi dan eksploitasi. Pendekatan ini memungkinkan MCTS mengevaluasi ruang pencarian yang sangat besar secara efisien, tidak memerlukan fungsi heuristik yang kompleks, serta bersifat anytime algorithm, di mana kualitas solusi dapat meningkat seiring dengan bertambahnya jumlah simulasi yang dilakukan. MCTS telah menunjukkan keberhasilan yang signifikan dalam berbagai aplikasi kecerdasan buatan. Salah satu contoh paling terkenal adalah penggunaannya dalam sistem permainan papan seperti catur, Shogi, dan terutama permainan Go. Sistem kecerdasan buatan modern seperti AlphaGo memanfaatkan kombinasi MCTS dengan jaringan saraf tiruan untuk mengevaluasi posisi permainan dan memilih langkah terbaik. Pendekatan ini memungkinkan sistem tersebut mengalahkan pemain manusia profesional dalam permainan yang sebelumnya dianggap sangat sulit bagi komputer (Horcas et al., 2021).

Aplikasi MCTS telah meluas ke berbagai domain lain termasuk perencanaan robot (*robot planning*), pengambilan keputusan dalam sistem otonom, optimisasi strategi dalam permainan video, serta simulasi sistem kompleks. Dalam robotika, misalnya, MCTS dapat digunakan untuk merencanakan rangkaian tindakan yang memungkinkan robot mencapai tujuan tertentu sambil

mempertimbangkan berbagai kemungkinan kondisi lingkungan (Jian et al., 2024). Varian dari MCTS seperti Upper Confidence Bound for Trees (UCT) memungkinkan trade-off yang lebih baik antara eksplorasi dan eksploitasi dalam pencarian. Kombinasi MCTS dengan deep neural networks, seperti dalam AlphaGo dan sistem-sistem kemudian, telah menunjukkan potensi yang luar biasa untuk mengatasi masalah-masalah yang sangat kompleks dengan cara yang efisien (Ding et al., 2025).

3.3.4 Dekomposisi Masalah dan Perencanaan Hierarki

Untuk permasalahan yang sangat kompleks, strategi yang sering digunakan dalam kecerdasan buatan adalah dekomposisi masalah, yaitu memecah suatu masalah besar menjadi beberapa submasalah yang lebih kecil dan dapat diselesaikan secara lebih mudah. Dengan pendekatan ini, setiap submasalah dapat dianalisis dan diselesaikan secara relatif independen sebelum digabungkan kembali untuk membentuk solusi dari masalah utama. Dekomposisi masalah membantu mengurangi kompleksitas ruang pencarian serta memungkinkan sistem untuk memfokuskan proses komputasi pada bagian-bagian masalah yang lebih spesifik. Selain itu, pendekatan ini juga mendukung prinsip divide and conquer, di mana permasalahan kompleks dipecah menjadi komponen-komponen yang lebih sederhana sehingga proses pemecahan masalah menjadi lebih terstruktur dan efisien.

Salah satu pendekatan yang sering digunakan dalam dekomposisi masalah adalah perencanaan hierarki (*hierarchical planning*). Dalam pendekatan ini, masalah diorganisasikan ke dalam

beberapa tingkat abstraksi, di mana tingkat yang lebih tinggi menangani keputusan yang bersifat strategis, sedangkan tingkat yang lebih rendah berfokus pada detail implementasi atau tindakan operasional. Dengan adanya struktur hierarki, sistem dapat terlebih dahulu menentukan tujuan atau strategi utama sebelum menguraikannya menjadi langkah-langkah yang lebih rinci. Pendekatan ini tidak hanya membantu mengurangi kompleksitas pencarian solusi, tetapi juga membuat proses pengambilan keputusan menjadi lebih sistematis, intuitif, dan mudah dijelaskan (Cook et al., 2023).

Dalam bidang kecerdasan buatan, perencanaan hierarki juga sering diimplementasikan dalam bentuk Hierarchical Task Network (HTN), di mana suatu tugas kompleks direpresentasikan sebagai jaringan tugas yang terdiri dari beberapa subtugas. Tugas-tugas tersebut kemudian diuraikan secara bertahap hingga mencapai tindakan dasar yang dapat dieksekusi secara langsung oleh sistem. Dengan cara ini, proses perencanaan dapat dilakukan secara lebih efisien karena sistem tidak perlu mengeksplorasi seluruh ruang pencarian secara sekaligus, melainkan secara bertahap berdasarkan struktur tugas yang telah ditentukan (Zhang & Lyu, 2024).

Selain dalam sistem perencanaan otomatis, konsep dekomposisi masalah dan perencanaan hierarki juga banyak digunakan dalam pengambilan keputusan multikriteria dan multistakeholder. Dalam konteks organisasi atau sistem yang melibatkan banyak pihak, keputusan sering kali harus mempertimbangkan berbagai faktor dan kepentingan yang berbeda.

Salah satu kerangka kerja yang sering digunakan dalam pendekatan ini adalah *Analytic Hierarchy Process* (AHP) serta berbagai metode *Multi-Criteria Decision Analysis* (MCDA). Metode-metode tersebut memungkinkan pengambil keputusan untuk menguraikan suatu masalah kompleks menjadi beberapa kriteria, subkriteria, dan alternatif solusi. Dengan struktur hierarki tersebut, organisasi dapat mengevaluasi setiap alternatif secara sistematis serta memastikan bahwa keputusan yang diambil pada level operasional tetap konsisten dengan tujuan strategis pada level yang lebih tinggi. Penelitian menunjukkan bahwa kombinasi antara dekomposisi masalah dan metode pengambilan keputusan multikriteria dapat membantu organisasi menangani permasalahan kompleks, seperti manajemen sumber daya, perencanaan proyek, dan pengelolaan kebijakan publik secara lebih efektif (Tian et al., 2019).

3.4 Penalaran dalam Konteks Data dan Knowledge

3.4.1 Pembelajaran Mesin dan Penalaran Induktif

Pembelajaran mesin (*machine learning*) merupakan salah satu cabang utama dalam kecerdasan buatan yang berfokus pada pengembangan algoritma yang mampu mempelajari pola dari data secara otomatis. Melalui proses ini, sistem dapat menggunakan pola yang dipelajari untuk membuat prediksi, melakukan klasifikasi, atau mengambil keputusan tanpa harus diprogram secara eksplisit untuk setiap kemungkinan situasi. Secara konseptual, proses pembelajaran mesin dapat dipahami sebagai bentuk penalaran induktif, yaitu

proses menarik generalisasi atau kesimpulan umum dari sejumlah observasi spesifik yang terdapat dalam data pelatihan (training data). Dengan kata lain, sistem belajar dari contoh-contoh yang tersedia untuk membangun model yang dapat digunakan dalam memprediksi data baru (Ali et al., 2023).

Algoritma pembelajaran mesin mencakup berbagai pendekatan yang berbeda, mulai dari model statistik sederhana hingga model komputasi yang sangat kompleks. Contoh metode yang relatif sederhana antara lain regresi linear, Naive Bayes, dan *decision tree*, yang sering digunakan karena memiliki tingkat interpretabilitas yang tinggi. Di sisi lain, terdapat pula metode yang lebih kompleks seperti *support vector machines*, *ensemble methods*, dan berbagai jenis *neural networks* yang mampu menangani pola data yang lebih rumit. Setiap pendekatan memiliki kelebihan dan keterbatasan masing-masing, terutama dalam hal keseimbangan antara akurasi prediksi, fleksibilitas model, serta kemudahan interpretasi hasil.

Perkembangan pesat dalam bidang pembelajaran mesin dalam beberapa tahun terakhir banyak dipengaruhi oleh kemajuan dalam *deep learning*, yaitu pendekatan yang menggunakan jaringan saraf tiruan dengan banyak lapisan untuk mempelajari representasi data secara hierarkis. Model *deep learning* mampu mengekstraksi fitur-fitur kompleks dari data mentah tanpa memerlukan proses rekayasa fitur secara manual. Beberapa arsitektur *deep learning* yang populer antara lain *Convolutional Neural Networks* (CNN) yang banyak digunakan dalam pemrosesan citra dan *Recurrent Neural*

Networks (RNN) yang dirancang untuk memproses data sekuensial seperti teks atau sinyal waktu. Model-model tersebut telah mencapai tingkat performa yang sangat tinggi, bahkan melampaui kemampuan manusia dalam beberapa tugas tertentu seperti pengenalan objek dalam gambar atau pengenalan suara. Selain CNN dan RNN, perkembangan terbaru juga menunjukkan peningkatan penggunaan *transformer-based models* dalam berbagai aplikasi pemrosesan bahasa alami (*natural language processing*). Model-model seperti ini mampu menangani hubungan konteks yang kompleks dalam data teks serta mendukung pengembangan sistem yang lebih canggih, seperti sistem penerjemahan otomatis, sistem tanya jawab, dan model bahasa skala besar.

Meskipun demikian, keberhasilan model pembelajaran mesin yang semakin kompleks juga menimbulkan tantangan baru, terutama dalam hal interpretabilitas dan transparansi model. Banyak model *deep learning* sering dianggap sebagai “*black box*”, karena proses internal yang digunakan untuk menghasilkan prediksi sulit untuk dipahami oleh manusia. Hal ini dapat menjadi masalah dalam aplikasi yang memerlukan akuntabilitas tinggi, seperti sistem medis, sistem keuangan, atau sistem pengambilan keputusan berbasis kebijakan. Oleh karena itu, penelitian terbaru dalam bidang *explainable artificial intelligence* (XAI) berupaya mengembangkan metode yang dapat menjelaskan bagaimana suatu model menghasilkan keputusan tertentu, sehingga meningkatkan kepercayaan pengguna terhadap sistem kecerdasan buatan (Singh et al., 2022).

3.4.2 Reasoning dengan Knowledge Graphs dan Ontologi

Knowledge graphs merupakan representasi pengetahuan yang menggunakan struktur graf untuk menggambarkan hubungan antara berbagai entitas dalam suatu domain. Dalam struktur ini, entitas direpresentasikan sebagai node, sedangkan hubungan antar entitas direpresentasikan sebagai edge. Dengan pendekatan ini, pengetahuan dapat disusun dalam bentuk jaringan yang terstruktur sehingga memudahkan sistem dalam memahami hubungan antar konsep. *Knowledge graph* banyak digunakan dalam berbagai aplikasi kecerdasan buatan, seperti sistem pencarian semantik, sistem rekomendasi, serta sistem tanya jawab berbasis pengetahuan.

Salah satu keunggulan utama dari *knowledge graph* adalah kemampuannya untuk mendukung proses *reasoning* atau penalaran. *Reasoning over knowledge graphs* memungkinkan sistem melakukan inferensi untuk menemukan hubungan atau fakta baru yang tidak secara eksplisit dinyatakan dalam data awal. Melalui mekanisme penalaran ini, sistem dapat memperluas pengetahuan yang tersedia dan menghasilkan informasi tambahan yang relevan. Beberapa teknik yang sering digunakan dalam pengolahan *knowledge graph* antara lain *link prediction*, yang bertujuan memprediksi hubungan yang mungkin ada antara dua entitas, serta *entity resolution*, yang digunakan untuk mengidentifikasi dan menggabungkan entitas yang sebenarnya merujuk pada objek yang sama dalam berbagai sumber data. Teknik-teknik ini dapat membantu melengkapi struktur *knowledge graph* sekaligus

meningkatkan kualitas dan konsistensi data yang tersimpan di dalamnya (Hearst, 1992).

Ontologi merupakan model formal yang mendefinisikan konsep-konsep utama dalam suatu domain serta hubungan di antara konsep tersebut. Dengan menggunakan ontologi, sistem dapat memahami makna semantik dari suatu konsep dan hubungan antar konsep secara lebih jelas. Ontologi biasanya digunakan untuk menentukan kelas, atribut, serta relasi antar entitas dalam suatu domain pengetahuan. Dengan adanya ontologi, sistem kecerdasan buatan dapat melakukan penalaran berbasis aturan (*rule-based reasoning*) maupun penalaran berbasis deskripsi logika (*description logic reasoning*), sehingga memungkinkan sistem untuk menyimpulkan informasi baru dari fakta-fakta yang telah tersedia (Pileggi, 2024).

Integrasi antara *knowledge graphs* dan model pembelajaran mesin telah menghasilkan pendekatan yang sangat kuat dalam menggabungkan penalaran logis dengan kemampuan belajar dari data. Dalam beberapa tahun terakhir, berbagai metode knowledge graph embedding telah dikembangkan untuk merepresentasikan entitas dan relasi dalam knowledge graph ke dalam ruang vektor berdimensi rendah. Metode seperti TransE, DistMult, dan ComplEx memungkinkan struktur graf yang kompleks direpresentasikan dalam bentuk numerik sehingga dapat diproses menggunakan algoritma pembelajaran mesin. Pendekatan ini membuka peluang untuk menerapkan teknik pembelajaran statistik pada struktur pengetahuan yang sebelumnya hanya dapat diproses menggunakan

metode logika simbolik secara murni (Ezhilarasi & Kalavathy, 2018). Perkembangan terbaru dalam bidang kecerdasan buatan juga menunjukkan adanya integrasi antara *knowledge graphs* dan *Large Language Models* (LLMs). Dalam pendekatan ini, *knowledge graph* digunakan sebagai sumber pengetahuan terstruktur yang dapat meningkatkan akurasi dan konsistensi informasi yang dihasilkan oleh model bahasa. Sebaliknya, model bahasa dapat membantu mengekstraksi informasi baru dari teks tidak terstruktur untuk memperkaya *knowledge graph*. Kombinasi antara representasi pengetahuan terstruktur dan kemampuan pemahaman bahasa alami dari LLM menghasilkan sistem penalaran yang lebih kuat, transparan, dan dapat dipercaya.

3.4.3 Causal Reasoning dan Analisis Kausal

Penalaran kausal (*causal reasoning*) merupakan pendekatan dalam kecerdasan buatan yang berfokus pada pemahaman hubungan sebab–akibat antara berbagai variabel dalam suatu sistem. Berbeda dengan analisis statistik tradisional yang umumnya hanya mengidentifikasi hubungan korelasional antar variabel, penalaran kausal berusaha menjelaskan bagaimana suatu peristiwa dapat mempengaruhi terjadinya peristiwa lain. Dengan memahami hubungan sebab–akibat tersebut, sistem kecerdasan buatan dapat membuat prediksi yang lebih akurat serta mendukung proses pengambilan keputusan yang lebih tepat. Salah satu pendekatan utama dalam analisis kausal adalah *causal inference*, yaitu metode yang digunakan untuk memperkirakan efek kausal dari suatu tindakan atau intervensi terhadap variabel tertentu. Teknik ini

memungkinkan sistem untuk menjawab pertanyaan seperti “apa yang akan terjadi jika suatu tindakan dilakukan?” bahkan ketika data yang tersedia hanya berupa data observasional. Untuk memodelkan hubungan kausal secara formal, digunakan representasi seperti *Directed Acyclic Graphs* (DAGs), yaitu graf berarah tanpa siklus yang menggambarkan hubungan sebab–akibat antar variabel. Selain itu, konsep *do-calculus* digunakan untuk menganalisis efek intervensi dalam suatu model kausal, sehingga memungkinkan sistem untuk membedakan antara korelasi sederhana dan hubungan kausal yang sebenarnya (Jordan, 2004).

Selain di bidang kesehatan, penalaran kausal juga digunakan dalam berbagai bidang lain, seperti ekonomi, ilmu sosial, sistem rekomendasi, serta analisis kebijakan publik. Dalam sistem rekomendasi, misalnya, pemahaman tentang hubungan kausal dapat membantu membedakan antara preferensi pengguna yang sebenarnya dengan pola perilaku yang hanya bersifat korelasional. Hal ini memungkinkan sistem memberikan rekomendasi yang lebih relevan dan efektif.

Pemahaman tentang kausalitas sangat penting dalam berbagai aplikasi kecerdasan buatan modern. Salah satu contohnya adalah dalam bidang medis dan kesehatan, khususnya dalam pengembangan sistem personalisasi terapi. Dalam konteks ini, sistem perlu memahami bagaimana suatu pengobatan tertentu mempengaruhi kondisi pasien, sehingga keputusan medis dapat disesuaikan dengan karakteristik individu. Dengan menggunakan pendekatan *causal inference*, sistem dapat memperkirakan efek dari

suatu pengobatan atau intervensi terhadap hasil kesehatan pasien secara lebih akurat. Penelitian terbaru dalam bidang analisis kausal berfokus pada pengembangan metode yang lebih robust terhadap pelanggaran asumsi serta mampu menangani hubungan kausal yang kompleks. Pendekatan baru juga mencoba mengintegrasikan metode pembelajaran mesin dengan model kausal, sehingga memungkinkan sistem untuk mempelajari struktur kausal secara otomatis dari data dalam skala besar. Dengan kombinasi antara teknik pembelajaran mesin dan kerangka analisis kausal, sistem kecerdasan buatan diharapkan mampu menghasilkan model yang tidak hanya akurat dalam prediksi, tetapi juga mampu menjelaskan hubungan sebab-akibat yang mendasari fenomena yang diamati (Garbayo et al., 2022).

3.4.4 Ensemble Methods dan Kombinasi Reasoning Approaches

Ensemble methods merupakan pendekatan dalam pembelajaran mesin dan kecerdasan buatan yang menggabungkan beberapa model atau strategi penalaran untuk menghasilkan prediksi atau keputusan yang lebih akurat dan lebih stabil dibandingkan dengan penggunaan satu model secara tunggal. Ide utama dari pendekatan ini adalah bahwa setiap model memiliki kelebihan dan kelemahan masing-masing, sehingga dengan mengombinasikan beberapa model sekaligus, kesalahan yang dihasilkan oleh satu model dapat dikompensasi oleh model lainnya. Pendekatan ini telah banyak digunakan dalam berbagai bidang analisis data dan terbukti mampu meningkatkan performa prediksi secara signifikan. Beberapa teknik ensemble yang umum digunakan antara lain

bagging, *boosting*, dan *stacking*. *Bagging* (*bootstrap aggregating*) bekerja dengan melatih beberapa model pada subset data yang berbeda dan kemudian menggabungkan hasil prediksinya, biasanya melalui rata-rata atau voting. *Boosting*, di sisi lain, membangun model secara bertahap dengan memberikan perhatian lebih pada data yang sulit diprediksi oleh model sebelumnya. Sementara itu, *stacking* menggabungkan beberapa model dasar dengan menggunakan model lain sebagai meta-learner yang bertugas mengintegrasikan hasil prediksi dari model-model tersebut. Teknik-teknik ini telah terbukti sangat efektif dalam meningkatkan akurasi serta mengurangi risiko overfitting pada model prediktif.

Sebagai contoh, penelitian terbaru menunjukkan bahwa kombinasi beberapa LLM dengan teknik *process reward-guided tree search* dapat meningkatkan kemampuan sistem dalam menyelesaikan tugas penalaran yang kompleks, seperti pemecahan masalah geometri atau tugas logika yang memerlukan beberapa langkah penalaran. Dalam pendekatan ini, model menghasilkan beberapa kemungkinan jalur penalaran, kemudian sistem menggunakan *process reward model* untuk mengevaluasi kualitas setiap langkah dalam jalur tersebut. Dengan cara ini, sistem dapat memilih jalur penalaran yang paling menjanjikan dan menghasilkan tingkat akurasi yang lebih tinggi dibandingkan penggunaan model tunggal (Park et al., 2024).

Selain itu, penggunaan *adaptive selection mechanisms* dalam *ensemble systems* memungkinkan sistem kecerdasan buatan untuk secara dinamis memilih strategi penalaran yang paling sesuai dengan

karakteristik suatu permasalahan. Misalnya, untuk permasalahan yang bersifat deterministik, sistem dapat lebih mengandalkan penalaran logis berbasis aturan, sedangkan untuk permasalahan yang mengandung ketidakpastian, sistem dapat memanfaatkan pendekatan probabilistik atau pembelajaran statistik. Pendekatan adaptif ini memungkinkan pengembangan sistem cerdas yang lebih fleksibel dan mampu menyesuaikan strategi penyelesaian masalah sesuai dengan konteks yang dihadapi.

3.5 Aplikasi Praktis dan Studi Kasus

3.5.1 Decision Support Systems dan Aplikasinya

Sistem penunjang keputusan (*Decision Support Systems* — DSS) mengintegrasikan berbagai teknik analysis dan reasoning untuk membantu decision makers dalam membuat keputusan yang lebih baik. DSS modern menggabungkan data warehouses, advanced analytics, visualization tools, dan automated reasoning engines untuk menyediakan insights yang actionable. Dalam konteks bisnis, DSS dapat membantu dalam forecasting penjualan, optimisasi inventory, dan strategic planning (Ali et al., 2023).

Aplikasi DSS dalam healthcare telah terbukti sangat valuable, membantu clinicians dalam diagnosis dan treatment planning. Sistem seperti those menggunakan machine learning untuk analisis medical images dapat achieve diagnostic accuracy yang sebanding atau bahkan melebihi human experts dalam beberapa cases. Integrasi dari clinical knowledge dengan data-driven insights

menciptakan powerful tools yang dapat meningkatkan outcomes pasien. Tantangan yang tetap ada termasuk memastikan system transparency untuk regulatory compliance dan maintaining human oversight untuk safety-critical applications (Salari & Amini, 2025).

3.5.2 Intelligent Agents dan Autonomous Decision-Making

Intelligent agents adalah sistem yang dapat perceive lingkungannya, reason tentang state saat ini dan tujuan yang diinginkan, dan take actions untuk mencapai tujuan tersebut. Agents dapat beroperasi dalam static atau dynamic environments, dengan perfect atau partial information, dan dapat belajar dari experience mereka. Arsitektur agent yang sophisticated menggabungkan perception, reasoning, planning, dan learning dalam unified framework (Nascimento et al., 2023).

Aplikasi intelligent agents telah meluas ke berbagai domain termasuk robotics, autonomous vehicles, dan game playing. Di robotics, agents harus integrate multiple sensing modalities, melakukan real-time reasoning tentang dynamics lingkungan, dan mengadapt their strategies berdasarkan feedback dari environment. Penggunaan modern LLMs sebagai "reasoning engines" dalam agents telah membuka kemungkinan untuk agents yang lebih flexible dan capable dari handling novel situations (Zhou et al., 2024).

3.5.3 Geometric Problem Solving dan Formal Reasoning

Geometric problem solving merepresentasikan challenging domain untuk automated reasoning karena memerlukan kombinasi dari symbolic geometric knowledge, visual understanding, dan

mathematical reasoning. Sistem seperti FGeo-SSS menggunakan formal systems untuk represent geometric knowledge dan perform deductive reasoning, achieving problem-solving success rates yang significant pada large-scale benchmarks (Zhang et al., 2024). Kombinasi dari symbolic solvers dengan neural approaches yang dapat interpret natural language descriptions telah menunjukkan promising results.

Penelitian terbaru mengintegrasikan chain-of-thought reasoning dengan formal language untuk improve geometric problem solving. Dengan menggunakan supervised fine-tuning pada synthetic datasets yang dikombinasikan dengan reinforcement learning, models dapat learn untuk produce hybrid reasoning traces yang combine natural language explanation dengan formal program (Yang et al., 2025). Pendekatan ini menunjukkan improvement significant dalam accuracy sambil producing reasoning traces yang shorter dan lebih interpretable dibandingkan dengan pure natural language approaches.

3.5.4 Temporal dan Sequential Reasoning

Banyak masalah dunia nyata melibatkan sequences atau temporal dependencies yang harus di-reason about. Temporal reasoning memerlukan understanding tentang temporal relationships seperti precedence, duration, dan concurrency. Teknik-teknik seperti temporal logic dan constraint satisfaction problems telah dikembangkan untuk handle temporal reasoning (Lim et al., 2025).

Dalam konteks of modern applications seperti dialogue systems dan sequential decision making, neural models seperti

recurrent neural networks dan transformer-based models telah shown impressive capability untuk capturing temporal dependencies dalam data. Kombinasi dari temporal models dengan reasoning engines memungkinkan systems untuk perform complex temporal inference seperti dalam scheduling problems atau planning untuk sequences of actions (Li et al., 2025).

3.6 Latihan Soal

1. Jelaskan perbedaan antara penalaran deduktif, induktif, dan abduktif. Berikan contoh konkret untuk masing-masing bentuk penalaran dalam konteks pemecahan masalah praktis dalam penelitian atau industri.
2. Diskusikan keuntungan dan keterbatasan dari Monte Carlo Tree Search dibandingkan dengan algoritma pencarian tradisional seperti depth-first search dan breadth-first search. Dalam situasi apa MCTS lebih efektif?
3. Analisis bagaimana Large Language Models telah mengubah paradigma information retrieval tradisional. Apa keuntungan dan tantangan dari penggunaan LLMs dalam sistem pencarian informasi?
4. Jelaskan konsep ensemble methods dalam konteks penalaran (*reasoning*). Bagaimana kombinasi dari multiple reasoning approaches dapat menghasilkan hasil yang lebih baik dibandingkan dengan individual approaches?

5. Berikan analisis mendalam tentang peran knowledge graphs dalam reasoning systems. Bagaimana integrasi antara knowledge graphs dan learning models dapat meningkatkan kemampuan sistem untuk melakukan causal inference dan reasoning?

Bab 4: Pembelajaran Mesin

Pembelajaran mesin (*machine learning*) merupakan salah satu cabang utama dari kecerdasan buatan (*artificial intelligence*) yang memungkinkan sistem komputer untuk belajar dan meningkatkan performanya dari pengalaman tanpa harus diprogram secara eksplisit untuk setiap tugas. Sebagai teknologi yang terus berkembang, pembelajaran mesin telah menjadi fondasi dari banyak aplikasi praktis dalam kehidupan sehari-hari, mulai dari sistem rekomendasi, pengenalan wajah, diagnosis medis, hingga otomasi industri. Bab ini mengulas secara komprehensif tentang pembelajaran mesin, mencakup konsep-konsep fundamental, berbagai jenis dan algoritma pembelajaran mesin, serta aplikasinya di berbagai bidang (Gupta et al., 2024).

4.1 Konsep Fundamental Pembelajaran Mesin

Pembelajaran mesin adalah suatu pendekatan dalam kecerdasan buatan yang memfokuskan pada pengembangan algoritma dan model statistik yang dapat mempelajari pola dari data tanpa harus diprogram secara eksplisit untuk setiap skenario spesifik. Inti dari pembelajaran mesin adalah kemampuan untuk mengidentifikasi pola-pola dalam data, membuat keputusan berdasarkan pola tersebut, dan terus meningkatkan akurasi dengan menerima feedback baru.

4.1.1 Definisi dan Tujuan Pembelajaran Mesin

Pembelajaran mesin (*machine learning*) merupakan salah satu cabang penting dalam bidang kecerdasan buatan yang berfokus pada pengembangan algoritma dan model yang memungkinkan komputer untuk belajar dari data. Secara umum, pembelajaran mesin dapat didefinisikan sebagai kemampuan suatu sistem komputer untuk mempelajari pola dari data dan meningkatkan kinerjanya secara otomatis tanpa harus diprogram secara eksplisit untuk setiap tugas tertentu. Dalam prosesnya, sistem pembelajaran mesin menganalisis data masukan (input data), mengidentifikasi pola-pola tersembunyi, serta membangun model matematis yang dapat digunakan untuk membuat prediksi atau keputusan. Model yang dihasilkan kemudian digunakan untuk memproses data baru yang belum pernah ditemui sebelumnya. Dengan demikian, sistem tidak hanya mengandalkan aturan yang dibuat oleh manusia, tetapi mampu menyesuaikan diri berdasarkan pengalaman yang diperoleh dari data.

Salah satu karakteristik utama dari pembelajaran mesin adalah kemampuan untuk melakukan generalisasi, yaitu kemampuan model untuk memberikan hasil yang akurat tidak hanya pada data pelatihan (training data), tetapi juga pada data baru yang belum pernah digunakan dalam proses pelatihan. Model yang memiliki kemampuan generalisasi yang baik dianggap berhasil karena dapat diterapkan pada berbagai situasi nyata yang dinamis dan bervariasi (Alnuaimi & Al-Baldawi, 2024).

Tujuan utama dari pembelajaran mesin adalah untuk mengembangkan sistem yang mampu mempelajari hubungan atau pola yang terdapat dalam data sehingga dapat digunakan untuk berbagai tugas, seperti klasifikasi, prediksi, pengenalan pola, deteksi anomali, maupun pengambilan keputusan otomatis. Dengan memanfaatkan teknik pembelajaran mesin, komputer dapat mengolah data dalam jumlah besar secara efisien dan menemukan informasi yang mungkin sulit atau bahkan tidak mungkin ditemukan melalui analisis manual. Selain itu, pembelajaran mesin juga bertujuan untuk meningkatkan kemampuan sistem dalam beradaptasi terhadap perubahan lingkungan atau data baru. Seiring dengan bertambahnya data yang diproses, model pembelajaran mesin dapat diperbarui dan ditingkatkan sehingga performanya semakin baik dari waktu ke waktu. Hal ini menjadikan pembelajaran mesin sangat penting dalam berbagai aplikasi modern seperti sistem rekomendasi, pengenalan wajah, analisis teks, kendaraan otonom, dan sistem pendukung keputusan.

4.1.2 Komponen-Komponen Utama dalam Pembelajaran Mesin

Proses pembelajaran mesin melibatkan beberapa komponen utama yang saling berinteraksi untuk menghasilkan model yang mampu mempelajari pola dari data dan membuat prediksi secara akurat. Setiap komponen memiliki peran penting dalam memastikan bahwa proses pembelajaran berjalan dengan baik dan menghasilkan model yang efektif.

Komponen pertama adalah data pelatihan (*training data*). Data pelatihan merupakan kumpulan data yang digunakan untuk

melatih model agar dapat mengenali pola atau hubungan tertentu. Data ini biasanya terdiri dari sejumlah contoh yang mewakili permasalahan yang ingin diselesaikan. Dalam pembelajaran terawasi (*supervised learning*), data pelatihan umumnya dilengkapi dengan label atau nilai target yang menjadi acuan bagi model untuk belajar. Komponen kedua adalah fitur (*features*) atau atribut data. Fitur merupakan karakteristik atau variabel yang digunakan sebagai input bagi model pembelajaran mesin. Kualitas dan relevansi fitur sangat mempengaruhi performa model yang dihasilkan. Oleh karena itu, proses pemilihan dan pengolahan fitur (*feature selection* dan *feature engineering*) sering menjadi langkah penting dalam pengembangan sistem pembelajaran mesin. Komponen ketiga adalah model. Model dalam pembelajaran mesin merupakan representasi matematis atau statistik yang digunakan untuk menggambarkan hubungan antara data input dan output. Model ini dapat berupa berbagai bentuk, seperti model regresi, pohon keputusan, jaringan saraf tiruan, atau metode lain yang dirancang untuk menangkap pola dari data yang tersedia. Komponen keempat adalah algoritma pembelajaran (*learning algorithm*). Algoritma pembelajaran merupakan prosedur atau metode yang digunakan untuk melatih model dengan cara menyesuaikan parameter-parameter model berdasarkan data pelatihan. Algoritma ini bertugas mencari parameter terbaik sehingga model dapat menghasilkan prediksi yang akurat. Contoh algoritma pembelajaran yang umum digunakan antara lain *gradient descent*, *k-nearest neighbors*, *support vector machine*, dan berbagai algoritma optimasi lainnya. Komponen kelima adalah fungsi objektif

atau fungsi kerugian (objective function atau loss function). Fungsi ini digunakan untuk mengukur seberapa baik kinerja model dalam memprediksi data. Fungsi kerugian menghitung selisih antara hasil prediksi model dengan nilai sebenarnya pada data pelatihan. Nilai kerugian yang lebih kecil menunjukkan bahwa model memiliki performa yang lebih baik. Selama proses pelatihan, algoritma pembelajaran akan berusaha meminimalkan nilai fungsi kerugian tersebut. Komponen keenam adalah proses evaluasi (*evaluation*). Evaluasi dilakukan untuk mengukur kinerja model setelah proses pelatihan selesai. Pada tahap ini, model diuji menggunakan data yang tidak digunakan selama pelatihan, yang biasanya disebut sebagai data pengujian (*testing data*) atau data validasi (*validation data*). Evaluasi bertujuan untuk memastikan bahwa model memiliki kemampuan generalisasi yang baik dan tidak hanya bekerja secara optimal pada data pelatihan saja (Yalamanchili, 2023).

Selain komponen-komponen tersebut, dalam praktiknya sering juga digunakan proses pra-pemrosesan data (*data preprocessing*), seperti pembersihan data, normalisasi, dan transformasi data agar dapat digunakan secara optimal oleh algoritma pembelajaran mesin. Tahap ini sangat penting karena kualitas data yang digunakan akan sangat mempengaruhi hasil pembelajaran yang diperoleh.

4.1.3 Proses Pembelajaran Mesin

Proses pembelajaran mesin merupakan serangkaian tahapan yang dilakukan untuk membangun model yang mampu mempelajari pola dari data dan menghasilkan prediksi atau keputusan yang

akurat. Proses ini tidak hanya melibatkan pemilihan algoritma, tetapi juga mencakup pengelolaan data, evaluasi model, serta penerapan model dalam lingkungan nyata. Secara umum, proses pembelajaran mesin terdiri dari beberapa tahap utama yang saling berkaitan.

Tahap pertama adalah pengumpulan dan persiapan data (*data collection and preparation*). Pada tahap ini, data yang relevan dengan permasalahan yang ingin diselesaikan dikumpulkan dari berbagai sumber, seperti basis data, sensor, sistem informasi, atau sumber data lainnya. Setelah data dikumpulkan, dilakukan proses pembersihan data (*data cleaning*) untuk menghilangkan data yang tidak lengkap, duplikat, atau mengandung kesalahan. Selain itu, data juga dapat melalui proses transformasi, normalisasi, atau pengkodean agar sesuai dengan kebutuhan algoritma pembelajaran mesin. Tahap kedua adalah pemilihan dan rekayasa fitur (*feature selection dan feature engineering*). Dalam tahap ini, fitur-fitur yang paling relevan dipilih dari keseluruhan atribut yang tersedia dalam data. Tujuan dari pemilihan fitur adalah untuk meningkatkan kualitas model dengan menghilangkan atribut yang tidak penting atau redundan. Selain itu, pada tahap rekayasa fitur, fitur baru dapat dibuat dari fitur yang sudah ada untuk membantu model memahami pola dalam data dengan lebih baik. Tahap ketiga adalah pemilihan model (*model selection*). Pada tahap ini, peneliti atau pengembang menentukan algoritma pembelajaran mesin yang paling sesuai dengan jenis permasalahan yang dihadapi. Misalnya, untuk masalah klasifikasi dapat digunakan algoritma seperti *decision tree*, *support vector machine*, atau *neural network*, sedangkan untuk masalah

prediksi numerik dapat digunakan algoritma regresi. Pemilihan model biasanya mempertimbangkan karakteristik data, kompleksitas model, serta kebutuhan komputasi. Tahap keempat adalah pelatihan model (*training*). Dalam tahap ini, model dilatih menggunakan data pelatihan (*training data*). Algoritma pembelajaran akan menyesuaikan parameter-parameter model agar dapat meminimalkan kesalahan prediksi berdasarkan fungsi kerugian yang digunakan. Proses pelatihan ini memungkinkan model untuk mempelajari hubungan antara fitur input dan target output. Tahap kelima adalah evaluasi model (*evaluation*). Setelah proses pelatihan selesai, model perlu diuji menggunakan data yang tidak digunakan selama proses pelatihan, seperti data validasi atau data pengujian. Tujuan evaluasi adalah untuk mengukur seberapa baik model mampu melakukan generalisasi terhadap data baru. Beberapa metrik evaluasi yang umum digunakan antara lain akurasi, presisi, recall, F1-score untuk masalah klasifikasi, serta mean squared error atau mean absolute error untuk masalah regresi. Tahap keenam adalah penyetelan hiperparameter (*hyperparameter tuning*). Hiperparameter merupakan parameter yang mengontrol perilaku algoritma pembelajaran, tetapi tidak dipelajari secara langsung dari data. Contohnya adalah jumlah pohon dalam random forest, nilai learning rate dalam neural network, atau nilai parameter dalam support vector machine. Penyetelan hiperparameter dilakukan untuk menemukan kombinasi nilai yang memberikan performa model terbaik. Tahap ketujuh adalah pengujian akhir (*final testing*). Pada tahap ini, model yang telah dioptimalkan diuji menggunakan data

pengujian yang benar-benar terpisah dari proses pelatihan dan validasi. Tujuannya adalah untuk memperoleh gambaran yang lebih objektif mengenai performa model ketika diterapkan pada data baru. Tahap terakhir adalah implementasi atau penempatan model (*deployment*). Setelah model dianggap memiliki performa yang memadai, model dapat diintegrasikan ke dalam sistem aplikasi atau lingkungan produksi. Dalam tahap ini, model digunakan untuk memproses data nyata dan menghasilkan prediksi atau keputusan secara otomatis. Selain itu, model juga perlu dipantau secara berkala karena perubahan data di dunia nyata dapat mempengaruhi kinerja model dari waktu ke waktu (Faouzi & Colliot, 2023).

4.1.4 Metrik Evaluasi Pembelajaran Mesin

Evaluasi model merupakan tahap penting dalam pembelajaran mesin untuk mengetahui seberapa baik model mampu melakukan prediksi terhadap data baru. Untuk melakukan evaluasi tersebut, digunakan berbagai metrik yang disesuaikan dengan jenis permasalahan yang ditangani, seperti klasifikasi atau regresi. Pemilihan metrik yang tepat sangat penting karena setiap metrik memberikan perspektif yang berbeda terhadap kinerja model.

Pada masalah klasifikasi, model bertujuan untuk mengelompokkan data ke dalam kelas-kelas tertentu. Beberapa metrik yang umum digunakan untuk mengevaluasi model klasifikasi antara lain sebagai berikut: Akurasi (*accuracy*) merupakan metrik yang paling sederhana dan sering digunakan. Akurasi mengukur proporsi jumlah prediksi yang benar dibandingkan dengan total prediksi yang dilakukan oleh model. Metrik ini cocok digunakan

ketika distribusi kelas dalam data relatif seimbang. Namun, pada kasus data yang tidak seimbang (*imbalanced data*), akurasi seringkali tidak cukup representatif untuk menggambarkan kinerja model secara menyeluruh. Presisi (*precision*) mengukur proporsi prediksi positif yang benar terhadap seluruh prediksi yang diklasifikasikan sebagai positif oleh model. Presisi menjadi penting dalam situasi di mana kesalahan prediksi positif harus diminimalkan, misalnya dalam sistem deteksi spam atau deteksi penyakit. Recall (*recall*) atau sensitivitas (*sensitivity*) mengukur proporsi sampel positif yang berhasil diidentifikasi dengan benar oleh model dibandingkan dengan seluruh sampel positif yang sebenarnya ada dalam data. Recall menjadi sangat penting dalam kasus di mana kegagalan mendeteksi suatu kejadian memiliki konsekuensi serius, seperti dalam diagnosis medis atau deteksi penipuan.

F1-score merupakan rata-rata harmonis antara presisi dan recall. Metrik ini digunakan untuk memberikan keseimbangan antara kedua nilai tersebut, terutama ketika terdapat ketidakseimbangan antara kelas positif dan negatif. F1-score sering digunakan ketika peneliti ingin mempertimbangkan baik presisi maupun recall secara bersamaan. Selain itu, terdapat metrik *Area Under the Receiver Operating Characteristic Curve* (AUROC). AUROC mengukur kemampuan model dalam membedakan antara kelas positif dan negatif pada berbagai nilai ambang keputusan (*threshold*). Nilai AUROC berkisar antara 0 hingga 1, di mana nilai yang lebih tinggi menunjukkan kemampuan diskriminasi model yang lebih baik.

Pada masalah regresi, tujuan model adalah memprediksi nilai numerik atau kontinu. Oleh karena itu, metrik evaluasi yang digunakan berfokus pada pengukuran kesalahan antara nilai prediksi dan nilai sebenarnya. Salah satu metrik yang umum digunakan adalah *Mean Squared Error* (MSE), yang mengukur rata-rata kuadrat dari selisih antara nilai prediksi dan nilai aktual. Metrik ini memberikan penalti yang lebih besar terhadap kesalahan yang memiliki selisih besar karena menggunakan kuadrat dari kesalahan. Metrik lainnya adalah *Mean Absolute Error* (MAE), yang mengukur rata-rata nilai absolut dari kesalahan prediksi. MAE lebih mudah diinterpretasikan karena nilai kesalahan dinyatakan dalam satuan yang sama dengan data asli. Selain itu, terdapat koefisien determinasi (*R-squared* atau R^2) yang mengukur seberapa besar proporsi variasi dalam data target yang dapat dijelaskan oleh model. Nilai R^2 berkisar antara 0 hingga 1, di mana nilai yang lebih tinggi menunjukkan bahwa model mampu menjelaskan variasi data dengan lebih baik.

Dalam praktiknya, pemilihan metrik evaluasi harus disesuaikan dengan tujuan analisis dan karakteristik data yang digunakan. Dengan menggunakan metrik evaluasi yang tepat, pengembang dapat memperoleh gambaran yang lebih akurat mengenai performa model pembelajaran mesin serta menentukan langkah perbaikan yang diperlukan untuk meningkatkan kualitas model (Krishna & Alamir, 2021).

4.2 Jenis-Jenis Pembelajaran Mesin

Pembelajaran mesin dapat diklasifikasikan menjadi beberapa kategori utama berdasarkan cara model belajar dari data dan jenis tugas yang diselesaikan.

4.2.1 Pembelajaran Terbimbing (Supervised Learning)

Pembelajaran terbimbing (*supervised learning*) merupakan salah satu jenis pembelajaran mesin yang paling umum digunakan. Dalam pendekatan ini, model dilatih menggunakan data berlabel, yaitu data yang setiap sampelnya telah dilengkapi dengan nilai target atau label yang benar. Dengan kata lain, setiap data input memiliki pasangan output yang telah diketahui sebelumnya. Tujuan utama dari pembelajaran terbimbing adalah untuk mempelajari hubungan antara variabel input (fitur) dan variabel output (label) sehingga model dapat memprediksi label yang tepat untuk data baru yang belum pernah dilihat sebelumnya. Dalam proses pembelajaran terbimbing, algoritma akan mencoba menemukan fungsi pemetaan (*mapping function*) yang dapat menghubungkan data input dengan output yang sesuai. Setelah model dilatih menggunakan data pelatihan (*training data*), model tersebut kemudian dapat digunakan untuk melakukan prediksi pada data baru. Keberhasilan model dalam pembelajaran terbimbing sangat bergantung pada kualitas data pelatihan, relevansi fitur yang digunakan, serta pemilihan algoritma yang tepat. Secara umum, pembelajaran terbimbing dibagi menjadi dua kategori utama, yaitu klasifikasi (*classification*) dan regresi (*regression*).

Klasifikasi (classification) merupakan tugas pembelajaran mesin yang bertujuan untuk memprediksi kategori atau kelas diskrit dari suatu sampel data. Dalam klasifikasi, model menentukan kelas tertentu yang paling sesuai dengan data input yang diberikan. Contoh penerapan klasifikasi antara lain adalah menentukan apakah suatu email termasuk spam atau bukan spam, mengidentifikasi apakah suatu tumor bersifat jinak atau ganas, serta mengenali objek dalam gambar seperti kendaraan, hewan, atau manusia. Algoritma klasifikasi yang banyak digunakan antara lain *Decision Tree*, *Random Forest*, *Support Vector Machine* (SVM), *Logistic Regression*, *Naive Bayes*, dan *k-Nearest Neighbors* (KNN) (Alnuaimi & Al-Baldawi, 2024).

Regresi merupakan jenis pembelajaran terbimbing yang bertujuan untuk memprediksi nilai numerik yang bersifat kontinu. Dalam regresi, model tidak menghasilkan kategori, tetapi menghasilkan nilai prediksi berupa angka. Contoh penerapan regresi antara lain memprediksi harga rumah berdasarkan fitur-fitur seperti luas bangunan, lokasi, dan usia rumah; memperkirakan tingkat penjualan suatu produk berdasarkan data historis; atau memprediksi suhu udara berdasarkan pola cuaca sebelumnya. Beberapa algoritma regresi yang umum digunakan meliputi *Linear Regression*, *Polynomial Regression*, *Ridge Regression*, *Lasso Regression*, dan metode berbasis *ensemble* seperti *Gradient Boosting* (Gunasekara et al., 2025).

4.2.2 Pembelajaran Tanpa Bimbingan (Unsupervised Learning)

Pembelajaran tanpa bimbingan (unsupervised learning) merupakan salah satu jenis pembelajaran mesin di mana model dilatih menggunakan data yang tidak memiliki label atau nilai target. Berbeda dengan pembelajaran terbimbing, dalam pendekatan ini algoritma tidak diberikan informasi mengenai hasil yang benar sehingga model berupaya menemukan struktur, pola, atau hubungan tersembunyi dalam data secara otomatis. Algoritma dalam pembelajaran tanpa bimbingan menganalisis karakteristik data dan mengelompokkan atau merepresentasikan data berdasarkan tingkat kemiripan tertentu. Metode ini sangat bermanfaat ketika data yang tersedia tidak memiliki label atau ketika peneliti ingin melakukan eksplorasi terhadap pola baru yang belum diketahui sebelumnya. Dalam praktiknya, pembelajaran tanpa bimbingan banyak digunakan dalam berbagai bidang seperti analisis eksploratif data, segmentasi pelanggan, analisis dokumen, sistem rekomendasi, serta pengolahan citra dan suara. Secara umum, pendekatan ini mencakup beberapa teknik utama, antara lain klusterisasi (*clustering*) untuk mengelompokkan data yang memiliki kemiripan, reduksi dimensionalitas (*dimensionality reduction*) untuk menyederhanakan jumlah fitur dalam data, serta pembelajaran aturan asosiasi (*association rule learning*) untuk menemukan hubungan antarvariabel dalam suatu dataset.

Klusterisasi (*clustering*) merupakan teknik yang digunakan untuk mengelompokkan data ke dalam beberapa kelompok atau cluster berdasarkan tingkat kemiripan antar data. Data yang berada

dalam satu cluster memiliki karakteristik yang lebih mirip satu sama lain dibandingkan dengan data yang berada pada cluster yang berbeda. Klusterisasi sering digunakan dalam berbagai bidang, seperti segmentasi pelanggan dalam pemasaran, pengelompokan dokumen dalam analisis teks, dan pengelompokan gen dalam bioinformatika. Beberapa algoritma klusterisasi yang populer antara lain *K-means*, *Hierarchical Clustering*, DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*), serta *Gaussian Mixture Models* (GMM).

Reduksi dimensionalitas (*dimensionality reduction*) bertujuan untuk mengurangi jumlah fitur atau variabel dalam suatu dataset dengan tetap mempertahankan informasi penting yang terkandung di dalamnya. Reduksi dimensionalitas sangat berguna ketika data memiliki jumlah fitur yang sangat besar, yang dapat meningkatkan kompleksitas komputasi serta menyulitkan proses analisis. Dengan mengurangi dimensi data, proses pemodelan menjadi lebih efisien dan visualisasi data menjadi lebih mudah dilakukan. Beberapa algoritma reduksi dimensionalitas yang sering digunakan antara lain *Principal Component Analysis* (PCA), *t-distributed Stochastic Neighbor Embedding* (t-SNE), serta *autoencoder* yang berbasis jaringan saraf tiruan.

Pembelajaran aturan asosiasi (*association rule learning*) bertujuan untuk menemukan hubungan atau pola keterkaitan antara variabel-variabel dalam suatu dataset. Salah satu contoh penerapan yang paling terkenal adalah analisis keranjang belanja (*market basket analysis*) dalam sistem ritel, di mana algoritma digunakan

untuk menemukan pola pembelian produk yang sering muncul secara bersamaan. Contohnya, pelanggan yang membeli roti sering juga membeli mentega. Algoritma yang umum digunakan dalam pembelajaran aturan asosiasi antara lain *Apriori*, *Eclat*, dan *FP-Growth*.

Meskipun pembelajaran tanpa bimbingan tidak memerlukan data berlabel, pendekatan ini sering kali lebih kompleks untuk dievaluasi karena tidak terdapat nilai target yang jelas sebagai pembanding. Namun demikian, metode ini sangat berguna untuk eksplorasi data dan penemuan pola baru, terutama ketika data yang tersedia sangat besar dan tidak memiliki label. Oleh karena itu, pembelajaran tanpa bimbingan memainkan peran penting dalam analisis data modern dan pengembangan sistem cerdas (Adaga et al., 2023).

4.2.3 Pembelajaran Semi-Terbimbing (Semi-Supervised Learning)

Pembelajaran semi-terbimbing (*semi-supervised learning*) merupakan salah satu pendekatan dalam pembelajaran mesin yang menggabungkan karakteristik dari pembelajaran terbimbing (*supervised learning*) dan pembelajaran tanpa bimbingan (*unsupervised learning*). Dalam metode ini, model dilatih menggunakan kombinasi data yang memiliki label dan data yang tidak memiliki label. Pendekatan ini menjadi sangat berguna dalam situasi di mana jumlah data berlabel terbatas atau proses pelabelan data memerlukan biaya, waktu, dan tenaga yang besar, sementara data tanpa label tersedia dalam jumlah yang jauh lebih banyak.

Terdapat beberapa strategi yang umum digunakan dalam pembelajaran semi-terbimbing. Salah satunya adalah *self-training*, yaitu pendekatan di mana model terlebih dahulu dilatih menggunakan data berlabel. Setelah model awal terbentuk, model tersebut digunakan untuk memprediksi label pada data yang tidak memiliki label. Prediksi yang memiliki tingkat kepercayaan tinggi kemudian dianggap sebagai label baru dan ditambahkan ke dalam dataset pelatihan untuk memperbaiki model secara bertahap. Strategi lainnya adalah *co-training*, yaitu metode yang melibatkan dua atau lebih model yang dilatih menggunakan representasi fitur yang berbeda dari dataset yang sama. Setiap model kemudian membantu memberikan label pada data tanpa label untuk model lainnya. Dengan cara ini, kedua model dapat saling melengkapi dalam proses pembelajaran sehingga meningkatkan kualitas prediksi. Selain kedua metode tersebut, terdapat pula pendekatan lain seperti *graph-based methods* dan *semi-supervised support vector machines*, yang memanfaatkan hubungan antar data untuk menyebarkan informasi label dari data yang telah diketahui ke data yang belum memiliki label.

Pembelajaran semi-terbimbing banyak digunakan dalam berbagai aplikasi modern, seperti pengenalan gambar, pengolahan bahasa alami (*natural language processing*), klasifikasi dokumen, serta analisis data medis. Pendekatan ini menjadi solusi yang efektif dalam situasi di mana ketersediaan data berlabel terbatas tetapi data mentah tersedia dalam jumlah besar (Yalamanchili, 2023).

4.2.4 Pembelajaran Penguatan (Reinforcement Learning)

Pembelajaran penguatan (*reinforcement learning*) merupakan salah satu jenis pembelajaran mesin di mana suatu agen (*agent*) belajar melalui proses interaksi secara langsung dengan lingkungan (*environment*). Dalam pendekatan ini, agen melakukan serangkaian tindakan atau aksi (*actions*) berdasarkan keadaan lingkungan (*state*) yang diamati. Setelah melakukan suatu aksi, agen akan menerima umpan balik berupa reward (hadiah) atau penalti (hukuman) dari lingkungan. Tujuan utama dari pembelajaran penguatan adalah untuk menemukan strategi atau kebijakan (*policy*) yang mampu memaksimalkan *total reward* kumulatif dalam jangka panjang. Berbeda dengan pembelajaran terbimbing yang menggunakan data berlabel sebagai acuan, pembelajaran penguatan tidak menyediakan jawaban yang benar secara langsung kepada model. Sebaliknya, agen harus belajar melalui proses *trial and error*, yaitu mencoba berbagai tindakan dan mengamati konsekuensi dari tindakan tersebut. Dengan cara ini, agen secara bertahap dapat memahami tindakan mana yang memberikan hasil terbaik dalam berbagai kondisi lingkungan.

Berbagai algoritma telah dikembangkan untuk mengimplementasikan pembelajaran penguatan, seperti Q-learning, SARSA, serta metode berbasis *Deep Reinforcement Learning* yang menggabungkan pembelajaran penguatan dengan jaringan saraf tiruan. Metode-metode ini memungkinkan agen untuk belajar secara efektif bahkan dalam lingkungan yang kompleks dan memiliki ruang keadaan yang besar.

Pembelajaran penguatan telah banyak digunakan dalam berbagai bidang, seperti permainan komputer dan video game, robotika, kendaraan otonom, sistem navigasi, serta optimisasi alokasi sumber daya dalam sistem komputasi dan jaringan. Salah satu contoh keberhasilan pembelajaran penguatan adalah sistem kecerdasan buatan yang mampu mengalahkan manusia dalam permainan strategi kompleks seperti catur dan Go. Dengan kemampuannya untuk belajar melalui interaksi dan pengalaman, pembelajaran penguatan menjadi salah satu pendekatan penting dalam pengembangan sistem cerdas modern (Li, 2017).

4.3 Algoritma-Algoritma Pembelajaran Mesin

Terdapat banyak algoritma pembelajaran mesin yang dapat digunakan untuk menyelesaikan berbagai jenis masalah. Setiap algoritma memiliki kekuatan dan kelemahan yang unik.

4.3.1 Algoritma Berbasis Pohon Keputusan (Tree-Based Algorithms)

Algoritma berbasis pohon keputusan merupakan salah satu metode yang banyak digunakan dalam pembelajaran mesin karena kemampuannya dalam menghasilkan model yang mudah dipahami dan diinterpretasikan. Pendekatan ini membangun model prediksi dalam bentuk struktur pohon yang menggambarkan serangkaian keputusan berdasarkan nilai-nilai fitur dalam data. Setiap cabang pada pohon merepresentasikan suatu kondisi atau aturan keputusan yang digunakan untuk memisahkan data ke dalam kelompok

tertentu. Salah satu algoritma yang paling dasar dalam kategori ini adalah *Decision Tree* (Pohon Keputusan). Algoritma ini bekerja dengan cara memecah ruang fitur secara rekursif menjadi subruang yang semakin kecil berdasarkan nilai-nilai fitur tertentu. Struktur pohon terdiri dari node akar (*root node*) yang mewakili awal proses pemisahan data, node internal yang berisi pengujian terhadap suatu atribut atau fitur, cabang (*branches*) yang menunjukkan hasil dari pengujian tersebut, serta node daun (*leaf nodes*) yang berisi keputusan akhir atau nilai prediksi. *Decision tree* banyak digunakan dalam berbagai aplikasi karena strukturnya yang intuitif dan mudah dipahami oleh manusia.

Pengembangan dari pohon keputusan tunggal adalah metode *Random Forest*. *Random Forest* merupakan teknik ensemble learning yang menggabungkan banyak pohon keputusan untuk menghasilkan model yang lebih stabil dan akurat. Dalam metode ini, setiap pohon dilatih menggunakan subset data dan subset fitur yang dipilih secara acak. Prediksi akhir dihasilkan dengan cara menggabungkan prediksi dari semua pohon, biasanya melalui metode *voting* pada masalah klasifikasi atau *averaging* pada masalah regresi. Pendekatan ini mampu mengurangi masalah *overfitting* yang sering terjadi pada pohon keputusan tunggal karena proses agregasi dari banyak model. Selain *Random Forest*, terdapat pula metode ensemble lainnya seperti *Gradient Boosting* dan *Extreme Gradient Boosting* (XGBoost). Metode *gradient boosting* membangun model secara bertahap dengan menambahkan pohon keputusan baru yang dirancang untuk memperbaiki kesalahan dari model sebelumnya.

Setiap pohon baru berfokus pada data yang sulit diprediksi oleh model sebelumnya sehingga kinerja model secara keseluruhan dapat meningkat secara bertahap. XGBoost merupakan pengembangan dari metode *gradient boosting* yang dirancang untuk meningkatkan efisiensi komputasi, kecepatan pelatihan, serta kemampuan generalisasi model.

Algoritma berbasis pohon keputusan memiliki sejumlah keunggulan, antara lain kemampuan menangani data dengan berbagai jenis fitur, tidak memerlukan proses normalisasi data yang kompleks, serta mudah diinterpretasikan. Oleh karena itu, metode ini banyak digunakan dalam berbagai bidang seperti analisis keuangan, diagnosis medis, sistem rekomendasi, serta analisis risiko. Dalam beberapa tahun terakhir, algoritma berbasis pohon keputusan juga menjadi salah satu metode yang sangat populer dalam kompetisi pembelajaran mesin dan aplikasi industri karena kemampuannya dalam menghasilkan performa prediksi yang tinggi (Ouchra et al., 2023).

4.3.2 Algoritma Berbasis Vektor Dukungan (Support Vector Machines)

Support Vector Machine (SVM) merupakan salah satu algoritma pembelajaran mesin yang banyak digunakan untuk tugas klasifikasi maupun regresi. Algoritma ini bekerja dengan mencari *hyperplane* atau bidang pemisah optimal yang mampu memisahkan data dari dua kelas dengan margin maksimal, yaitu jarak terjauh antara *hyperplane* dengan titik data terdekat dari masing-masing kelas. Titik-titik data yang berada paling dekat dengan batas pemisah

ini disebut *support vectors*, dan titik-titik inilah yang menentukan posisi *hyperplane* secara optimal.

Pada kasus yang sederhana, SVM dapat digunakan untuk memisahkan data yang bersifat *linear separable*, yaitu data yang dapat dipisahkan secara langsung dengan garis atau bidang lurus. Namun, dalam banyak kasus nyata, data sering kali tidak dapat dipisahkan secara linear. Untuk mengatasi hal tersebut, SVM menggunakan teknik yang disebut kernel trick, yaitu fungsi yang memetakan data ke dalam ruang berdimensi lebih tinggi sehingga data yang awalnya tidak dapat dipisahkan secara linear menjadi dapat dipisahkan dengan *hyperplane*. Beberapa fungsi kernel yang umum digunakan antara lain *linear kernel*, *polynomial kernel*, *radial basis function* (RBF) kernel, dan *sigmoid kernel*.

Salah satu keunggulan utama dari SVM adalah kemampuannya untuk bekerja secara efektif pada data berdimensi tinggi, di mana jumlah fitur dalam data sangat besar. Selain itu, SVM juga dikenal memiliki kemampuan generalisasi yang baik karena berfokus pada pemilihan batas keputusan yang memaksimalkan margin pemisahan antar kelas. Algoritma ini juga relatif efisien dalam penggunaan memori karena hanya menggunakan sebagian kecil dari data pelatihan, yaitu support vectors, dalam proses prediksi. Meskipun memiliki berbagai kelebihan, SVM juga memiliki beberapa keterbatasan. Salah satu kelemahannya adalah waktu pelatihan yang relatif lama ketika diterapkan pada dataset yang sangat besar. Selain itu, pemilihan fungsi kernel dan parameter yang tepat sering kali memerlukan proses eksperimen yang cukup

kompleks. Interpretasi model SVM juga cenderung lebih sulit dibandingkan dengan metode yang lebih sederhana seperti pohon keputusan (Ghosh et al., 2019).

4.3.3 Algoritma Berbasis Jaringan Saraf (Neural Networks)

Neural Networks atau jaringan saraf merupakan model pembelajaran mesin yang terinspirasi dari struktur dan fungsi otak manusia. Jaringan ini terdiri dari unit-unit pemroses sederhana yang disebut neuron, yang saling terhubung melalui bobot (*weights*) yang dapat disesuaikan. Setiap neuron menerima input, melakukan transformasi melalui fungsi aktivasi, dan meneruskan output ke neuron berikutnya. Dalam konteks pembelajaran mesin, jaringan saraf bertujuan untuk mempelajari pola dan hubungan kompleks dalam data melalui penyesuaian bobot pada koneksi antar neuron. *Artificial Neural Network* (ANN) atau jaringan saraf buatan umumnya memiliki tiga jenis lapisan: lapisan input (*input layer*) yang menerima fitur dari data, lapisan tersembunyi (*hidden layers*) yang melakukan pemrosesan internal untuk mengekstrak representasi data, dan lapisan output (*output layer*) yang menghasilkan prediksi akhir. Selama proses pelatihan, bobot pada koneksi neuron diperbarui menggunakan algoritma *backpropagation*, yang menghitung gradien dari fungsi kerugian terhadap setiap bobot dan menyesuaikannya secara iteratif untuk meminimalkan kesalahan prediksi. *Deep Learning* merupakan subkategori dari pembelajaran mesin yang menggunakan jaringan saraf dengan banyak lapisan tersembunyi (*deep neural networks*). Arsitektur *deep learning* memungkinkan model untuk mempelajari

representasi data pada berbagai tingkat abstraksi, sehingga mampu menangani data yang sangat kompleks seperti gambar, audio, dan teks.

Beberapa arsitektur khusus jaringan saraf mencakup (Ibrahim & Khorsheed, 2025):

1. *Convolutional Neural Networks* (CNN), yang dirancang khusus untuk memproses data berbentuk grid, seperti gambar. CNN menggunakan operasi konvolusi untuk mengekstrak fitur lokal dari input, serta operasi pooling untuk mengurangi dimensi data sambil mempertahankan informasi penting. CNN banyak digunakan dalam pengenalan citra, deteksi objek, dan visi komputer.
2. *Recurrent Neural Networks* (RNN), yang dirancang untuk memproses data sekuensial atau temporal, seperti teks atau sinyal waktu. RNN memiliki koneksi umpan balik (*feedback connections*) yang memungkinkan informasi dari langkah waktu sebelumnya memengaruhi langkah waktu berikutnya. Hal ini membuat RNN efektif dalam memodelkan dependensi jangka pendek dan pola temporal dalam data sekuensial.
3. *Long Short-Term Memory* (LSTM), yang merupakan varian RNN, dirancang untuk mengatasi masalah *vanishing gradient* pada jaringan sekuensial yang panjang. LSTM menggunakan *gate units*, yaitu *input gate*, *forget gate*, dan *output gate*, untuk mengontrol aliran informasi, menentukan data mana yang perlu disimpan dalam memori dan mana yang dapat dilupakan,

sehingga memungkinkan jaringan untuk mempelajari dependensi jangka panjang secara lebih efektif.

Jaringan saraf dan variannya telah menjadi tulang punggung dari berbagai aplikasi modern, termasuk pengenalan wajah, pengolahan bahasa alami, sistem rekomendasi, kendaraan otonom, dan analisis sinyal biologis. Kemampuannya untuk menangkap pola non-linear yang kompleks menjadikan neural networks salah satu algoritma paling fleksibel dan kuat dalam pembelajaran mesin.

4.3.4 Algoritma Lainnya dan Ensemble Methods

Selain algoritma berbasis pohon, vektor dukungan, dan jaringan saraf, terdapat beberapa algoritma pembelajaran mesin lain yang juga banyak digunakan. *Logistic Regression* adalah algoritma klasifikasi linear yang memodelkan probabilitas suatu sampel termasuk dalam kelas positif menggunakan fungsi logistik (sigmoid). Algoritma ini efektif untuk masalah klasifikasi biner dan memberikan interpretasi probabilistik yang jelas. *Naive Bayes* adalah algoritma probabilistik yang didasarkan pada teorema Bayes, dengan asumsi bahwa setiap fitur bersifat independen satu sama lain. Meskipun asumsi kemandirian fitur jarang sepenuhnya terpenuhi dalam praktik, *Naive Bayes* tetap terbukti cepat dan efektif dalam berbagai aplikasi, termasuk klasifikasi teks dan analisis spam. *K-Nearest Neighbors (KNN)* adalah algoritma non-parametrik yang sederhana namun powerful, yang mengklasifikasikan sampel baru berdasarkan mayoritas kelas dari k tetangga terdekat dalam ruang fitur. KNN juga dapat digunakan untuk regresi dengan mengambil rata-rata nilai dari tetangga terdekat.

Selain algoritma tunggal, *Ensemble Methods* merupakan pendekatan yang menggabungkan prediksi dari beberapa model untuk menghasilkan prediksi yang lebih akurat dan robust terhadap variasi data. Pendekatan ini didasarkan pada prinsip bahwa kombinasi beberapa model yang lemah atau moderat dapat menghasilkan model gabungan yang lebih kuat. Metode ensemble yang populer meliputi *Bagging (Bootstrap Aggregating)*, yang membangun beberapa model secara independen pada subset data acak dan menggabungkan prediksi melalui *voting* atau *averaging*; *Boosting*, yang membangun model secara bertahap dengan memberi bobot lebih pada sampel yang sulit diprediksi; dan *Stacking*, yang menggabungkan berbagai model dasar melalui model meta untuk meningkatkan akurasi prediksi. *Ensemble Methods* telah terbukti sangat efektif dalam berbagai kompetisi *machine learning* dan aplikasi praktis, termasuk prediksi finansial, deteksi penyakit, dan analisis citra (Gaikwad et al., 2024).

4.4 Pembelajaran Mesin Lanjutan dan Aplikasi Praktis

Pembelajaran mesin telah berkembang menjadi teknologi canggih dengan berbagai aplikasi praktis di berbagai industri.

4.4.1 Deep Learning dan Jaringan Saraf Konvolusi Rekuren

Deep Learning telah merevolusi berbagai bidang dengan memberikan tingkat akurasi yang belum pernah terjadi sebelumnya. Kombinasi dari *Convolutional Neural Networks (CNN)*

dan *Recurrent Neural Networks* (RNN) telah terbukti sangat efektif untuk tugas-tugas yang melibatkan data spasial dan temporal secara bersamaan. Arsitektur hibrida CNN-RNN dapat digunakan untuk tugas-tugas seperti klasifikasi video, pengenalan kegiatan manusia, dan prediksi deret waktu multidimensional. Dalam konteks pengenalan emosi pidato (*speech emotion recognition*), jaringan CNN-RNN dengan mekanisme atansi ganda (*multiple attention mechanisms*) telah mencapai hasil yang sangat baik dengan memanfaatkan informasi waktu-frekuensi dan informasi sekuensial secara simultam (Jiang et al., 2022). *Transformers*, yang diperkenalkan dengan mekanisme *self-attention*, telah menjadi arsitektur yang dominan dalam pemrosesan bahasa alami dan telah mulai diterapkan pada tugas-tugas visi komputer dan analisis data temporal (Mienye & Swart, 2024).

4.4.2 Aplikasi dalam Bidang Kesehatan

Pembelajaran mesin telah membuat kontribusi signifikan dalam bidang kesehatan dan kedokteran. Dalam diagnostik medis, *deep learning* khususnya CNN telah digunakan untuk menganalisis citra medis seperti X-ray, CT scan, dan MRI untuk mendeteksi dan mengklasifikasi penyakit. Dalam prediksi penyakit, model pembelajaran mesin dapat dilatih untuk memprediksi risiko penyakit tertentu berdasarkan faktor-faktor medis dan demografis pasien. Misalnya, dalam prediksi hipoglikemia dini pada pasien diabetes tipe 2 menggunakan teknik pembelajaran mesin hibrida, model *gradient boosting* dan neural networks menunjukkan akurasi yang tinggi (Gaikwad et al., 2024). Dalam penemuan obat,

pembelajaran mesin dapat menganalisis data eksperimental untuk mengidentifikasi kandidat obat potensial dan memprediksi aktivitas farmakologis mereka. Dalam pemongan citra medis, *deep learning* telah memungkinkan segmentasi otomatis struktur anatomi dan lesi patologi dengan akurasi tinggi (Xu et al., 2024).

4.4.3 Aplikasi dalam Bidang Keuangan dan E-Commerce

Dalam bidang keuangan, pembelajaran mesin digunakan secara luas untuk deteksi penipuan (*fraud detection*), penilaian risiko kredit, dan prediksi harga saham. Model pembelajaran mesin dapat menganalisis pola transaksi untuk mengidentifikasi aktivitas mencurigakan yang mungkin menunjukkan penipuan. Dalam evaluasi kredit, model dapat menggunakan data historis nasabah untuk memprediksi probabilitas default. Dalam prediksi pasar, model seperti *CNN*, *LSTM*, dan *deep reinforcement learning* telah digunakan untuk memprediksi tren harga saham. Dalam e-commerce, pembelajaran mesin digunakan untuk sistem rekomendasi (*recommendation systems*) yang memprediksi produk yang mungkin disukai pelanggan berdasarkan perilaku pembelian mereka sebelumnya dan perilaku pelanggan serupa lainnya. Pembelajaran mesin juga digunakan untuk optimisasi pricing (*price optimization*), manajemen inventori, dan personalisasi pengalaman belanja (Adaga et al., 2023).

4.4.4 Aplikasi dalam Bidang Lainnya

Selain kesehatan dan keuangan, pembelajaran mesin memiliki aplikasi luas di berbagai bidang lainnya. Dalam visi komputer, pembelajaran mesin digunakan untuk pengenalan objek,

deteksi wajah, segmentasi citra, dan klasifikasi citra. Dalam pemrosesan bahasa alami, pembelajaran mesin digunakan untuk terjemahan mesin, pengenalan entitas bernama, analisis sentimen, dan sistem pertanyaan-jawaban. Dalam robotika, pembelajaran mesin digunakan untuk perencanaan gerakan, kontrol otomatis, dan pembelajaran dari demonstrasi. Dalam manufaktur, pembelajaran mesin digunakan untuk deteksi cacat otomatis, pemeliharaan prediktif, dan optimisasi proses produksi. Dalam transportasi, pembelajaran mesin digunakan untuk sistem navigasi otomatis, prediksi lalu lintas, dan deteksi pengemudi yang mengantuk. Dalam pertanian, pembelajaran mesin digunakan untuk memprediksi hasil panen, deteksi penyakit tanaman, dan optimisasi irigasi (Shah et al., 2024).

4.5 Tantangan dan Masa Depan Pembelajaran Mesin

Meskipun pembelajaran mesin telah mencapai kemajuan yang signifikan dan digunakan secara luas di berbagai bidang, masih terdapat sejumlah tantangan yang perlu diatasi agar teknologi ini dapat diterapkan secara aman, efektif, dan etis.

4.5.1 Tantangan Teknis dan Metodologis

Salah satu tantangan utama dalam pembelajaran mesin adalah kualitas dan kuantitas data. Model membutuhkan data pelatihan yang cukup besar, representatif, dan bersih agar mampu belajar pola yang valid. Data yang tidak lengkap, tidak seimbang

(*imbalanced*), atau mengandung noise dapat menurunkan performa model secara signifikan. Tantangan berikutnya adalah *overfitting* dan *underfitting*. *Overfitting* terjadi ketika model terlalu menyesuaikan diri dengan data pelatihan, termasuk noise atau outlier, sehingga gagal melakukan generalisasi pada data baru. Sebaliknya, *underfitting* muncul ketika model terlalu sederhana untuk menangkap pola kompleks dalam data, sehingga prediksi menjadi tidak akurat. Kompleksitas komputasi juga menjadi kendala, terutama pada model *deep learning*. Proses pelatihan jaringan saraf yang besar membutuhkan daya komputasi tinggi, memori besar, dan waktu yang lama, yang sering memerlukan infrastruktur hardware mahal seperti GPU dan TPU. Selain itu, interpretabilitas model menjadi masalah penting. Banyak model, terutama deep learning, bersifat “*black box*”, sehingga sulit untuk memahami bagaimana model membuat keputusan. Hal ini menjadi kritis pada aplikasi yang menuntut transparansi tinggi, seperti diagnosis medis, sistem hukum, dan keuangan (Rane et al., 2024).

4.5.2 Tantangan Etis dan Sosial

Tantangan etis dalam pembelajaran mesin mencakup bias dan fairness. Model dapat memperkuat bias yang sudah ada dalam data, sehingga menghasilkan diskriminasi terhadap kelompok minoritas atau rentan. Privasi data juga menjadi perhatian utama. Penggunaan dataset besar yang berisi informasi pribadi menimbulkan risiko kebocoran data atau penyalahgunaan informasi individu. Akuntabilitas dan tanggung jawab menjadi tantangan ketika model membuat kesalahan atau keputusan yang merugikan.

Tidak selalu jelas pihak mana yang bertanggung jawab atas keputusan otomatis yang dihasilkan model. Selain itu, penerapan pembelajaran mesin dapat menimbulkan dampak sosial dan ekonomi, seperti otomatisasi pekerjaan yang dapat menggantikan manusia, meningkatkan pengangguran, dan memperlebar kesenjangan ekonomi (Mittal et al., 2024).

4.5.3 Tren dan Arah Masa Depan

Beberapa tren penting dalam perkembangan pembelajaran mesin antara lain (Deshpande & Sharma, 2025):

1. *Transfer learning*, yaitu memanfaatkan pengetahuan dari satu tugas untuk mempercepat pembelajaran pada tugas lain, sehingga memungkinkan penggunaan dataset lebih kecil.
2. *Federated learning*, di mana model dilatih secara terdistribusi di perangkat lokal tanpa harus memindahkan data mentah ke server pusat, meningkatkan privasi dan keamanan data.
3. Pembelajaran penguatan berskala besar, yang telah menunjukkan hasil mengesankan dalam permainan kompleks, robotika, dan pengendalian sistem otonom.
4. *Quantum machine learning*, yang berpotensi menyelesaikan beberapa masalah secara eksponensial lebih cepat dibanding komputer klasik.
5. *Explainable AI* (XAI), yang bertujuan meningkatkan transparansi dan kepercayaan pengguna terhadap keputusan model, menjadi semakin penting terutama dalam bidang kritis.

6. *Meta-learning* atau “*learning to learn*”, yang memungkinkan model untuk mempelajari algoritma pembelajaran itu sendiri sehingga dapat beradaptasi lebih cepat pada tugas baru.

Dengan mengatasi tantangan teknis, metodologis, dan etis, serta memanfaatkan tren dan inovasi terbaru, pembelajaran mesin diprediksi akan semakin matang, dapat diterapkan secara aman dan bertanggung jawab, serta menjadi fondasi penting bagi pengembangan sistem cerdas di masa depan.

4.6 Latihan Soal

1. Jelaskan perbedaan antara pembelajaran terbimbing dan pembelajaran tanpa bimbingan. Berikan contoh konkret untuk masing-masing jenis pembelajaran ini dan jelaskan kasus penggunaan di mana setiap jenis pembelajaran paling tepat digunakan.
2. Bandingkan kinerja dan karakteristik algoritma Decision Tree, Random Forest, dan Support Vector Machine dalam menangani masalah klasifikasi. Diskusikan keuntungan dan kerugian dari masing-masing algoritma serta situasi di mana setiap algoritma paling cocok digunakan.
3. Jelaskan arsitektur dan fungsi Convolutional Neural Networks (CNN) dalam pemrosesan data gambar. Bagaimana operasi konvolusi dan pooling berkontribusi pada ekstraksi fitur? Berikan contoh aplikasi praktis CNN dalam bidang kesehatan atau industri.

4. Diskusikan tantangan utama dalam menerapkan pembelajaran mesin dalam aplikasi dunia nyata. Fokuskan pada masalah data quality, overfitting, interpretabilitas model, dan pertimbangan etis. Bagaimana cara mengatasi tantangan-tantangan ini?
5. Jelaskan konsep ensemble methods dalam pembelajaran mesin dan bagaimana metode seperti bagging, boosting, dan stacking bekerja untuk meningkatkan performa prediksi. Berikan contoh kapan ensemble methods lebih disukai dibandingkan model tunggal.

Bab 5: Algoritma Klasifikasi dan Prediksi

5.1 Pengenalan Algoritma Klasifikasi

Algoritma klasifikasi merupakan teknik pembelajaran mesin yang fundamental dalam *machine learning* untuk memprediksi kategori atau kelas dari data baru berdasarkan pola yang dipelajari dari data pelatihan. Klasifikasi adalah jenis pembelajaran terarah (*supervised learning*) yang memiliki aplikasi luas di berbagai bidang seperti diagnosis medis, deteksi spam, pengenalan gambar, dan prediksi churn pelanggan (Deist et al., 2018). Dalam klasifikasi, sistem dilatih dengan dataset yang berisi fitur-fitur masukan dan label kelas yang sesuai, sehingga dapat mempelajari pola hubungan antara fitur dan kelas target.

5.1.1 Konsep Dasar dan Terminologi

Klasifikasi pada dasarnya adalah proses pengelompokan data ke dalam kategori yang telah ditentukan sebelumnya. Setiap sampel data dalam proses pelatihan terdiri dari vektor fitur dan label kelas yang diketahui. Model klasifikasi belajar dari data pelatihan ini untuk mengembangkan fungsi keputusan yang dapat memetakan fitur masukan ke label kelas dengan akurasi tinggi (Maxwell et al., 2018). Terminologi penting dalam klasifikasi mencakup fitur

(atribut masukan), label (kelas target), model (fungsi yang dipelajari), dan prediksi (output model untuk data baru).

5.1.2 Jenis-jenis Masalah Klasifikasi

Masalah klasifikasi dapat dikategorikan menjadi beberapa jenis berdasarkan jumlah kelas target. Klasifikasi biner adalah kasus paling sederhana dengan dua kelas yang saling eksklusif, seperti prediksi apakah pasien menderita penyakit tertentu atau tidak. Klasifikasi multi-kelas melibatkan lebih dari dua kategori, seperti klasifikasi jenis bunga *Iris* yang memiliki tiga spesies berbeda (Berhane et al., 2018). Klasifikasi multi-label terjadi ketika setiap sampel dapat termasuk dalam beberapa kelas secara bersamaan, seperti kategorisasi artikel berita yang dapat memiliki beberapa topik.

5.1.3 Tahapan Proses Klasifikasi

Proses klasifikasi melibatkan beberapa tahapan penting. Pertama adalah pengumpulan dan pemrosesan data yang mencakup pembersihan data, penanganan nilai yang hilang, dan normalisasi fitur. Tahap kedua adalah pemilihan fitur untuk mengurangi dimensionalitas dan meningkatkan efisiensi model. Ketiga adalah pembagian data menjadi set pelatihan dan pengujian dengan rasio yang tepat. Keempat adalah pelatihan model dengan algoritma yang dipilih. Terakhir adalah evaluasi model menggunakan metrik performa yang sesuai (Aslan et al., 2018).

5.1.4 Evaluasi dan Metrik Performa Klasifikasi

Evaluasi model klasifikasi memerlukan metrik yang tepat untuk mengukur performa secara akurat. Akurasi adalah metrik

paling dasar yang menghitung persentase prediksi yang benar dari total prediksi. Namun, akurasi tidak selalu cukup terutama pada dataset yang tidak seimbang. Presisi mengukur dari prediksi positif yang benar, sementara recall (sensitivitas) mengukur proporsi sampel positif yang teridentifikasi dengan benar. F1-score adalah rata-rata harmonis dari presisi dan recall yang memberikan gambaran seimbang dari performa model (Wang et al., 2018). Kurva ROC (Receiver Operating Characteristic) dan AUC (Area Under Curve) juga penting untuk evaluasi menyeluruh.

5.2 Algoritma Klasifikasi Berbasis Pohon Keputusan

Pohon keputusan (*Decision Tree*) adalah algoritma klasifikasi yang mudah dipahami dan cukup populer karena interpretabilitas tinggi dan kemampuan menangani data non-linear. Algoritma ini membuat keputusan dengan mempartisi ruang fitur secara rekursif menggunakan kondisi pertanyaan pada setiap simpul.

5.2.1 Prinsip Kerja Pohon Keputusan

Pohon keputusan membangun model dengan membuat serangkaian pertanyaan tentang nilai fitur untuk membagi data menjadi subset yang semakin homogen. Setiap simpul internal merepresentasikan tes pada fitur tertentu, setiap cabang merepresentasikan hasil tes, dan setiap daun merepresentasikan label kelas. Proses pembelajaran memilih fitur dan nilai ambang yang memberikan pemisahan terbaik berdasarkan kriteria seperti *gain*

informasi atau *Gini impurity* (Alom et al., 2018). Keputusan dibuat secara greedy di setiap tingkat untuk memaksimalkan homogenitas kelas dalam subset yang dihasilkan.

5.2.2 Keuntungan dan Keterbatasan Pohon Keputusan

Keuntungan utama pohon keputusan adalah interpretabilitas tinggi karena aturan keputusan dapat dengan mudah dijelaskan kepada pengguna yang tidak memiliki latar belakang teknis. Algoritma ini tidak memerlukan normalisasi fitur dan dapat menangani fitur kategori dan numerik tanpa transformasi khusus. Pohon keputusan juga cepat dalam membuat prediksi untuk data baru. Namun, keterbatasan signifikan adalah kecenderungan overfitting terutama pada dataset dengan banyak fitur atau pohon yang terlalu dalam. Model tunggal sering kali kurang stabil dan akurasi lebih rendah dibanding algoritma ensemble (Kirichenko et al., 2018).

5.2.3 Teknik Pemangkasan dan Optimalisasi

Untuk mengatasi *overfitting*, teknik pemangkasan (*pruning*) dilakukan untuk menghapus cabang pohon yang tidak signifikan. Pemangkasan dapat dilakukan sebelum atau setelah proses pembelajaran selesai. Pembatasan kedalaman pohon (*max depth*), jumlah sampel minimum di setiap simpul, dan faktor kompleksitas adalah hyperparameter penting yang dapat disesuaikan untuk mengoptimalkan performa. Validasi silang (*cross-validation*) digunakan untuk menentukan nilai hyperparameter terbaik yang meminimalkan kesalahan generalisasi (Ahmadi et al., 2018).

5.2.4 Variasi Algoritma Pohon Keputusan

Terdapat beberapa variasi algoritma pohon keputusan yang dikembangkan oleh peneliti berbeda. C4.5 dan ID3 adalah algoritma klasik yang menggunakan *information gain*. CART (*Classification and Regression Trees*) menggunakan *Gini impurity* dan dapat digunakan untuk regresi maupun klasifikasi. Algoritma-algoritma ini memiliki keunggulan dan kelemahan yang berbeda dalam menangani berbagai jenis dataset dan masalah klasifikasi (Shrivastava & Dubey, 2018).

5.3 Algoritma Ensemble: Random Forest dan Gradient Boosting

Algoritma ensemble menggabungkan prediksi dari beberapa model dasar untuk menghasilkan prediksi yang lebih akurat dan robust. Random Forest dan Gradient Boosting adalah dua algoritma ensemble yang sangat populer dan terbukti efektif dalam berbagai aplikasi (Deist et al., 2018).

5.3.1 Random Forest

Random Forest adalah algoritma ensemble yang membangun banyak pohon keputusan dengan data dan fitur yang dipilih secara acak. Setiap pohon dalam forest dilatih pada subset acak dari data pelatihan (*bootstrap samples*), dan pada setiap pemisahan, hanya subset acak fitur yang dievaluasi (Berhane et al., 2018). Prediksi akhir diperoleh melalui voting mayoritas untuk klasifikasi. Keunggulan Random Forest adalah mengurangi *overfitting*,

meningkatkan akurasi, dan dapat menangani data dengan banyak fitur serta interaksi non-linear. Algoritma ini juga memberikan ukuran kepentingan fitur yang bermanfaat untuk analisis data (Kirichenko et al., 2018).

5.3.2 Gradient Boosting

Gradient Boosting membangun model ensemble secara sekuensial di mana setiap model baru dilatih untuk memperbaiki kesalahan model sebelumnya. Proses pembelajaran melibatkan perhitungan gradien dari fungsi loss dan membuat model baru yang meminimalkan kesalahan residual (Alom et al., 2018). Algoritma ini sangat powerful dan sering menghasilkan akurasi tertinggi dalam kompetisi *machine learning*. Varian modern seperti XGBoost, LightGBM, dan CatBoost telah meningkatkan efisiensi komputasi dan performa model.

5.3.3 Perbandingan Performa Ensemble Methods

Penelitian empiris menunjukkan bahwa Random Forest dan Gradient Boosting sering memberikan performa yang lebih baik dibanding model tunggal. Dalam studi yang membandingkan berbagai algoritma klasifikasi pada dataset radiotherapy, Random Forest dan elastic net logistic regression menunjukkan diskriminasi terbaik dengan AUC median 0.72 pada multiple datasets (Deist et al., 2018). Gradient Boosting juga terbukti memberikan akurasi yang kompetitif atau lebih baik daripada algoritma lain dalam berbagai aplikasi seperti deteksi spam, klasifikasi sentimen, dan prediksi penyakit.

5.3.4 Hyperparameter Tuning untuk Ensemble Methods

Optimalisasi hyperparameter ensemble methods seperti jumlah pohon, kedalaman pohon, *learning rate*, dan *regularization* sangat penting untuk mencapai performa optimal. Teknik seperti grid search, random search, dan Bayesian optimization dapat digunakan untuk menemukan kombinasi *hyperparameter* terbaik (Aslan et al., 2018). Validasi silang dengan lipatan banyak (*k-fold cross-validation*) diterapkan untuk memastikan generalisasi model yang baik pada data yang tidak terlihat.

5.4 Support Vector Machine dan Naive Bayes

Support Vector Machine (*SVM*) dan Naive Bayes adalah algoritma klasifikasi yang berbasis pada prinsip probabilitas dan geometri yang berbeda dengan pohon keputusan.

5.4.1 Support Vector Machine (SVM)

Support Vector Machine adalah algoritma powerful yang menemukan *hyperplane* optimal untuk memisahkan dua kelas dengan margin maksimal. Konsep margin maksimal memberikan generalisasi yang baik pada data baru. Untuk masalah non-linear, SVM menggunakan *kernel trick* untuk memetakan data ke ruang dimensi tinggi di mana pemisahan linear mungkin terjadi (Maxwell et al., 2018). Kernel yang umum digunakan termasuk linear, polynomial, radial basis function (RBF), dan sigmoid. SVM effective untuk klasifikasi biner dan multi-kelas, serta dapat digunakan untuk regresi.

Konsep dasar SVM berakar pada pencarian garis pemisah (untuk data 2 dimensi) atau bidang pemisah (untuk dimensi tinggi) yang paling efisien dalam membedakan antar kelompok data.

1. Prinsip Dasar dan Maksimalisasi Margin

Secara matematis, tujuan utama SVM adalah mencari fungsi keputusan yang didefinisikan sebagai:

$$f(x) = \text{sign}(w \cdot x - b)$$

Di mana w adalah vektor bobot yang menentukan orientasi garis dan b adalah bias (Vapnik, 1995). Berbeda dengan klasifikasi linear biasa yang hanya mencari garis asal pisah, SVM berfokus pada maksimalisasi Margin yaitu jarak terlebar antara garis pemisah dengan titik data terdekat dari masing-masing kelas (Cortes & Vapnik, 1995). Garis dengan margin terbesar dianggap optimal karena memiliki kemampuan generalisasi yang lebih baik dan lebih tahan terhadap *noise* pada data baru.

2. Peran Support Vectors

Kekuatan unik dari SVM terletak pada penggunaan *Support Vectors*. Ini adalah sekumpulan titik data yang berada tepat di tepi margin dan menjadi "penopang" bagi posisi serta orientasi dari *hyperplane* (Navastara, n.d.). Menariknya, jika kita menghapus seluruh data lain dan hanya menyisakan *support vectors* ini, posisi garis pemisah tidak akan berubah sedikit pun. Hal ini membuat SVM sangat efisien karena model akhir hanya bergantung pada segelintir data kritis tersebut (Maxwell et al., 2018).

3. Penanganan Data Non-Linear: Kernel Trick

Dalam skenario dunia nyata, data sering kali tidak dapat dipisahkan secara lurus (*linearly separable*). Untuk mengatasi tantangan ini, SVM menggunakan teknik yang disebut Kernel Trick. Teknik ini memetakan data dari ruang dimensi rendah ke ruang dimensi yang lebih tinggi menggunakan fungsi kernel (Vapnik, 1995). Di ruang dimensi tinggi tersebut, sebuah bidang datar dapat ditarik untuk memisahkan data yang sebelumnya tampak tercampur. Beberapa jenis kernel yang sering digunakan yaitu:

- Linear: Digunakan jika data sudah dapat dipisahkan secara sederhana dengan garis lurus.
- Polynomial: Cocok untuk pola data yang memiliki interaksi antar fitur yang lebih kompleks.
- Radial Basis Function (RBF): Kernel yang paling populer karena kemampuannya menangani sebaran data yang membentuk pola melingkar atau tidak beraturan.

4. Soft Margin dan Parameter C

Untuk menangani data yang memiliki banyak *noise* atau *outlier*, SVM menerapkan konsep *Soft Margin*. Pendekatan ini memberikan toleransi terhadap beberapa data yang "menyeberang" garis demi mendapatkan margin yang lebih lebar secara keseluruhan (Cortes & Vapnik, 1995). Fleksibilitas ini dikontrol oleh parameter regulasi nilai C ; nilai C yang besar akan meminimalkan kesalahan klasifikasi namun mempersempit margin, sementara C yang kecil akan memberikan margin yang lebih lebar dengan risiko adanya beberapa data yang salah klasifikasi.

5.4.2 Naive Bayes

Naive Bayes adalah algoritma berbasis probabilitas yang mengaplikasikan teorema Bayes dengan asumsi bahwa semua fitur independen secara kondisional diberikan label kelas. Meskipun asumsi ini jarang terpenuhi dalam praktik, algoritma ini sering memberikan hasil yang baik dalam banyak aplikasi (Wang et al., 2018). Naive Bayes sangat cepat dalam training dan prediksi, robust terhadap data noise, dan cocok untuk dataset dengan banyak fitur. Varian Naive Bayes meliputi Gaussian, Multinomial, dan Bernoulli yang disesuaikan dengan distribusi fitur yang berbeda.

5.4.3 Perbandingan SVM dan Naive Bayes

Dalam berbagai aplikasi klasifikasi, SVM umumnya menghasilkan akurasi lebih tinggi dibanding Naive Bayes karena kemampuannya menangani hubungan non-linear dan mencari hyperplane optimal. Namun, Naive Bayes memiliki keunggulan dalam hal kecepatan training, interpretabilitas, dan kemampuan menangani dataset kecil (Alom et al., 2018). Penelitian pada klasifikasi email spam menunjukkan bahwa baik SVM maupun Naive Bayes dapat mencapai akurasi lebih dari 95%, tetapi SVM sering lebih akurat dengan preprocessing yang tepat.

5.4.4 Aplikasi dan Tuning Parameter

SVM memerlukan tuning careful pada parameter kernel dan regularization (C , γ) untuk mencapai performa optimal. Naive Bayes memiliki sedikit hyperparameter yang perlu tuning, menjadikannya lebih mudah digunakan dalam praktik. Kedua algoritma efektif untuk berbagai aplikasi termasuk deteksi anomali,

klasifikasi teks, dan prediksi medis, masing-masing dengan kelebihan yang berbeda tergantung karakteristik dataset (Ahmadi et al., 2018).

5.5 K-Nearest Neighbors dan Logistic Regression

K-Nearest Neighbors (*KNN*) dan Logistic Regression mewakili dua pendekatan yang berbeda untuk klasifikasi dengan karakteristik unik masing-masing.

5.5.1 K-Nearest Neighbors (KNN)

K-Nearest Neighbors adalah algoritma *lazy learning* yang menyimpan seluruh data pelatihan dan membuat prediksi berdasarkan k tetangga terdekat dari sampel query. Prediksi untuk klasifikasi diperoleh melalui voting mayoritas dari label tetangga terdekat (Maxwell et al., 2018). Keuntungan KNN adalah kesederhanaan implementasi, tidak memerlukan training fase eksplisit, dan dapat menangani masalah multi-kelas dengan mudah. Namun, KNN memiliki kelemahan berupa computational cost tinggi selama prediksi, sensitif terhadap feature scaling, dan performa menurun dengan data dimensi tinggi (*curse of dimensionality*).

5.5.2 Logistic Regression

Logistic Regression adalah algoritma klasifikasi linear yang menggunakan fungsi logistik untuk memodelkan probabilitas sampel termasuk dalam kelas positif. Meskipun namanya mengandung kata "regression", algoritma ini digunakan untuk klasifikasi melalui thresholding pada probabilitas output (Wang et

al., 2018). Logistic Regression efficient dalam training dan prediksi, mudah diinterpretasi karena output adalah probabilitas, dan less prone to *overfitting* pada dataset kecil. Algoritma ini cocok sebagai baseline model dan sering memberikan hasil yang kompetitif pada dataset dengan hubungan linear antara fitur dan kelas.

5.5.3 Pemilihan K pada KNN

Pemilihan nilai K sangat krusial dalam KNN karena mempengaruhi performa model signifikan. K terlalu kecil dapat menyebabkan overfitting dan sensitif terhadap noise, sementara K terlalu besar dapat menyebabkan underfitting. Validasi silang digunakan untuk menemukan nilai K optimal (Alom et al., 2018). Jarak yang digunakan juga penting, dengan jarak Euclidean paling umum, tetapi jarak Manhattan atau Minkowski dapat lebih efektif pada beberapa dataset.

5.5.4 Regularisasi dalam Logistic Regression

Untuk mencegah overfitting pada Logistic Regression, teknik regularisasi seperti L1 (*Lasso*) dan L2 (*Ridge*) dapat diterapkan yang menambahkan penalty pada koefisien model. Elastic Net menggabungkan L1 dan L2 regularization untuk fleksibilitas lebih. Strength regularisasi dikontrol melalui hyperparameter lambda yang dapat disesuaikan menggunakan cross-validation (Maxwell et al., 2018). Logistic Regression dengan regularisasi yang tepat sering mengungguli model lebih kompleks pada data dengan banyak fitur yang berkorelasi.

5.6 Deep Learning untuk Klasifikasi

Deep learning telah merevolusi bidang klasifikasi dengan kemampuan menangani data kompleks seperti gambar, teks, dan audio dengan akurasi yang mengesankan.

5.6.1 Artificial Neural Networks

Artificial Neural Networks (ANN) atau Multi-Layer Perceptron (MLP) terdiri dari lapisan-lapisan node yang terhubung dengan bobot yang dapat dipelajari. Lapisan input menerima fitur data, satu atau lebih lapisan hidden memproses informasi melalui fungsi aktivasi non-linear, dan lapisan output menghasilkan prediksi (Neeraj et al., 2020). Deep neural networks dengan banyak lapisan hidden dapat menangani pola kompleks dan non-linear, tetapi memerlukan data pelatihan besar dan power komputasi tinggi.

5.6.2 Convolutional Neural Networks

Convolutional Neural Networks (CNN) dirancang khusus untuk data grid seperti gambar. Lapisan konvolusi menerapkan filter ke input untuk ekstraksi fitur otomatis, sementara lapisan pooling mengurangi dimensionalitas (Goldstein et al., 2020). CNN telah mencapai state-of-the-art pada berbagai task visi komputer termasuk klasifikasi gambar, deteksi objek, dan segmentasi semantik. Arsitektur populer seperti VGG, ResNet, dan Inception telah menjadi fondasi untuk banyak aplikasi praktis.

5.6.3 Recurrent Neural Networks

Recurrent Neural Networks (RNN) dengan varian Long Short-Term Memory (LSTM) dan Gated Recurrent Unit (GRU)

efektif untuk data sekuensial seperti teks dan time series. Struktur recursif memungkinkan jaringan mempertahankan memori informasi sebelumnya, penting untuk tugas klasifikasi yang mempertimbangkan konteks temporal (Bahri et al., 2020). LSTM mengatasi vanishing gradient problem dengan mekanisme gating yang sophisticated, menjadikannya pilihan populer untuk NLP tasks.

5.6.4 Transfer Learning dan Fine-tuning

Transfer learning memanfaatkan model pre-trained pada dataset besar untuk task baru dengan data pelatihan terbatas. Fine-tuning melibatkan pelatihan ulang sebagian layer dari model pre-trained dengan data spesifik domain (Enguehard et al., 2019). Pendekatan ini sangat efektif karena menghemat waktu training dan data, serta menghasilkan model dengan akurasi tinggi. Contoh aplikasi termasuk penggunaan ImageNet pre-trained models untuk medical image classification dengan hasil yang competitive.

5.7 Latihan Soal

1. Jelaskan perbedaan mendasar antara klasifikasi biner dan klasifikasi multi-kelas, serta berikan contoh aplikasi masing-masing dalam kehidupan nyata.
2. Bandingkan keuntungan dan keterbatasan antara Random Forest dan Logistic Regression untuk masalah klasifikasi. Dalam situasi apa Anda akan memilih satu dibanding yang lain?

3. Mengapa *overfitting* merupakan masalah signifikan pada pohon keputusan tunggal, dan bagaimana ensemble methods seperti Random Forest mengatasi masalah ini?
4. Jelaskan bagaimana Support Vector Machine menggunakan *kernel trick* untuk menangani masalah klasifikasi non-linear, dan berikan contoh kernel yang umum digunakan.
5. Diskusikan trade-offs antara akurasi dan interpretabilitas dalam memilih algoritma klasifikasi. Algoritma mana yang prioritaskan untuk aplikasi medis versus aplikasi big data analytics?

Bab 6: Jaringan Syaraf Tiruan

6.1 Pengertian dan Konsep Dasar Jaringan Syaraf Tiruan

6.1.1 Definisi Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (JST) atau *Artificial Neural Networks* (ANN) adalah suatu sistem komputasi yang terinspirasi dari struktur dan fungsi jaringan biologis dalam otak manusia. JST terdiri atas sejumlah unit pemrosesan yang saling terhubung, yang disebut neuron atau *node*, yang bekerja secara kolektif untuk memproses informasi dan melakukan komputasi. Setiap neuron dalam JST dapat menerima input dari neuron-neuron lainnya, memproses informasi tersebut melalui fungsi aktivasi tertentu, dan menghasilkan output yang diteruskan ke neuron-neuron berikutnya (Hammad, 2024).

Konsep dasar JST bermula dari pemodelan matematis tentang bagaimana neuron biologis bekerja dalam mentransmisikan sinyal. Dalam model neuron artificial ini, setiap input dikalikan dengan bobot (*weight*) yang menunjukkan kekuatan koneksi, kemudian semua input terbobot dijumlahkan dan ditambahkan dengan bias untuk diproses oleh fungsi aktivasi (Li, 2024).

6.1.2 Struktur dan Arsitektur Jaringan Syaraf Tiruan

Arsitektur JST umumnya terdiri dari tiga lapisan utama: lapisan input (*input layer*), lapisan tersembunyi (*hidden layer*), dan

lapisan output (*output layer*). Lapisan input menerima data mentah dari luar sistem. Lapisan tersembunyi melakukan transformasi nonlinear terhadap data input melalui serangkaian neuron yang saling terhubung. Lapisan output menghasilkan keluaran akhir dari jaringan (Hammad, 2024).

Setiap neuron dalam JST dihubungkan dengan neuron-neuron di lapisan berikutnya melalui koneksi dengan bobot yang dapat disesuaikan. Jaringan dengan banyak lapisan tersembunyi disebut *deep neural network* atau jaringan syaraf dalam, yang mampu menangani hubungan nonlinear yang sangat kompleks dalam data (Li, 2024). Topologi jaringan dapat bervariasi, mulai dari jaringan feedforward yang sederhana hingga jaringan recurrent yang memiliki koneksi umpan balik.

6.1.3 Jenis-Jenis Jaringan Syaraf Tiruan

Terdapat berbagai jenis JST yang dirancang untuk aplikasi yang berbeda. *Convolutional Neural Networks* (CNN) dirancang khusus untuk memproses data berstruktur grid seperti gambar dan video (Li et al., 2021). CNN menggunakan lapisan konvolusi yang dapat mendeteksi fitur lokal dengan efisien. *Recurrent Neural Networks* (RNN) dirancang untuk memproses data sekuensial seperti teks dan data *time series*, dengan kemampuan untuk mengingat informasi dari waktu sebelumnya (Mienye et al., 2024).

Varian penting dari RNN adalah *Long Short-Term Memory* (LSTM) dan *Gated Recurrent Unit* (GRU), yang dapat mengatasi masalah *vanishing gradient* yang sering terjadi pada RNN standar. Terdapat pula *Spiking Neural Networks* (SNNs) yang

lebih mirip dengan neuron biologis, menggunakan spike events untuk transmisi informasi (Tavanaei et al., 2018). Setiap jenis jaringan memiliki kekuatan dan kelemahan yang berbeda tergantung pada jenis masalah yang akan diselesaikan.

6.1.4 Parameter dan Komponen Penting dalam Jaringan Syaraf Tiruan

Parameter penting dalam JST meliputi bobot (*weight*) dan bias yang menghubungkan antar neuron. Bobot menentukan kekuatan pengaruh dari satu neuron ke neuron lainnya, sementara bias memungkinkan pengaturan nilai ambang dari neuron. Fungsi aktivasi adalah fungsi nonlinear yang diterapkan pada output neuron untuk memperkenalkan nonlinearitas ke dalam model (Hammad, 2024).

Fungsi aktivasi yang umum digunakan meliputi sigmoid, tanh, ReLU (*Rectified Linear Unit*), dan variannya. Pemilihan fungsi aktivasi yang tepat sangat penting untuk kinerja jaringan. Parameter lainnya yang penting adalah learning rate yang mengontrol kecepatan pembaruan bobot selama pelatihan, dan batch size yang menentukan jumlah sampel data yang diproses sebelum pembaruan bobot dilakukan (Li, 2024).

6.2 Proses Pembelajaran dan Pelatihan Jaringan Syaraf Tiruan

6.2.1 Algoritma Backpropagation

Backpropagation adalah algoritma pembelajaran yang paling banyak digunakan untuk melatih JST *feedforward*. Algoritma ini bekerja dalam dua fase: fase *forward pass* di mana input diproses melalui jaringan untuk menghasilkan output, dan fase *backward pass* di mana error dihitung dan diteruskan balik melalui jaringan untuk memperbarui bobot (Al-Hajj et al., 2024).

Dalam *forward pass*, setiap neuron menghitung output berdasarkan input dan bobot yang ada. Dalam *backward pass*, gradien dari fungsi loss terhadap setiap bobot dihitung menggunakan aturan rantai (*chain rule*) dari kalkulus. Bobot kemudian diperbarui dalam arah yang berlawanan dengan gradien untuk mengurangi error (Hazrati et al., 2024). Proses ini diulang untuk banyak epoch hingga jaringan mencapai konvergensi atau memenuhi kriteria berhenti yang telah ditetapkan.

6.2.2 Algoritma Optimasi dan Tuning Hyperparameter

Meskipun backpropagation efektif, algoritma ini dapat mengalami beberapa masalah seperti terjebak pada minimum lokal, *vanishing gradient*, dan *exploding gradient*. Untuk mengatasi masalah-masalah ini, berbagai algoritma optimasi telah dikembangkan, seperti *Stochastic Gradient Descent* (SGD), Adam, RMSprop, dan *Adagrad* (Eker et al., 2023). Algoritma-algoritma ini

menyesuaikan learning rate secara adaptif berdasarkan gradien historis.

Selain itu, terdapat algoritma berbasis evolusi (*evolutionary algorithms*) seperti *Genetic Algorithm*, *Particle Swarm Optimization* (PSO), dan *Ant Colony Optimization* (ACO) yang dapat digunakan untuk melatih JST (Kurilovich, 2022). Algoritma-algoritma ini berguna ketika landscape fungsi loss sangat kompleks dan sulit untuk dioptimasi menggunakan gradient descent biasa. Tuning hyperparameter seperti learning rate, batch size, jumlah layer, dan jumlah neuron sangat penting untuk mencapai performa optimal (Al-Hajj et al., 2024).

6.2.3 Teknik Regularisasi dan Pencegahan Overfitting

Overfitting terjadi ketika JST belajar terlalu baik terhadap data pelatihan tetapi gagal menggeneralisasi dengan baik terhadap data baru. Untuk mengatasi ini, berbagai teknik regularisasi digunakan. L1 dan L2 *regularization* menambahkan penalti terhadap bobot yang besar pada fungsi loss, mendorong jaringan untuk belajar bobot yang lebih kecil dan lebih sederhana (Zhang et al., 2021).

Dropout adalah teknik yang secara acak menghilangkan beberapa unit neuron selama pelatihan, yang memaksa jaringan untuk belajar representasi yang lebih robust dan mengurangi co-adaptation antar unit (Shorten & Khoshgoftaar, 2019). *Early stopping* menghentikan pelatihan ketika performa pada data validasi mulai memburuk, mencegah jaringan untuk belajar noise dalam data pelatihan. *Batch normalization* menstandarisasi input ke setiap

lapisan, yang dapat mempercepat pembelajaran dan mengurangi sensitivitas terhadap inisialisasi bobot (Li, 2024).

6.2.4 Evaluasi dan Validasi Model Jaringan Syaraf Tiruan

Evaluasi yang tepat adalah kunci untuk menilai apakah JST telah belajar dengan baik dan dapat menggeneralisasi ke data baru. Metrik evaluasi yang umum digunakan meliputi *Mean Squared Error* (MSE) untuk regresi dan *accuracy*, *precision*, *recall*, dan *F1-score* untuk klasifikasi (Hazrati et al., 2024). Untuk masalah yang lebih kompleks, *Area Under the Curve* (AUC) dari *Receiver Operating Characteristic* (ROC) dapat memberikan gambaran yang lebih komprehensif tentang performa model.

Teknik *cross-validation* digunakan untuk memastikan bahwa hasil evaluasi tidak tergantung pada pembagian data pelatihan dan pengujian yang spesifik. Dalam *k-fold cross-validation*, data dibagi menjadi k bagian, dan model dilatih k kali dengan setiap bagian sebagai data pengujian dan bagian lainnya sebagai data pelatihan (Al-Hajj et al., 2024). Hasil evaluasi dari semua fold kemudian di-rata-ratakan untuk mendapatkan estimasi performa yang lebih stabil dan dapat diandalkan.

6.3 Arsitektur Jaringan Syaraf Tiruan yang Lanjut

6.3.1 Convolutional Neural Networks untuk Pengolahan Citra

CNN adalah arsitektur khusus yang dirancang untuk bekerja dengan data yang memiliki struktur grid, terutama citra digital. Lapisan konvolusi dalam CNN menerapkan filter-filter kecil

(*kernels*) ke input untuk mendeteksi fitur lokal seperti tepi, tekstur, dan bentuk. Setiap filter dikonvolusi dengan input untuk menghasilkan *feature map* yang menangkap kehadiran fitur tertentu pada lokasi yang berbeda (Li et al., 2021).

Setelah lapisan konvolusi, biasanya terdapat lapisan *pooling* yang mengurangi dimensi spasial dari feature maps dengan mengambil nilai maksimum atau rata-rata dari daerah tetangga. Ini mengurangi komputasi dan membantu membuat fitur yang terdeteksi lebih invariant terhadap transformasi kecil (Zhao et al., 2018). Lapisan konvolusi dan *pooling* dapat ditumpuk untuk membentuk hirarki fitur yang semakin abstrak dan kompleks. Akhirnya, *fully connected layers* digunakan untuk melakukan klasifikasi atau tugas lain berdasarkan fitur-fitur yang telah diekstrak.

6.3.2 Recurrent Neural Networks untuk Data Sekuensial

RNN dirancang untuk memproses data sekuensial dengan mempertahankan *hidden state* yang terus diperbarui seiring dengan pemrosesan setiap elemen dalam urutan (Mienye et al., 2024). Ini memungkinkan RNN untuk mengingat informasi dari waktu sebelumnya, menjadikannya sangat cocok untuk tugas-tugas seperti pemodelan bahasa, terjemahan mesin, dan peramalan *time series*.

LSTM adalah varian dari RNN yang mengatasi masalah *vanishing gradient* dengan memperkenalkan gating mechanisms yang mengontrol aliran informasi. LSTM memiliki unit memori (*cell state*) yang dapat menyimpan informasi untuk jangka waktu yang panjang, dengan tiga gate (*forget gate*, *input gate*,

dan *output gate*) yang mengatur apa yang harus diingat, apa yang harus ditambahkan, dan apa yang harus dikeluarkan (Apaydn et al., 2020). GRU adalah simplifikasi dari LSTM yang menggabungkan beberapa gate dan mengurangi jumlah parameter sambil tetap mempertahankan kemampuan untuk menangani dependensi jangka panjang.

6.3.3 Hybrid Architectures dan Network-in-Network

Hybrid architectures menggabungkan kekuatan dari berbagai jenis jaringan. Misalnya, CNN-LSTM menggabungkan lapisan konvolusi untuk ekstraksi fitur spasial dengan lapisan LSTM untuk pemodelan temporal (Zhang et al., 2023). Arsitektur ini sangat berguna untuk tugas-tugas seperti action recognition dalam video, di mana ada struktur spasial dalam frame dan struktur temporal dalam urutan frame.

Teknik *attention mechanism* telah menjadi komponen penting dalam banyak arsitektur modern. Attention memungkinkan jaringan untuk fokus pada bagian-bagian yang paling relevan dari input, mirip dengan perhatian manusia. Transformer, yang didasarkan sepenuhnya pada attention mechanisms tanpa konvolusi atau recurrence, telah terbukti sangat efektif untuk memproses data sekuensial dan menjadi dasar untuk banyak model bahasa besar yang sukses hari ini (Battaglia et al., 2018).

6.3.4 Transfer Learning dan Fine-tuning

Transfer learning adalah teknik yang memanfaatkan pengetahuan yang telah dipelajari dari satu tugas untuk membantu dalam belajar tugas lain yang terkait. Dalam praktik, model yang

telah dilatih pada dataset besar (seperti ImageNet untuk citra) dapat digunakan sebagai titik awal (*pretrained model*) dan kemudian *fine-tuned* pada dataset yang lebih kecil untuk tugas spesifik (Wen et al., 2018).

Pendekatan ini sangat berguna ketika dataset untuk tugas target terbatas, karena model pretrained telah belajar banyak fitur berguna yang dapat ditransfer. Ada beberapa strategi untuk *fine-tuning*: (1) melatih ulang semua bobot model, (2) hanya melatih beberapa lapisan terakhir sambil membekukan lapisan-lapisan sebelumnya, atau (3) melatih semua bobot dengan learning rate yang lebih kecil dari biasanya. Pilihan strategi tergantung pada jumlah dan karakteristik dari dataset target.

6.4 Aplikasi Jaringan Syaraf Tiruan

6.4.1 Aplikasi dalam Pengenalan Citra dan Visi Komputer

Salah satu aplikasi paling sukses dari JST adalah dalam pengenalan citra dan visi komputer. CNN telah mencapai akurasi manusia dan bahkan melampaui dalam tugas-tugas seperti klasifikasi citra, deteksi objek, dan segmentasi semantik (Zhao et al., 2018). Dalam deteksi melanoma, model CNN telah terbukti dapat mengungguli dermatolog berpengalaman dalam mengidentifikasi kasus-kasus keganasan (Haenssle et al., 2018).

Aplikasi lain termasuk pengenalan wajah, yang memiliki implikasi penting untuk keamanan dan privasi. Model CNN yang mendalam dapat mengekstrak fitur-fitur wajah yang sangat

diskriminatif dan tahan terhadap variasi dalam pose, pencahayaan, dan ekspresi. Deteksi dan segmentasi objek dalam citra medis, seperti tumor otak dalam MRI atau kanker payudara dalam mammogram, adalah area aplikasi yang sangat penting di mana akurasi dapat membuat perbedaan hidup atau mati (Maqsood et al., 2022).

6.4.2 Aplikasi dalam Pemrosesan Bahasa Alami dan Analisis Teks

LSTM dan Transformer telah merevolusi pemrosesan bahasa alami (NLP), memungkinkan model untuk memahami dan menghasilkan teks yang bermakna (Mienye et al., 2024). Aplikasi mencakup terjemahan mesin otomatis, analisis sentimen, klasifikasi teks, dan pemodelan bahasa. Model bahasa besar yang dilatih pada corpus teks yang sangat besar telah menunjukkan kemampuan yang menakjubkan dalam berbagai tugas NLP dengan sedikit atau tanpa pelatihan khusus.

Untuk tugas klasifikasi teks dan analisis sentimen, kombinasi CNN dan LSTM sering memberikan hasil yang lebih baik daripada menggunakan salah satu saja (Hidri et al., 2025). CNN dapat menangkap fitur n-gram lokal, sementara LSTM dapat menangkap dependensi jangka panjang dalam teks. Deteksi phishing dan spam adalah aplikasi keamanan siber yang penting di mana NLP dan deep learning dapat membantu mengidentifikasi komunikasi berbahaya (Melon et al., 2025).

6.4.3 Aplikasi dalam Peramalan dan Prediksi

JST, terutama RNN dan LSTM, telah banyak digunakan untuk peramalan *time series*, seperti prediksi harga saham, konsumsi energi, dan data meteorologi. Kemampuan RNN untuk menangkap dependensi temporal membuat mereka sangat cocok untuk jenis data ini (Abdulkadhim et al., 2024). Kombinasi CNN dan LSTM atau model *ensemble* yang menggabungkan berbagai arsitektur sering memberikan performa yang lebih baik daripada model individual (Song & Choi, 2023).

Dalam peramalan permintaan energi, model deep learning telah terbukti mengungguli metode statistik tradisional seperti ARIMA dan metode machine learning lainnya (Gassar, 2024). Dalam prediksi hasil kesehatan, seperti prediksi kanker payudara dari data gen menggunakan RNN hibrida yang menggabungkan lapisan konvolusi dengan LSTM dapat mencapai akurasi hingga 97,5% (S.u. & Katariya, 2024).

6.4.4 Aplikasi dalam Deteksi Anomali dan Keamanan

JST, khususnya autoencoder dan RNN, dapat digunakan untuk deteksi anomali dan identifikasi pola yang tidak biasa dalam data (Oney & Peker, 2018). Dalam deteksi penipuan kartu kredit, model deep learning seperti LSTM dan CNN dapat mengidentifikasi transaksi yang mencurigakan dengan akurasi tinggi (Mienye & Jere, 2024). Dalam sistem keamanan jaringan, deep learning dapat digunakan untuk mendeteksi serangan intrusi dengan menganalisis pola lalu lintas jaringan (Tran et al., 2025).

Untuk pengenalan aktivitas manusia dari data sensor, model CNN-LSTM yang dioptimalkan dapat mencapai akurasi hingga 97,8%, mengungguli pendekatan *machine learning* tradisional (Kumari et al., 2024). Aplikasi ini penting untuk sistem keamanan, surveillance, dan monitoring kesehatan. Dalam konteks pertanian, deep learning dapat digunakan untuk deteksi penyakit tanaman dari citra daun menggunakan CNN, membantu petani dalam mengambil keputusan manajemen yang lebih baik (Slimani et al., 2023).

6.5 Tantangan dan Penelitian Terkini dalam Jaringan Syaraf Tiruan

6.5.1 Tantangan dalam Pelatihan dan Generalisasi

Meskipun JST telah mencapai kesuksesan yang luar biasa, masih ada tantangan yang signifikan. Salah satunya adalah masalah *overfitting* di mana model belajar noise dalam data pelatihan daripada mengextract pola yang benar-benar berguna. Ini terjadi terutama ketika dataset pelatihan kecil atau model terlalu kompleks (Zhang et al., 2021). Masalah *underfitting*, di mana model terlalu sederhana untuk menangkap pola kompleks dalam data, juga perlu dipertimbangkan.

Vanishing dan *exploding gradient* adalah masalah numerik yang terjadi ketika melatih jaringan yang sangat dalam, di mana gradien menjadi terlalu kecil atau terlalu besar saat diteruskan balik melalui banyak lapisan. Meskipun teknik-teknik seperti batch normalization dan inisialisasi bobot yang cermat telah membantu

mengatasi ini, masalah ini tetap menjadi area penelitian yang aktif (Hammad, 2024).

6.5.2 Interpretabilitas dan Explainability

Salah satu kritik utama terhadap deep neural networks adalah bahwa mereka berfungsi sebagai "black box" – sulit untuk memahami bagaimana keputusan tertentu dibuat oleh model. Ini menjadi masalah serius dalam aplikasi yang kritis seperti diagnostik medis dan pengambilan keputusan hukum, di mana interpretabilitas sangat penting (Zhang et al., 2021).

Berbagai teknik telah dikembangkan untuk membuat deep learning models lebih *interpretable*, seperti *feature visualization*, *saliency maps*, dan *attention visualization*. Teknik-teknik ini dapat membantu mengungkap fitur-fitur apa yang dipelajari oleh model dan bagaimana mereka mempengaruhi keputusan akhir. Namun, interpretabilitas tetap menjadi area penelitian yang aktif dengan banyak pertanyaan yang belum terjawab (Garca et al., 2018).

6.5.3 Efisiensi Komputasi dan Mobile Deployment

Melatih model deep learning besar memerlukan komputasi yang sangat intensif dan sering memerlukan GPU atau TPU yang mahal. Selain itu, *men-deploy* model yang terlatih pada perangkat mobile atau embedded memiliki keterbatasan memori dan daya komputasi (Shah et al., 2018). Teknik model *compression* seperti pruning, quantization, dan *knowledge distillation* telah dikembangkan untuk mengurangi ukuran dan kompleksitas model sambil mempertahankan akurasi (Shorten & Khoshgoftaar, 2019).

Baru-baru ini, *neural architecture search* (NAS) telah menjadi area penelitian yang aktif, bertujuan untuk secara otomatis menemukan arsitektur jaringan yang optimal untuk tugas dan sumber daya yang diberikan (Wang et al., 2018). Model yang lebih efisien seperti MobileNet dan EfficientNet telah dikembangkan khusus untuk deployment pada perangkat mobile dengan keseimbangan yang baik antara akurasi dan efisiensi komputasi (Kazeneza et al., 2025).

6.5.4 Keamanan dan Robustness Terhadap Adversarial Attacks

JST telah ditunjukkan rentan terhadap *adversarial examples* – input yang dirancang secara khusus untuk menyebabkan model membuat prediksi yang salah. Ini memiliki implikasi serius untuk keamanan sistem yang mengandalkan *deep learning* (Goswami et al., 2018). Memahami kerentanan ini dan mengembangkan teknik untuk membuatnya lebih *robust* adalah area penelitian yang penting.

Penelitian telah menunjukkan bahwa model yang lebih besar dan lebih kompleks tidak selalu lebih robust, dan bahwa trade-off antara akurasi standar dan robustness terhadap adversarial attacks sering ada. Teknik seperti adversarial training, yang melatih model pada contoh adversarial yang dihasilkan, telah terbukti dapat meningkatkan robustness (Scaman & Virmaux, 2018).

6.6 Latihan Soal

1. Jelaskan perbedaan antara *feedforward neural networks* dan *recurrent neural networks*, serta berikan contoh aplikasi untuk masing-masing jenis jaringan.
2. Uraikan bagaimana algoritma *backpropagation* bekerja dalam melatih jaringan syaraf, termasuk dua fase utama yaitu *forward pass* dan *backward pass*.
3. Diskusikan problem *overfitting* dalam jaringan syaraf tiruan dan jelaskan tiga teknik yang dapat digunakan untuk mencegahnya.
4. Bandingkan antara *Convolutional Neural Networks* (CNN) dan *Recurrent Neural Networks* (RNN) dalam hal arsitektur, kemampuan pemrosesan, dan keunggulan relatif masing-masing.
5. Jelaskan konsep *transfer learning* dan bagaimana teknik ini dapat dimanfaatkan untuk meningkatkan performa model ketika dataset target terbatas.

Bab 7: Deep Learning

7.1 Fondasi dan Konsep Dasar Deep Learning

Deep learning merupakan subset dari machine learning yang menggunakan jaringan neural dengan banyak lapisan untuk mempelajari representasi data secara hierarki. Istilah "*deep*" mengacu pada penggunaan beberapa lapisan hidden dalam arsitektur *neural network* yang memungkinkan model untuk secara otomatis menemukan fitur-fitur yang diperlukan untuk deteksi atau klasifikasi dari data mentah (Elshenawy et al., 2025). Kedalaman jaringan *neural* ini memungkinkan pembelajaran representasi abstrak tingkat tinggi dari data input yang kompleks, menjadikan deep learning sebagai alat yang sangat ampuh untuk berbagai aplikasi dalam computer vision, pemrosesan bahasa alami, dan domain-domain lainnya (Huang, 2025).

7.1.1 Arsitektur Neural Networks Dasar

Jaringan neural buatan merupakan model komputasi yang terinspirasi oleh struktur biologis otak manusia. Setiap unit dalam jaringan, yang disebut neuron atau node, menerima input, melakukan komputasi, dan menghasilkan output. Dalam *deep learning*, jaringan-jaringan ini terdiri dari banyak lapisan yang saling terhubung, dengan setiap lapisan melakukan transformasi non-linear terhadap inputnya (Author, 2023). Lapisan-lapisan ini dapat mengekstrak fitur dari data pada berbagai tingkat abstraksi, dimulai

dari fitur sederhana di lapisan awal hingga fitur yang sangat abstrak di lapisan yang lebih dalam (Yamini et al., 2023).

7.1.2 Fungsi Aktivasi dan Propagasi Maju

Fungsi aktivasi memainkan peran krusial dalam neural networks dengan memperkenalkan non-linearitas, memungkinkan jaringan untuk belajar dari data yang tidak linear. Beberapa fungsi aktivasi yang umum digunakan termasuk *Rectified Linear Unit* (ReLU), *sigmoid*, dan *tanh* (Author, 2023). Proses propagasi maju melibatkan perhitungan output dari setiap lapisan berdasarkan input yang diterima dan bobot (weights) yang telah diinisialisasi. Bobot-bobot ini akan dioptimalkan selama proses pelatihan untuk meminimalkan kesalahan prediksi model (Yao et al., 2024).

7.1.3 Backpropagation dan Optimasi

Backpropagation adalah algoritma fundamental dalam pelatihan neural networks yang menghitung gradien dari fungsi kerugian (*loss function*) terhadap setiap bobot dalam jaringan. Gradien ini kemudian digunakan untuk memperbarui bobot dalam arah yang mengurangi kerugian. Algoritma optimasi seperti *Stochastic Gradient Descent* (SGD), Adam, dan RMSProp digunakan untuk menentukan laju dan arah pembaruan bobot yang optimal (Shen et al., 2024).

7.1.4 Regularisasi dan Generalisasi

Regularisasi merupakan teknik untuk mencegah *overfitting*, kondisi di mana model mempelajari pola spesifik dalam data pelatihan yang tidak generalisasi baik ke data baru. Teknik regularisasi termasuk *dropout*, *L1/L2 regularization*, dan *early*

stopping. *Dropout* secara acak "mematikan" beberapa neuron selama pelatihan, memaksa jaringan untuk belajar fitur yang lebih robust dan independen (Yamini et al., 2023).

7.2 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) telah menjadi arsitektur dominan dalam tugas-tugas *computer vision* sejak pencapaian revolusioner mereka dalam kompetisi ImageNet pada tahun 2012 (Krizhevsky et al., 2017). CNN dirancang khusus untuk memproses data grid seperti gambar dengan memanfaatkan prinsip *translational invariance* melalui operasi konvolusi (Li et al., 2021).

7.2.1 Operasi Konvolusi dan Pooling

Operasi konvolusi melibatkan penerapan filter (kernel) kecil pada input untuk mengekstrak fitur lokal. Setiap filter dapat dilihat sebagai detektor fitur yang belajar mengenali pola tertentu dalam data. Operasi pooling, biasanya *max pooling* atau *average pooling*, mengurangi dimensi spasial dari representasi feature map sambil mempertahankan informasi penting. Kombinasi operasi konvolusi dan pooling memungkinkan CNN untuk mengekstrak fitur hierarki dari gambar (Chen et al., 2021).

7.2.2 Arsitektur CNN Klasik dan Modern

Arsitektur CNN klasik seperti AlexNet, yang memenangkan ImageNet 2012, terdiri dari lima lapisan konvolusi yang diikuti oleh tiga lapisan fully-connected (Krizhevsky et al., 2017). Arsitektur

yang lebih baru dan lebih dalam seperti VGGNet, ResNet, dan Inception menunjukkan bahwa meningkatkan kedalaman jaringan dapat meningkatkan akurasi, namun memerlukan teknik khusus untuk mengatasi masalah seperti *vanishing gradients* (Simonyan & Zisserman, 2014). *ResNet* memperkenalkan koneksi skip (*skip connections*) yang memungkinkan gradien untuk mengalir lebih efisien melalui jaringan yang sangat dalam (Li et al., 2021).

7.2.3 Aplikasi CNN dalam Computer Vision

CNN telah diterapkan dengan sukses dalam berbagai tugas *computer vision* termasuk klasifikasi gambar, deteksi objek, segmentasi semantik, dan pengenalan wajah. Dalam klasifikasi gambar, model CNN yang dilatih pada dataset ImageNet yang besar dapat mencapai akurasi lebih dari 90% pada berbagai benchmark (Chen et al., 2021). Untuk deteksi objek, arsitektur seperti YOLO (*You Only Look Once*) dan Faster R-CNN telah menjadi standar industri yang mampu mendeteksi multiple objects dalam satu gambar secara real-time (Patel et al., 2022).

7.2.4 Efisiensi dan Kompresi Model CNN

Meskipun CNN sangat powerful, model-model ini sering kali memiliki jutaan parameter yang memerlukan komputasi dan memori yang signifikan. Teknik kompresi model seperti *network pruning*, *quantization*, dan *knowledge distillation* telah dikembangkan untuk mengurangi ukuran model sambil mempertahankan akurasi. *Network slimming*, misalnya, mengidentifikasi dan menghilangkan *channel-channel* yang kurang penting dalam jaringan, menghasilkan

pengurangan ukuran model hingga 20 kali dengan hanya sedikit penurunan akurasi (Liu et al., 2017).

7.3 Recurrent Neural Networks (RNN) dan LSTM

Sementara CNN dirancang untuk data grid seperti gambar, *Recurrent Neural Networks* (RNN) dirancang khusus untuk memproses data sekuensial. RNN mempertahankan hidden state yang diperbarui seiring pemrosesan setiap elemen dalam sekuens, memungkinkan jaringan untuk menyimpan informasi tentang elemen-elemen sebelumnya (Elshenawy et al., 2025).

7.3.1 Mekanisme Recurrence dan Temporal Dependencies

Mekanisme recurrence dalam RNN memungkinkan jaringan untuk memodelkan dependensi temporal dengan cara melewatkan informasi dari satu langkah waktu ke langkah berikutnya. Namun, RNN standar mengalami masalah *vanishing gradient problem*, di mana gradien menjadi sangat kecil saat dipropagasi kembali melalui waktu, membuat jaringan sulit belajar dependensi jangka panjang (Venkateswarlu et al., 2025).

7.3.2 Long Short-Term Memory (LSTM) Networks

Long Short-Term Memory (LSTM) adalah varian RNN yang memecahkan masalah *vanishing gradient* dengan memperkenalkan mekanisme *gating* yang memungkinkan informasi untuk mengalir langsung melalui waktu (Hajoui, 2025). LSTM memiliki tiga *gating mechanisms*: *input gate*, *forget gate*, dan *output gate*, yang

mengontrol aliran informasi. Struktur ini memungkinkan LSTM untuk mempelajari dependensi jangka panjang dengan lebih efektif dibandingkan RNN standar (Xu et al., 2018).

7.3.3 Gated Recurrent Unit (GRU) dan Varian Lainnya

Gated Recurrent Unit (GRU) adalah varian RNN lainnya yang menyederhanakan arsitektur LSTM dengan mengurangi jumlah gating mechanisms menjadi dua: *reset gate* dan *update gate* (Joshi, 2025). GRU memiliki kompleksitas komputasi yang lebih rendah dibandingkan LSTM namun sering kali mencapai kinerja yang sebanding. Varian lainnya seperti *Bidirectional LSTM* (BiLSTM) memproses sekuens dalam dua arah—maju dan mundur—memungkinkan jaringan untuk mengakses konteks dari kedua sisi (Duy & Srisongkram, 2025).

7.3.4 Aplikasi RNN dalam Natural Language Processing dan Time Series

RNN dan LSTM telah menjadi standar emas untuk tugas-tugas *Natural Language Processing* (NLP) seperti *machine translation*, *sentiment analysis*, dan *text generation*. Dalam machine translation, model sequence-to-sequence yang menggunakan encoder-decoder architecture dengan LSTM telah menghasilkan peningkatan signifikan dalam kualitas terjemahan (Elshenawy et al., 2025). Dalam *time series forecasting*, LSTM telah menunjukkan kemampuan yang superior dalam menangkap pola kompleks dan dependensi jangka panjang, mengalahkan model statistik tradisional seperti ARIMA (Rasheed et al., 2023).

7.4 Transfer Learning dan Pre-trained Models

Transfer learning adalah teknik di mana pengetahuan yang dipelajari dari satu tugas diaplikasikan untuk meningkatkan pembelajaran pada tugas lain. Dalam konteks deep learning, ini sering berarti menggunakan model yang telah dilatih pada dataset besar (seperti ImageNet) sebagai titik awal untuk tugas-tugas baru (Venkateswarlu et al., 2025).

7.4.1 Fine-tuning dan Feature Extraction

Ada dua pendekatan utama dalam transfer learning: *feature extraction* dan *fine-tuning*. Dalam *feature extraction*, lapisan-lapisan dari model yang sudah dilatih dibekukan dan digunakan untuk mengekstrak fitur dari data baru, dengan hanya lapisan klasifikasi final yang dilatih. Dalam *fine-tuning*, seluruh jaringan atau hanya beberapa lapisan terakhir dibuka untuk pelatihan ulang dengan data baru, biasanya dengan *learning rate* yang rendah untuk menjaga fitur yang telah dipelajari (Elshenawy et al., 2025).

7.4.2 Domain Adaptation dan Generalisasi

Tantangan utama dalam transfer learning adalah *domain gap*—perbedaan antara distribusi data pelatihan awal dan distribusi data target. Berbagai teknik telah dikembangkan untuk mengatasi masalah ini, termasuk *domain adversarial training*, *feature alignment*, dan *data augmentation* (Venkateswarlu et al., 2025). Penelitian menunjukkan bahwa transfer learning dapat secara signifikan mengurangi kebutuhan data label dan mempercepat konvergensi model (Elshenawy et al., 2025).

7.4.3 Pre-trained Models dalam Medical Imaging

Dalam domain medis, transfer learning telah menunjukkan hasil yang sangat menjanjikan. Model seperti ResNet, VGG, dan Inception yang dilatih pada ImageNet telah berhasil ditransfer ke tugas-tugas seperti deteksi kanker payudara, klasifikasi tumor otak, dan diagnosis penyakit mata, sering kali mencapai akurasi yang sebanding atau bahkan melampaui ahli manusia (Shin et al., 2016).

7.4.4 Multimodal Transfer Learning

Pengembangan terbaru dalam transfer learning melibatkan pembelajaran dari multiple modalities (gambar, teks, audio) secara bersamaan. Model-model seperti *CLIP* (Contrastive Language-Image Pre-training) menunjukkan bahwa pembelajaran bersama dari gambar dan teks dapat menghasilkan representasi yang sangat powerful dan transferable ke berbagai downstream tasks (Islam et al., 2023).

7.5 Generative Models dan Aplikasinya

Generative models adalah model deep learning yang dapat menghasilkan data baru yang mirip dengan data pelatihan. Dua jenis utama generative models adalah *Variational Autoencoders* (VAE) dan *Generative Adversarial Networks* (GAN) (Islam, 2025).

7.5.1 Generative Adversarial Networks (GANs)

GAN terdiri dari dua jaringan yang berkompetisi: generator yang menciptakan data palsu dan discriminator yang membedakan data asli dari yang palsu. Melalui proses pelatihan adversarial ini,

generator belajar untuk menghasilkan data yang semakin realistic (Islam, 2025). GANs telah berhasil diaplikasikan dalam berbagai tugas termasuk *image-to-image translation*, *style transfer*, *super-resolution*, dan *data augmentation*.

7.5.2 Synthetic Data Generation dan Privacy Preservation

GANs memiliki potensi besar dalam menghasilkan data sintesis berkualitas tinggi yang dapat digunakan untuk pelatihan model ketika data asli langka atau sensitif. Dalam domain medis, GANs dapat menghasilkan citra medis sintesis yang realistis tanpa membocorkan informasi pribadi pasien asli (Islam, 2025). Ini membuka peluang untuk berbagi data untuk penelitian sambil melindungi privasi pasien.

7.5.3 Conditional GANs dan Text-to-Image Synthesis

Conditional GANs (cGAN) memungkinkan kontrol atas data yang dihasilkan melalui kondisi input, seperti label kelas atau deskripsi teks. Model seperti DALL-E dan Midjourney menggunakan arsitektur serupa untuk menghasilkan gambar dari deskripsi teks, menunjukkan kemampuan luar biasa dari generative models dalam memahami dan mengeksekusi instruksi bahasa alami (Islam, 2025).

7.5.4 Challenges dalam Training GANs

Meskipun GANs sangat powerful, melatihnya adalah tantangan yang signifikan. Masalah seperti *mode collapse* (generator hanya belajar menghasilkan beberapa jenis *output*), *training instability*, dan *vanishing gradients* adalah hambatan umum. Penelitian berkelanjutan dalam *loss functions* baru, *architectural*

improvements, dan *training strategies* terus dikembangkan untuk mengatasi masalah-masalah ini (Islam, 2025).

7.6 Optimasi dan Training Deep Neural Networks

Pelatihan neural networks yang dalam memerlukan optimasi hati-hati dari berbagai aspek termasuk learning rate, batch size, dan hyperparameters lainnya (Shen et al., 2024).

7.6.1 Algoritma Optimasi Lanjutan

Selain SGD dasar, berbagai algoritma optimasi lanjutan telah dikembangkan untuk mempercepat konvergensi dan mencapai solusi yang lebih baik. *Adam* (*Adaptive Moment Estimation*) menggabungkan ide dari *momentum* dan *RMSProp*, secara adaptif menyesuaikan learning rate untuk setiap parameter (Jadhav et al., 2025). *AdaGrad*, *Adadelta*, dan *Nadam* adalah varian lainnya, masing-masing dengan keuntungan dan kekurangan tertentu (Al-Tameemi & Alzaghir, 2024).

7.6.2 Hyperparameter Tuning dan Automated Machine Learning

Pemilihan hyperparameter yang tepat (*learning rate*, *batch size*, *number of layers*, dll.) memiliki dampak besar pada kinerja model. Grid search dan random search adalah metode *brute-force* tradisional, sementara metode yang lebih sophisticated seperti *Bayesian optimization* dan *genetic algorithms* dapat menemukan hyperparameter optimal dengan lebih efisien (Wu et al., 2023).

AutoML mengotomatisasi seluruh proses pemilihan model dan hyperparameter tuning (Shen et al., 2024).

7.6.3 Data Augmentation dan Preprocessing

Data augmentation adalah teknik untuk memperluas dataset pelatihan dengan membuat variasi dari data asli melalui transformasi seperti rotasi, *flipping*, *cropping*, dan *color jittering*. Untuk NLP, augmentation dapat melibatkan *paraphrasing*, *synonym replacement*, dan *back-translation* (Li et al., 2021). Preprocessing yang efektif termasuk normalisasi, *handling missing values*, dan *balancing class distribution* adalah kritis untuk training yang sukses (Shen et al., 2024).

7.6.4 Distributed Training dan Hardware Acceleration

Pelatihan model *deep learning* yang besar sering memerlukan distributed training di atas multiple GPUs atau TPUs. Teknik seperti *data parallelism* dan *model parallelism* memungkinkan pembagian komputasi across multiple devices (Shen et al., 2024). *Framework* seperti *TensorFlow*, *PyTorch*, dan *JAX* menyediakan abstractions untuk distributed training yang efisien.

7.7 Aplikasi Deep Learning dalam Domain-Domain Praktis

Deep learning telah merevolusionisasi berbagai bidang dengan menghasilkan breakthrough dalam tugas-tugas yang sebelumnya dianggap sulit untuk diotomatisasi (Wang, 2024).

7.7.1 Medical Imaging dan Diagnosis

Dalam medical imaging, deep learning telah mencapai atau melampaui kinerja radiolog manusia dalam berbagai tugas. CNN dapat mendeteksi lesi kanker dalam mammogram, tumor otak dalam MRI, dan anomali dalam chest X-rays dengan akurasi yang tinggi. Model yang dilatih khusus untuk tugas spesifik seperti deteksi tumor otak mencapai akurasi 93.3% dengan AUC 98.43%, melampaui banyak radiolog (Mahmud et al., 2023).

7.7.2 Natural Language Processing dan Speech Recognition

Transformers dan model *large language models* (LLMs) seperti BERT, GPT-3, dan GPT-4 telah merevolusi NLP, mencapai kinerja state-of-the-art dalam berbagai tugas termasuk *machine translation*, *question answering*, *sentiment analysis*, dan *text generation* (Islam et al., 2023). Dalam *speech recognition*, *deep learning* telah meningkatkan akurasi secara dramatis dan memungkinkan sistem seperti Alexa, Siri, dan Google Assistant (Elshenawy et al., 2025).

7.7.3 Autonomous Vehicles dan Robotics

Deep learning adalah komponen kritis dalam pengembangan kendaraan otonom, memungkinkan sistem untuk mengenali pedestrians, lane markings, traffic signs, dan obstacle (Elshenawy et al., 2025). Dalam robotics, CNN dan RNN digunakan untuk memproses visual input dan membuat keputusan navigasi real-time (Wang, 2024).

7.7.4 Time Series Forecasting dan Anomaly Detection

LSTM dan hybrid models yang menggabungkan CNN dan RNN telah menunjukkan superior performance dalam *time series forecasting* untuk aplikasi seperti stock price prediction, energy load forecasting, dan weather prediction. Model hybrid CNN-LSTM khususnya dapat menangkap baik spatial patterns maupun temporal dependencies, menghasilkan peningkatan RMSE hingga 49% dibandingkan model baseline (Khan et al., 2026).

7.8 Tantangan dan Arah Masa Depan dalam Deep Learning

Meskipun deep learning telah mencapai kesuksesan yang luar biasa, masih banyak tantangan yang perlu ditangani dan arah-arahan penelitian yang menjanjikan untuk eksplorasi (Elshenawy et al., 2025).

7.8.1 Interpretability dan Explainability

Salah satu kritik utama terhadap deep learning adalah sifatnya yang "*black-box*"—sulit untuk memahami mengapa model membuat prediksi tertentu. Ini adalah masalah kritis dalam aplikasi *high-stakes* seperti *medical diagnosis* dan *criminal justice*. Teknik seperti LIME (*Local Interpretable Model-agnostic Explanations*), SHAP (*SHapley Additive exPlanations*), dan *attention visualization* sedang dikembangkan untuk meningkatkan interpretability (Joshi et al., 2021).

7.8.2 Data Efficiency dan Few-Shot Learning

Model deep learning tradisional memerlukan dataset besar yang berlabel untuk mencapai kinerja yang baik. Penelitian dalam *few-shot learning*, *meta-learning*, dan *self-supervised learning* bertujuan untuk membuat model yang dapat belajar dari dataset yang lebih kecil atau dari data yang tidak berlabel (Gui et al., 2024).

7.8.3 Robustness dan Adversarial Attacks

Model *deep learning* telah ditunjukkan rentan terhadap *adversarial examples*—input yang dimodifikasi secara halus yang dapat membuat model membuat prediksi yang sangat berbeda. Penelitian dalam *adversarial training* dan *certified robustness* sedang berkembang untuk membuat model yang lebih robust (Noor & Ige, 2024).

7.8.4 Efisiensi Energi dan Edge Deployment

Pelatihan dan inference dari model deep learning yang besar memerlukan energi yang signifikan. Dengan meningkatnya *concern* terhadap *climate change*, penelitian dalam *efficient neural networks*, *model compression*, dan *hardware-aware optimization* menjadi semakin penting untuk deployment model pada edge devices (Noor & Ige, 2024).

7.9 Latihan Soal

1. Jelaskan perbedaan fundamental antara CNN dan RNN. Dalam konteks apa setiap arsitektur lebih cocok digunakan, dan berikan contoh aplikasi praktis untuk masing-masing.

2. Diskusikan mekanisme *gating* dalam LSTM dan jelaskan bagaimana hal ini memecahkan masalah *vanishing gradient problem* dalam RNN standar.
3. Analisis keuntungan dan kerugian dari *transfer learning*. Dalam skenario apa transfer learning sangat bermanfaat, dan kapan training dari scratch mungkin lebih baik?
4. Jelaskan konsep GAN dan bagaimana proses pelatihan adversarial antara generator dan discriminator bekerja untuk menghasilkan data sintesis.
5. Diskusikan tantangan utama dalam deployment deep learning models pada production systems, termasuk masalah interpretability, robustness, dan efisiensi komputasi.

Bab 8: Sistem Cerdas dan Multi Agen

8.1 Konsep Dasar Sistem Cerdas dan Arsitektur Multi Agen

8.1.1 Definisi Sistem Cerdas dan Agen Otonom

Sistem cerdas adalah sistem komputer yang dapat memproses informasi, belajar dari pengalaman, dan membuat keputusan untuk mencapai tujuan tertentu dengan tingkat otonomi yang tinggi (Yan et al., 2025). Dalam sistem ini, agen otonom (*autonomous agent*) adalah entitas perangkat lunak atau fisik yang dapat bertindak secara independen berdasarkan persepsi lingkungan mereka, tanpa memerlukan kontrol atau intervensi manusia yang konstan (Das et al., 2025). Setiap agen memiliki tujuan individual atau bersama, kemampuan untuk mengambil keputusan, dan kapasitas untuk berkomunikasi dengan agen-agen lain. Dalam konteks sistem multi-agen, kolaborasi antar agen menjadi kunci untuk mencapai tujuan kolektif yang kompleks dan tidak dapat diselesaikan oleh agen tunggal (Tarafdar, 2025).

8.1.2 Karakteristik dan Properti Agen Cerdas

Agan cerdas memiliki beberapa karakteristik fundamental yang membedakannya dari sistem tradisional (Das et al., 2025). Pertama, otonomi (*autonomy*) adalah kemampuan agen untuk

beroperasi tanpa kontrol berkelanjutan dari pengguna eksternal atau agen lain, membuat keputusan berdasarkan pengetahuan dan persepsi lokal mereka. Kedua, responsivitas (*responsiveness*) berarti agen dapat merespons perubahan lingkungan dengan cepat dan tepat. Ketiga, proaktivitas (*proactivity*) adalah kemampuan agen untuk mengambil inisiatif dan mengejar tujuan mereka secara aktif daripada hanya bereaksi terhadap peristiwa eksternal. Keempat, kemampuan komunikasi memungkinkan agen untuk berbagi informasi dan bernegosiasi dengan agen-agen lain (Yan et al., 2025). Kelima, kapasitas pembelajaran (*learning capacity*) memungkinkan agen untuk meningkatkan kinerjanya seiring waktu berdasarkan pengalaman dan umpan balik dari lingkungan.

8.1.3 Arsitektur Sistem Multi Agen dan Topologi Jaringan

Arsitektur sistem multi-agen menggambarkan bagaimana agen-agen diorganisir dan berinteraksi satu sama lain (Das et al., 2025). Ada beberapa jenis topologi jaringan yang umum digunakan dalam sistem multi-agen. Topologi terpusat (*centralized*) melibatkan agen kontrol pusat yang mengkoordinasikan semua agen lain, menawarkan kontrol yang kuat tetapi kurang scalable dan rentan terhadap kegagalan titik tunggal. Topologi terdistribusi (*distributed*) memungkinkan agen untuk mengambil keputusan secara independen berdasarkan informasi lokal dan komunikasi peer-to-peer, menawarkan skalabilitas dan ketahanan yang lebih baik (Tarafdar, 2025). Topologi hierarki (*hierarchical*) menggabungkan fitur dari keduanya, dengan agen diorganisir dalam tingkat-tingkat, memungkinkan delegasi dan koordinasi pada

berbagai skala (Das et al., 2025). Setiap topologi memiliki kelebihan dan kekurangan tergantung pada aplikasi spesifik dan batasan sistem.

8.1.4 Peran dan Fungsi Agen dalam Sistem Kolaboratif

Dalam sistem multi-agen kolaboratif, setiap agen memiliki peran spesifik yang berkontribusi pada keberhasilan keseluruhan sistem (Hu et al., 2023). Beberapa agen dapat berfungsi sebagai koordinator, memfasilitasi komunikasi dan alokasi sumber daya antar agen lain. Agen lain dapat berperan sebagai spesialis, menangani tugas-tugas tertentu yang memerlukan keahlian khusus. Ada juga agen yang berperan sebagai monitor, melacak keadaan sistem dan mendeteksi anomali atau kegagalan (Yan et al., 2025). Peran agen dapat bersifat statis atau dinamis, tergantung pada kebutuhan sistem dan kondisi lingkungan. Pengalokasian peran yang efektif adalah kunci untuk mencapai kolaborasi yang efisien dan mencegah redundansi atau konflik dalam sistem multi-agen (Hu et al., 2023).

8.2 Mekanisme Komunikasi dan Koordinasi Antar Agen

8.2.1 Protokol Komunikasi dalam Sistem Multi Agen

Protokol komunikasi adalah aturan dan standar yang mengatur bagaimana agen berkomunikasi satu sama lain (Chen et al., 2026). Protokol komunikasi tradisional dalam sistem multi-agen mencakup pertukaran pesan (*message passing*), di mana agen

mengirim dan menerima pesan untuk berbagi informasi atau membuat permintaan. Protokol ini dapat dirancang khusus untuk aplikasi tertentu atau mengadopsi standar industri yang ada. Beberapa protokol komunikasi modern termasuk Model Context Protocol (*MCP*), yang menyediakan interface berbasis JSON-RPC untuk integrasi alat dan pertukaran data yang aman, dan Agent Communication Protocol (*ACP*), yang mendefinisikan protokol komunikasi tujuan umum melalui HTTP RESTful (Ehtesham et al., 2025). Agent-to-Agent Protocol (*A2A*) memungkinkan delegasi tugas peer-to-peer menggunakan Agent Cards berbasis kapabilitas, sementara Agent Network Protocol (*ANP*) mendukung penemuan agen terbuka dan kolaborasi aman menggunakan identifikasi terdesentralisasi (Chang et al., 2025).

8.2.2 Strategi Koordinasi dan Negosiasi Antar Agen

Strategi koordinasi dalam sistem multi-agen merujuk pada mekanisme yang digunakan untuk menyelaraskan tindakan agen-agen dan mencapai tujuan kolektif (Yan et al., 2025). Negosiasi adalah salah satu strategi koordinasi yang paling umum, di mana agen berinteraksi melalui serangkaian pesan untuk mencapai kesepakatan atau keputusan bersama. Protokol negosiasi dapat berkisar dari yang sederhana seperti pengambilan suara mayoritas hingga yang kompleks seperti negosiasi multi-putaran dengan konsesi bertahap (Bolleddu, 2025). Pendekatan berbasis teori permainan menggunakan model matematika untuk memprediksi dan mengoptimalkan strategi agen dalam situasi kompetitif atau kooperatif (Jin et al., 2025). Mekanisme berbasis pasar (*market-*

based) memungkinkan agen untuk menegosiasikan alokasi sumber daya melalui sistem penawaran atau lelang, menciptakan keseimbangan yang efisien (Bhende et al., 2025).

8.2.3 Komunikasi Berbasis Pembelajaran dan Protokol Mereka

Dalam sistem multi-agen modern, agen dapat belajar untuk mengembangkan protokol komunikasi mereka sendiri melalui pembelajaran mesin (*machine learning*) dan pembelajaran penguatan (*reinforcement learning*) (Yan et al., 2025). Komunikasi yang muncul (*emergent communication*) adalah fenomena di mana agen belajar menggunakan sinyal atau bahasa terstruktur untuk meningkatkan koordinasi mereka, tanpa diprogramkan secara eksplisit (Chafii et al., 2023). Pendekatan berbasis pembelajaran mendalam memungkinkan agen untuk mempelajari representasi pesan yang efisien dan strategi komunikasi yang optimal. Namun, tantangan dalam pembelajaran komunikasi multi-agen termasuk penyelarasan arti pesan, penanganan komunikasi parsial, dan skalabilitas ke sistem besar dengan banyak agen (Chen et al., 2026).

8.2.4 Mekanisme Konsensus dan Pengambilan Keputusan Kolektif

Mekanisme konsensus adalah protokol yang memungkinkan kelompok agen untuk mencapai perjanjian tentang keputusan atau status sistem bahkan ketika beberapa agen mungkin gagal atau berperilaku tidak jujur (Hsiao & Chen, 2019). Protokol konsensus tradisional seperti voting mayoritas sederhana namun rentan terhadap manipulasi. Protokol Byzantium Fault Tolerant (*Byzantine Fault Tolerant*) lebih robust, mampu mencapai konsensus bahkan

ketika beberapa peserta berperilaku tidak jujur, menjadikannya cocok untuk sistem yang tidak tepercaya (Arabambi et al., 2025). Pengambilan keputusan kolektif dalam sistem multi-agen juga dapat didukung oleh mekanisme kepercayaan (*trust mechanisms*), di mana agen mengevaluasi reputasi agen lain berdasarkan interaksi sebelumnya dan menggunakan informasi ini untuk memandu keputusan mereka (Pan et al., 2025).

8.3 Pembelajaran Penguatan Multi Agen dan Kolaborasi

8.3.1 Fondasi Multi-Agent Reinforcement Learning

Pembelajaran penguatan multi-agen (*Multi-Agent Reinforcement Learning* atau MARL) adalah area penelitian yang fokus pada bagaimana agen dapat belajar kebijakan optimal melalui interaksi dengan lingkungan dan agen lain (Azadeh, 2024). Berbeda dengan pembelajaran penguatan agen tunggal, MARL menghadapi tantangan unik seperti non-stationaritas (kebijakan agen lain terus berubah), kutukan dimensionalitas (ruang status-aksi gabungan berkembang secara eksponensial), dan kebutuhan untuk menyeimbangkan eksplorasi individu dengan koordinasi kolektif (Guresti & Ure, 2021). Paradigma pembelajaran terpusat dengan eksekusi terdesentralisasi (Centralized Training with Decentralized Execution atau CTDE) adalah pendekatan populer yang memungkinkan agen untuk belajar kolaboratif selama pelatihan

sambil beroperasi secara independen selama eksekusi (Chen et al., 2022).

8.3.2 Metode Dekomposisi Nilai dan Koordinasi Eksplisit

Dekomposisi nilai (*value decomposition*) adalah teknik fundamental dalam MARL yang memecah nilai gabungan (*global value*) menjadi nilai individu agen dengan cara yang memastikan konsistensi (Chen et al., 2022). Metode VDN (*Value Decomposition Networks*) menggunakan penjumlahan monoton untuk mendekomposisi nilai global, sedangkan QMIX (*Quotient Multi-Agent Policy Gradient*) menggunakan fungsi mixing non-linier untuk menangani interaksi non-monoton antar agen (Gupta et al., 2025). Koordinasi eksplisit melibatkan pembelajaran mekanisme komunikasi yang memungkinkan agen untuk berbagi informasi tentang keadaan, intensi, atau rencana mereka (Lei et al., 2025). Teknik-teknik ini telah terbukti meningkatkan kinerja kolaborasi pada tugas-tugas kompleks, khususnya dalam lingkungan dengan observasi parsial dan komunikasi terbatas.

8.3.3 Pembelajaran Hierarki dan Aglomerasi Peran

Pembelajaran hierarki dalam MARL mengorganisir pembelajaran pada berbagai tingkat abstraksi, dari tugas tingkat tinggi hingga tindakan tingkat rendah (Salt et al., 2025). Agen dapat belajar subtask atau skill yang dapat dikombinasikan untuk menyelesaikan tugas yang lebih kompleks. Pembelajaran peran (*role learning*) adalah pendekatan di mana agen menemukan peran yang sesuai berdasarkan kemampuan dan kondisi mereka, kemudian mengoptimalkan kebijakan mereka untuk peran tersebut (Hu et al.,

2023). Pendekatan berbasis aglomerasi memungkinkan agen untuk dikelompokkan berdasarkan peran atau kesamaan perilaku mereka, mengurangi kompleksitas pembelajaran dengan memfokuskan koordinasi pada agen yang relevan (Chen et al., 2024).

8.3.4 Tantangan Skalabilitas dan Konvergensi dalam MARL

Salah satu tantangan terbesar dalam MARL adalah skalabilitas: ketika jumlah agen meningkat, ruang status dan aksi tumbuh secara eksponensial, membuat pembelajaran menjadi impraktis (Azadeh, 2024). Non-stationaritas adalah tantangan lain, di mana kebijakan agen-agen lain terus berubah selama pelatihan, membuat lanskap pembelajaran tidak stabil dan sulit untuk prediksi (Yamaguchi, 2025). Konvergensi ke solusi suboptimal atau titik sadel adalah masalah umum dalam MARL, terutama ketika agen hanya memiliki observasi parsial tentang keadaan sistem (Wang et al., 2023). Penelitian terbaru menunjukkan bahwa mekanisme seperti komunikasi yang dipelajari, reward intrinsik, dan hierarki dapat membantu mengatasi tantangan-tantangan ini, namun solusi umum masih merupakan area penelitian terbuka.

8.4 Aplikasi Praktis Sistem Multi Agen dalam Berbagai Domain

8.4.1 Sistem Multi Agen dalam Robotika dan Otomasi

Sistem multi-agen memiliki aplikasi yang kaya dalam robotika, di mana kelompok robot harus bekerja sama untuk menyelesaikan tugas seperti pencarian dan penyelamatan, pemetaan,

atau konstruksi kolektif (Devi et al., 2024). Kawanan robot (*robotic swarms*) mengadopsi prinsip-prinsip dari sistem biologis seperti perilaku serangga sosial, di mana interaksi lokal sederhana menghasilkan perilaku kolektif yang kompleks dan adaptif (Devi et al., 2024). Aplikasi praktis mencakup drone berkaliber untuk pemantauan lingkungan, robot mobile kolaboratif untuk manufaktur, dan sistem otomasi terproduksi untuk pabrik pintar (*smart factories*) (Salt et al., 2025). Tantangan dalam robotika multi-agen termasuk sinkronisasi waktu, penanganan kegagalan hardware, dan keselamatan dalam lingkungan yang bersama dengan manusia.

8.4.2 Sistem Multi Agen dalam Transportasi dan Kendaraan Otonom

Dalam domain transportasi, sistem multi-agen digunakan untuk mengoordinasikan kendaraan otonom, manajemen traffic, dan optimisasi rute (Hu et al., 2024). Kendaraan yang terhubung dan otonom (*Connected and Autonomous Vehicles* atau *CAVs*) dapat berkomunikasi satu sama lain untuk meningkatkan keselamatan jalan, efisiensi lalu lintas, dan kenyamanan penumpang (Cui et al., 2025). Protokol komunikasi berbasis bahasa alami memungkinkan kendaraan otonom untuk mengkomunikasikan intensi mereka dengan cara yang dapat dipahami oleh pengguna manusia dan kendaraan lain. Sistem multi-agen juga diterapkan dalam manajemen ramp merging, di mana koordinasi multi-agen membantu mengurangi kemacetan dan meningkatkan throughput kendaraan (Pan et al., 2025).

8.4.3 Sistem Multi Agen dalam Jaringan dan Manajemen Sumber Daya

Sistem multi-agen digunakan dalam jaringan komputer dan IoT untuk mengelola sumber daya secara terdesentralisasi dan efisien (G et al., 2025). Dalam jaringan energi pintar (*smart grids*), agen mewakili pembangkit daya, penyimpanan, dan konsumen, bernegosiasi untuk mengoptimalkan aliran energi dan meminimalkan biaya operasional (Ali et al., 2025). Sistem multi-agen juga diterapkan dalam manajemen spektrum nirkabel, di mana agen radio-kognitif dapat belajar untuk mendeteksi dan memanfaatkan pita spektrum yang tidak digunakan secara dinamis (Khan & Nayak, 2025). Dalam konteks cloud computing dan sistem terdistribusi, agen multi-agen dapat mengelola alokasi sumber daya, penyeimbangan beban, dan toleransi kesalahan dengan cara yang lebih adaptif dan responsif daripada sistem terpusat.

8.4.4 Sistem Multi Agen dalam Aplikasi Kesehatan dan Manajemen Krisis

Aplikasi sistem multi-agen dalam kesehatan mencakup sistem diagnosis terintegrasi di mana agen spesialis mengevaluasi aspek berbeda dari kondisi pasien dan berkolaborasi untuk membuat keputusan klinis (Toader et al., 2018). Dalam manajemen bencana, sistem multi-agen dapat mengkoordinasikan upaya penyelamatan dengan menugaskan tim-tim (masing-masing diwakili oleh agen) ke lokasi-lokasi yang berbeda berdasarkan prioritas dan ketersediaan sumber daya (Rasouli, 2018). Sistem multi-agen juga digunakan untuk pemantauan kesehatan masyarakat, di mana agen

mengumpulkan data dari berbagai sumber dan berkolaborasi untuk mengidentifikasi pola penyakit dan merekomendasikan intervensi kesehatan masyarakat.

8.5 Tantangan dan Arah Masa Depan dalam Sistem Multi Agen

8.5.1 Tantangan Teknis dan Skalabilitas

Tantangan teknis utama dalam sistem multi-agen mencakup manajemen komunikasi yang efisien, sinkronisasi terdistribusi, dan toleransi kesalahan (Uddin et al., 2025). Ketika sistem tumbuh besar dengan ribuan atau jutaan agen, overhead komunikasi dapat menjadi hambatan serius, dan mekanisme pengurangan komunikasi (*communication reduction*) diperlukan (Yan et al., 2025). Skalabilitas algoritma pembelajaran juga tetap menjadi tantangan, karena metode pembelajaran terpusat mungkin tidak praktis untuk sistem besar. Penelitian dalam pembelajaran yang dapat ditransfer dan generalisasi lintas skala agen sedang dilakukan untuk mengatasi masalah ini (Guresti & Ure, 2021).

8.5.2 Keamanan, Privasi, dan Kepercayaan dalam Sistem Multi Agen

Keamanan multi-agen adalah kepedulian kritis, terutama ketika agen berinteraksi dalam lingkungan yang tidak tepercaya (*untrusted environments*) (Shukla et al., 2018). Agen dapat menjadi sasaran serangan cyber, dan protokol komunikasi harus dienkripsi dan diautentikasi untuk mencegah intersepsi atau manipulasi

pesan (Chang et al., 2025). Privasi adalah masalah penting ketika agen berbagi informasi sensitif, dan teknik seperti pembelajaran terdistribusi dan pembelajaran federal (*federated learning*) digunakan untuk melindungi privasi data lokal (Yang et al., 2025). Mekanisme kepercayaan berbasis reputasi dan blockchain dapat digunakan untuk memverifikasi integritas interaksi agen dan mencegah perilaku buruk (Calvaresi et al., 2018).

8.5.3 Integrasi dengan Large Language Models dan AI Generatif

Perkembangan terbaru dalam penelitian sistem multi-agen adalah integrasi dengan *Large Language Models (LLM)* dan AI generatif (Ferrag et al., 2025). LLM dapat memberikan kemampuan penalaran dan perencanaan yang kuat kepada agen, memungkinkan mereka untuk menangani tugas-tugas kompleks yang memerlukan pemahaman konteks mendalam (Liu et al., 2025). Agen berbasis LLM dapat menggunakan bahasa alami untuk berkomunikasi dengan manusia dan agen lain, meningkatkan interpretabilitas dan kemudahan penggunaan (Sun et al., 2025). Namun, mengintegrasikan LLM dengan pembelajaran penguatan dan sistem multi-agen membawa tantangan baru dalam hal efisiensi komputasi, bias model, dan penyelarasan dengan nilai manusia (Liu et al., 2023).

8.5.4 Penelitian Terbuka dan Arah Masa Depan

Arah penelitian masa depan dalam sistem multi-agen mencakup pengembangan mekanisme komunikasi yang lebih efisien dan scalable, algoritma pembelajaran yang lebih robust terhadap non-stationaritas dan bias, dan kerangka kerja yang

mengintegrasikan pembelajaran, bahasa alami, dan kontrol untuk kolaborasi multi-agen yang lebih baik (Jin et al., 2025). Penelitian tentang *emergent behavior*, di mana perilaku sistem secara keseluruhan muncul dari interaksi lokal agen, terus mendapatkan perhatian dan menawarkan wawasan untuk sistem yang lebih adaptif dan resilient (March-Pons et al., 2025). Fokus pada keselamatan (*safety*) dan robustness dalam sistem multi-agen, terutama dalam aplikasi kritis seperti autonomous driving dan robotika, akan menjadi area penelitian yang semakin penting di masa depan (Tarafdar, 2025).

8.6 Latihan Soal

1. Jelaskan perbedaan antara topologi komunikasi terpusat, terdistribusi, dan hierarki dalam sistem multi-agen. Untuk setiap topologi, berikan satu contoh aplikasi praktis dan diskusikan kelebihan serta kekurangan dari masing-masing topologi dalam konteks sistem skala besar.
2. Dalam pembelajaran penguatan multi-agen (*Multi-Agent Reinforcement Learning*), jelaskan konsep "Centralized Training with Decentralized Execution" (CTDE). Mengapa pendekatan ini penting dalam sistem multi-agen? Berikan satu studi kasus di mana CTDE diterapkan dan bagaimana hasilnya dibandingkan dengan pendekatan alternatif.
3. Diskusikan tantangan utama dalam pengembangan protokol komunikasi untuk sistem multi-agen yang memiliki ratusan atau

ribuan agen. Bagaimana teknik seperti "komunikasi yang muncul" (*emergent communication*) dan pembelajaran protokol (*learned protocols*) dapat membantu mengatasi tantangan-tantangan ini?

4. Jelaskan bagaimana integrasi *Large Language Models* (LLM) dengan sistem multi-agen dapat meningkatkan kemampuan agen dalam hal reasoning, perencanaan, dan komunikasi dengan manusia. Apa tantangan teknis dan etis yang perlu dipertimbangkan dalam integrasi ini?
5. Dalam konteks manajemen bencana atau operasi pencarian-penyelamatan, desain arsitektur sistem multi-agen yang dapat mengkoordinasikan tim-tim respons di berbagai lokasi. Jelaskan bagaimana sistem ini menangani komunikasi, alokasi sumber daya, dan pengambilan keputusan kolektif di bawah kondisi ketidakpastian dan informasi yang terbatas.

Bab 9: Integrasi Artificial Intelligence dalam Sistem Komputasi

Integrasi kecerdasan buatan (*artificial intelligence* atau AI) ke dalam sistem komputasi modern telah menjadi salah satu transformasi teknologi yang paling signifikan di era digital ini. Konvergensi antara AI dengan infrastruktur komputasi yang terus berkembang menciptakan kemampuan komputasi yang belum pernah ada sebelumnya, memungkinkan pemrosesan data dalam skala besar, pengambilan keputusan real-time, dan otomasi sistem yang kompleks. Integrasi ini tidak hanya terbatas pada aspek teknis, tetapi juga mencakup dimensi arsitektur sistem, manajemen sumber daya, keamanan, dan efisiensi energi yang semakin kritis dalam konteks keberlanjutan lingkungan dan ekonomi (Kumar, 2025).

9.1 Fundamentasi Arsitektur Sistem Komputasi Berbasis AI

Arsitektur sistem komputasi modern yang mengintegrasikan AI telah mengalami evolusi signifikan dari model terpusat tradisional menuju ekosistem komputasi yang terdistribusi dan heterogen. Sistem komputasi berbasis AI saat ini menggabungkan

berbagai paradigma komputasi yang saling melengkapi untuk mencapai performa optimal dan efisiensi dalam penggunaan sumber daya.

9.1.1 Model Komputasi Cloud, Edge, dan Fog

Transformasi dari arsitektur komputasi tradisional menuju model yang terintegrasi dengan baik dimulai dengan pemahaman mendalam tentang tiga paradigma utama: *cloud computing*, *edge computing*, dan *fog computing*. *Cloud computing* menyediakan infrastruktur komputasi terpusat yang sangat skalabel dengan kapasitas penyimpanan dan pemrosesan yang hampir tidak terbatas, namun dengan latensi yang lebih tinggi dibandingkan dengan pemrosesan lokal. Sebaliknya, *edge computing* membawa kapabilitas komputasi lebih dekat ke sumber data, memungkinkan pemrosesan data dengan latensi ultra-rendah yang penting untuk aplikasi *real-time* seperti kendaraan otonom dan sistem kontrol industri (Chennupati, 2025).

Fog computing berperan sebagai lapisan perantara yang menghubungkan *edge* dan *cloud*, menyediakan pemrosesan tingkat menengah untuk tugas-tugas yang memerlukan sumber daya lebih besar daripada yang tersedia di *edge* tetapi masih memerlukan respons yang cepat. Integrasi ketiga paradigma ini menciptakan apa yang dikenal sebagai *edge-cloud continuum*, di mana pekerjaan AI dapat didistribusikan secara dinamis berdasarkan kebutuhan latensi, bandwidth, dan kapabilitas komputasi (Murthy et al., 2025).

9.1.2 Arsitektur Layering dan Orkestrasi Sistem

Arsitektur modern sistem komputasi berbasis AI mengadopsi pendekatan *layering* yang jelas untuk mengelola kompleksitas sistem yang terus meningkat. Lapisan pertama adalah *perception layer* yang terdiri dari perangkat IoT (*Internet of Things*) dan sensor yang mengumpulkan data real-time dari lingkungan fisik. Lapisan kedua adalah *edge intelligence layer* di mana model AI yang dioptimalkan beroperasi untuk melakukan inferensi lokal dan pengambilan keputusan awal. Lapisan ketiga adalah *cloud intelligence layer* yang menangani proses pelatihan model, analitik mendalam, dan penyimpanan data jangka panjang (Rane et al., 2024).

Orkestrasi sistem yang efektif memerlukan mekanisme otomatis untuk mengarahkan beban kerja AI ke tingkat komputasi yang paling sesuai berdasarkan faktor-faktor seperti ketersediaan sumber daya, kebutuhan latensi, dan efisiensi energi. Teknologi *containerization* seperti Docker dan *orchestration frameworks* seperti Kubernetes memainkan peran penting dalam memfasilitasi deployment yang fleksibel dan skalabel dari aplikasi berbasis AI di seluruh infrastruktur komputasi yang heterogen (Bershchansky et al., 2024).

9.1.3 Arsitektur Serverless dan Function-as-a-Service

Tren terbaru dalam arsitektur komputasi berbasis AI adalah adopsi model *serverless computing* atau *Function-as-a-Service* (FaaS), di mana pengembang dapat menulis dan menjalankan fungsi AI tanpa perlu mengelola infrastruktur underlying secara langsung.

Model ini sangat berguna untuk aplikasi AI yang *bersifat event-driven* atau memiliki beban kerja yang bervariasi secara signifikan. Dengan menggunakan *serverless architecture*, organisasi dapat mencapai skalabilitas otomatis, biaya operasional yang lebih rendah, dan pengembangan aplikasi yang lebih cepat (Nangi & Settipi, 2023).

9.1.4 Integrasi Quantum Computing dalam Infrastruktur AI

Meskipun masih berada di tahap awal, integrasi *quantum computing* dengan sistem AI tradisional mulai menunjukkan potensi untuk menyelesaikan masalah optimisasi yang sangat kompleks dengan kecepatan yang jauh lebih tinggi daripada komputer klasik. *Quantum computing* dapat diintegrasikan sebagai akselerator khusus untuk tugas-tugas tertentu seperti optimisasi parameter dalam model pembelajaran mesin besar (Rane et al., 2024).

9.2 Manajemen Sumber Daya dan Optimisasi Komputasi

Manajemen sumber daya yang efisien merupakan aspek krusial dalam sistem komputasi berbasis AI, terutama mengingat tuntutan komputasi yang terus meningkat dari model AI yang semakin kompleks dan ukuran dataset yang semakin besar. Strategi manajemen sumber daya yang tepat dapat menghasilkan peningkatan performa yang signifikan sambil mengurangi biaya operasional dan konsumsi energi.

9.2.1 Alokasi Sumber Daya Dinamis dan Predictive Scaling

Sistem modern untuk manajemen sumber daya menggunakan algoritma *machine learning* untuk memprediksi pola beban kerja dan secara proaktif mengalokasikan sumber daya sebelum lonjakan permintaan terjadi. Predictive auto-scaling menggunakan teknik seperti ARIMA (*Auto Regressive Integrated Moving Average*) dan LSTM (*Long Short-Term Memory*) untuk meramalkan kebutuhan sumber daya di masa depan, memungkinkan sistem untuk melakukan penskalaan otomatis dengan lebih efisien (Kamau & Myllynen, 2025).

Alokasi sumber daya dinamis juga melibatkan pengambilan keputusan real-time tentang bagaimana mendistribusikan beban kerja AI di antara berbagai node komputasi dengan karakteristik yang berbeda. Pendekatan berbasis pembelajaran mesin reinforcement learning dapat digunakan untuk mengoptimalkan keputusan alokasi berdasarkan pengalaman sistem yang terakumulasi (Kanungo, 2024).

9.2.2 Teknik Kompresi Model dan Optimisasi Inferensi

Untuk mengatasi keterbatasan sumber daya pada perangkat *edge* dan IoT, berbagai teknik kompresi model telah dikembangkan untuk mengurangi ukuran model tanpa mengorbankan akurasi secara signifikan. Teknik-teknik ini mencakup *pruning* (menghapus parameter model yang tidak signifikan), *quantization* (mengurangi presisi numerik parameter), dan *knowledge distillation* (transfer pengetahuan dari model besar ke model yang lebih kecil) (Dantas et al., 2024).

Quantization adalah salah satu teknik yang paling efektif, memungkinkan pengurangan ukuran model hingga 50-75% dengan penurunan akurasi yang minimal. Teknik *mixed-precision quantization* memungkinkan penggunaan presisi numerik yang berbeda untuk bagian yang berbeda dari model, mencapai keseimbangan optimal antara ukuran model dan akurasi (Duanmu et al., 2025).

9.2.3 Scheduling Beban Kerja dan Load Balancing

Penjadwalan beban kerja (*workload scheduling*) merupakan fungsi kritis dalam sistem komputasi terdistribusi berbasis AI. Algoritma penjadwalan yang cerdas harus mempertimbangkan berbagai faktor seperti latensi jaringan, kapabilitas komputasi yang berbeda-beda di setiap node, ketersediaan sumber daya, dan persyaratan *Quality of Service* (QoS) dari aplikasi individual (Sakthidevi et al., 2023).

Load balancing yang efektif memastikan bahwa tidak ada node komputasi yang menjadi bottleneck, sambil meminimalkan pengiriman data antar node untuk mengurangi *overhead* jaringan. Teknik *machine learning-based load balancing* dapat beradaptasi dengan pola beban kerja yang dinamis dan memprediksi beban kerja di masa depan untuk membuat keputusan penjadwalan yang lebih baik (Vora, 2025).

9.2.4 Monitoring dan Anomaly Detection

Sistem monitoring real-time sangat penting untuk mendeteksi masalah performa dan mengalokasikan sumber daya dengan lebih efektif. *Machine learning-based anomaly detection*

dapat digunakan untuk mengidentifikasi pola penggunaan sumber daya yang tidak normal, yang mungkin mengindikasikan malfungsi sistem atau serangan keamanan. Sistem monitoring canggih menggunakan *deep learning* models untuk menganalisis log sistem dan metrik kinerja dalam volume besar untuk memberikan wawasan yang actionable (Uddoh et al., 2021).

9.3 Integrasi Edge AI dan Cloud Intelligence

Integrasi yang harmonis antara *edge AI* dan *cloud intelligence* merupakan kunci untuk membangun sistem yang dapat menangani aplikasi AI yang kompleks dalam skala besar sambil mempertahankan latensi rendah dan privasi data.

9.3.1 Hybrid Inference dan Collaborative Processing

Model *hybrid inference* memungkinkan bagian dari proses inferensi AI untuk dilakukan di *edge* (untuk latensi rendah dan privacy) dan bagian lainnya dilakukan di *cloud* (untuk akurasi tinggi dan sumber daya yang lebih besar). Pendekatan ini sangat berguna untuk aplikasi yang memerlukan penyeimbangan antara persyaratan latensi yang ketat dan kebutuhan untuk akurasi prediksi yang tinggi (Chatterjee, 2025).

Sistem kolaboratif memungkinkan *edge nodes* untuk bekerja sama dalam melakukan tugas-tugas AI yang kompleks, dengan setiap *node* berkontribusi sesuai dengan kapabilitas dan ketersediaannya. Pola komunikasi dalam sistem kolaboratif harus

dirancang untuk meminimalkan overhead komunikasi sambil memastikan bahwa hasil akhir akurat (Bellala, 2025).

9.3.2 Federated Learning dan Privacy-Preserving AI

Federated learning adalah paradigma pembelajaran mesin di mana model dilatih pada perangkat terdistribusi tanpa mengirimkan data mentah ke server pusat, sehingga meningkatkan privasi data secara signifikan. Pendekatan ini sangat penting untuk aplikasi-aplikasi yang menangani data sensitif seperti data kesehatan atau informasi finansial (Khanna et al., 2025).

Dalam *federated learning*, setiap *edge device* melatih model lokal pada datanya sendiri, kemudian hanya model yang diperbarui (bukan data) yang dikirim ke server pusat untuk agregasi. Proses agregasi menggabungkan pembaruan model dari berbagai perangkat untuk menghasilkan model global yang lebih baik daripada apa yang bisa dicapai oleh satu perangkat individual (Chen et al., 2021).

9.3.3 Synchronization dan Consistency Management

Menjaga konsistensi data dan model di seluruh sistem komputasi terdistribusi adalah tantangan yang kompleks. Sistem harus mengelola *versioning model*, memastikan bahwa semua node menggunakan model yang kompatibel, dan menangani kegagalan jaringan yang mungkin menyebabkan inkonsistensi (Bershchansky et al., 2024).

Mekanisme *synchronization* yang canggih menggunakan algoritma seperti *gossip protocols* dan *consensus algorithms* untuk memastikan bahwa semua *node* memiliki view yang konsisten tentang state sistem. *Blockchain technology* juga mulai digunakan

untuk memberikan penjaminan konsistensi yang lebih kuat dalam sistem terdistribusi (Sujay et al., 2026).

9.3.4 Data Management dan Storage Optimization

Mengelola data dalam sistem komputasi berbasis AI yang terdistribusi memerlukan strategi yang cermat untuk *caching*, *replication*, dan *data placement* yang optimal. Data yang sering diakses harus disimpan lebih dekat ke edge untuk mengurangi latensi, sementara data yang jarang diakses dapat disimpan di cloud untuk menghemat sumber daya edge yang terbatas (Fahim et al., 2025).

Sistem penyimpanan modern menggunakan teknik kompresi data dan *incremental synchronization* untuk mengurangi bandwidth yang diperlukan untuk mentransfer data antara *nodes*. *Data deduplication* dan *compression* dapat mengurangi kebutuhan penyimpanan hingga 70-80%, menghemat biaya secara signifikan (Halapeti, 2025).

9.4 Optimisasi Energi dan Keberlanjutan

Konsumsi energi dari sistem komputasi berbasis AI telah menjadi masalah yang semakin penting seiring dengan meningkatnya ukuran dan kompleksitas model AI. Optimisasi energi tidak hanya menguntungkan lingkungan tetapi juga mengurangi biaya operasional secara signifikan.

9.4.1 Energy-Aware Resource Management

Sistem manajemen sumber daya modern mengintegrasikan pertimbangan energi ke dalam keputusan alokasi sumber daya. Algoritma *power-aware scheduling* dapat memilih untuk menjalankan tugas-tugas AI pada nodes yang lebih hemat energi, bahkan jika itu berarti latensi sedikit lebih tinggi. Teknik *dynamic voltage and frequency scaling* (DVFS) dapat mengurangi konsumsi daya dengan menurunkan frekuensi CPU ketika beban kerja memungkinkan (Pujol et al., 2025).

Machine learning dapat digunakan untuk memprediksi konsumsi energi dari berbagai konfigurasi sistem dan secara otomatis memilih konfigurasi yang paling hemat energi sambil memenuhi persyaratan *Quality of Service*. Pendekatan ini dapat mengurangi konsumsi energi hingga 40% dalam beberapa skenario (Zeng et al., 2021).

9.4.2 Hardware Accelerators dan Specialized Architectures

Penggunaan *hardware accelerators* seperti GPU, TPU, dan FPGA dapat secara signifikan meningkatkan efisiensi energi dalam menjalankan beban kerja AI dibandingkan dengan menggunakan CPU *general-purpose*. GPU sangat efisien untuk pemrosesan paralel yang besar, sementara TPU dioptimalkan khusus untuk operasi tensor yang umum dalam pembelajaran mesin (Mshragi et al., 2025).

FPGA menawarkan fleksibilitas tinggi dan efisiensi energi yang luar biasa dengan memungkinkan konfigurasi hardware yang disesuaikan dengan beban kerja spesifik. Untuk aplikasi *edge* dengan daya terbatas, FPGA dapat mencapai performa yang jauh

lebih tinggi per watt dibandingkan dengan CPU atau GPU umum (Hanz et al., 2025).

9.4.3 Model Optimization untuk Green Computing

Teknik optimisasi model tidak hanya mengurangi latensi dan penggunaan memori, tetapi juga mengurangi konsumsi energi secara signifikan. *Model pruning* menghilangkan parameter yang tidak perlu, mengurangi komputasi yang diperlukan dan dengan demikian konsumsi energi. *Quantization* mengurangi presisi numerik, yang mengurangi penggunaan memori dan bandwidth, menghemat energi (Echarif et al., 2025).

Knowledge distillation menghasilkan model yang lebih kecil dan lebih cepat yang mengkonsumsi lebih sedikit energi, membuatnya ideal untuk perangkat *edge* dengan daya terbatas. Dalam beberapa kasus, model yang dioptimalkan melalui *knowledge distillation* dapat mencapai performa yang hampir sama dengan model original sambil menggunakan 50-80% lebih sedikit energi (Salaudeen et al., 2024).

9.4.4 Sustainable Infrastructure dan Renewable Energy

Pada tingkat infrastruktur, pusat data modern semakin mengadopsi energi terbarukan untuk mengurangi jejak karbon mereka. Sistem otomatis dapat menjadwalkan beban kerja AI untuk berjalan pada waktu ketika energi terbarukan tersedia dalam jumlah besar, mengurangi kebutuhan untuk energi dari sumber yang kurang berkelanjutan (Cicceri et al., 2023).

Sistem *smart grid* yang terintegrasi dengan AI dapat memprediksi permintaan energi dan secara otomatis

mengeseimbangkan beban di antara berbagai sumber energi untuk memaksimalkan penggunaan energi terbarukan. Teknologi ini dapat meningkatkan efisiensi keseluruhan sistem energi hingga 20-30% (Akhtaruzzaman et al., 2025).

9.5 Keamanan dan Privasi dalam Sistem AI Terdistribusi

Keamanan dan privasi merupakan aspek kritis dalam sistem komputasi berbasis AI yang terdistribusi, mengingat volume data sensitif yang diproses dan potensi untuk serangan cyber yang lebih canggih.

9.5.1 Zero Trust Architecture dan AI-Powered Security

Paradigma keamanan tradisional yang mengandalkan perimeter pertahanan sudah tidak lagi cukup untuk sistem terdistribusi modern. *Zero Trust Architecture* (ZTA) mengadopsi prinsip "never trust, always verify" yang memerlukan verifikasi otentikasi untuk setiap akses, terlepas dari lokasi atau identitas pengguna (N et al., 2025).

Sistem keamanan berbasis AI menggunakan teknologi seperti *anomaly detection* berbasis *deep learning* dan *graph neural networks* untuk mengidentifikasi perilaku yang mencurigakan dengan akurasi yang jauh lebih tinggi daripada metode tradisional. Sistem ini dapat mendeteksi serangan yang sebelumnya tidak dikenal dengan mengidentifikasi pola yang menyimpang dari perilaku normal (N et al., 2025).

9.5.2 Encryption dan Data Protection

Enkripsi data baik saat transit maupun *at rest* merupakan persyaratan fundamental untuk melindungi data sensitif dalam sistem terdistribusi. Teknik seperti *homomorphic encryption* memungkinkan pemrosesan data yang dienkripsi tanpa perlu mendekripsi terlebih dahulu, memungkinkan komputasi pada data sensitif sambil mempertahankan privasi (Mohammed et al., 2025).

Differential privacy adalah teknik yang memungkinkan pengumpulan dan analisis data yang akurat sambil memberikan jaminan privasi formal kepada individu dalam dataset. Teknik ini sangat berguna untuk *federated learning* dan aplikasi lainnya di mana privasi data sangat penting (Khanna et al., 2025).

9.5.3 Threat Detection dan Incident Response

Sistem deteksi ancaman modern menggunakan pembelajaran mesin untuk mengidentifikasi pola yang menunjukkan aktivitas berbahaya atau anomali dalam jaringan dan sistem. *Machine learning*-based threat detection dapat mencapai akurasi 95% atau lebih tinggi dalam mengidentifikasi serangan yang sebelumnya tidak diketahui (Uddoh et al., 2021).

Sistem respons insiden otomatis dapat mengambil tindakan korektif secara real-time untuk mengisolasi sistem yang terkena, meminta bantuan manusia ketika diperlukan, dan melakukan forensik untuk memahami akar penyebab dari serangan. Otomasi respons insiden dapat mengurangi waktu pemulihan dari jam menjadi menit dalam beberapa kasus (Rana et al., 2025).

9.5.4 Blockchain untuk Integrity dan Accountability

Blockchain technology dapat digunakan untuk memberikan jaminan integritas data dan audit trail yang tidak dapat diubah dalam sistem komputasi berbasis AI. Setiap transaksi atau perubahan model dapat dicatat dalam *blockchain*, menciptakan catatan permanen yang dapat diperiksa oleh pihak yang berwenang (Sujay et al., 2026).

Sistem berbasis *blockchain* juga memungkinkan *decentralized trust management*, di mana kepercayaan tidak perlu dipusatkan pada otoritas tunggal. Ini sangat penting untuk sistem yang melibatkan banyak pihak yang saling tidak mempercayai tetapi perlu berkolaborasi (Sujay et al., 2026).

9.6 Latihan Soal

1. Jelaskan konsep *edge-cloud continuum* dan bagaimana integrasi antara edge computing dan cloud computing dapat meningkatkan efisiensi sistem AI berbasis IoT dalam menangani aplikasi real-time yang memiliki kebutuhan latensi rendah.
2. Analisis berbagai teknik kompresi model AI (pruning, quantization, knowledge distillation) dan jelaskan bagaimana teknik-teknik ini dapat meningkatkan efisiensi energi dan mengurangi kebutuhan sumber daya dalam perangkat edge yang resource-constrained.
3. Bagaimana federated learning dapat mengatasi tantangan privasi data dalam sistem AI terdistribusi? Jelaskan mekanisme

pembelajaran model tanpa mentransmisikan data mentah dan analisis trade-off antara akurasi model dan privasi data.

4. Diskusikan peran machine learning dalam optimisasi manajemen sumber daya dinamis dan predictive scaling dalam infrastruktur cloud. Bagaimana algoritma pembelajaran mesin dapat memprediksi beban kerja dan mengalokasikan sumber daya lebih efisien?
5. Jelaskan pendekatan Zero Trust Architecture dengan AI-powered security dalam sistem komputasi terdistribusi. Bagaimana anomaly detection berbasis deep learning dapat meningkatkan deteksi ancaman cyber dan respons insiden otomatis?

Bab 10: Integrasi Metode Kecerdasan Buatan dalam Sistem

Integrasi metode kecerdasan buatan (*Artificial Intelligence*) dalam sistem organisasi telah menjadi strategi kritis untuk mencapai transformasi digital dan keunggulan kompetitif di era modern. Perkembangan pesat dalam teknologi *machine learning*, *natural language processing*, dan *predictive analytics* telah membuka peluang baru untuk mengotomatisasi proses bisnis, meningkatkan kualitas keputusan, dan mengoptimalkan alokasi sumber daya di berbagai sektor industri. Bab ini akan membahas secara mendalam tentang berbagai aspek integrasi kecerdasan buatan dalam sistem enterprise, mulai dari fondasi konseptual, metodologi implementasi, aplikasi praktis, hingga tantangan dan peluang yang dihadapi organisasi dalam perjalanan transformasi digital mereka.

10.1 Fondasi Konseptual dan Strategi Integrasi AI

10.1.1 Paradigma dan Nilai Tambah Integrasi AI

Integrasi kecerdasan buatan dalam sistem organisasi merepresentasikan pergeseran paradigma dari otomasi berbasis

aturan tradisional menuju sistem yang adaptif, pembelajaran, dan responsif terhadap perubahan kondisi bisnis (Reddy, 2025). Paradigma ini mengubah cara organisasi memandang data, dari aset statis yang digunakan untuk pelaporan historis menjadi aset dinamis yang dimanfaatkan untuk prediksi, optimisasi, dan pengambilan keputusan real-time. Nilai tambah utama dari integrasi AI meliputi peningkatan efisiensi operasional melalui otomasi cerdas, peningkatan akurasi keputusan melalui analitik berbasis data, dan penciptaan peluang inovasi bisnis yang sebelumnya tidak mungkin (Kumari, 2024).

Penelitian menunjukkan bahwa organisasi yang berhasil mengintegrasikan AI dalam sistem mereka mencapai peningkatan signifikan dalam berbagai metrik kinerja, termasuk pengurangan biaya operasional hingga 42%, peningkatan efisiensi proses hingga 53%, dan peningkatan akurasi dalam prediksi manajemen sumber daya hingga 91% (Kumari, 2024). Namun, kesuksesan integrasi AI bukan hanya tentang adopsi teknologi, tetapi memerlukan pendekatan holistik yang mencakup transformasi data, perubahan proses bisnis, dan pengembangan kapabilitas organisasi (S. & Zeebaree, 2025).

10.1.2 Kerangka Kerja Integrasi AI dalam Enterprise Systems

Kerangka kerja yang solid merupakan fondasi penting untuk integrasi AI yang sukses dalam sistem enterprise. Kerangka kerja harus mencakup berbagai dimensi, termasuk dimensi teknologi, data, organisasi, dan manusia. Dari perspektif teknologi, kerangka kerja harus mengidentifikasi infrastruktur cloud yang sesuai,

platform integrasi data, dan arsitektur sistem yang mendukung skalabilitas dan keandalan (Baddam, 2025). Dimensi data mencakup strategi pengumpulan data, kualitas data, governess data, dan keamanan data, yang merupakan fondasi penting untuk keberhasilan model AI.

Dimensi organisasi melibatkan penetapan tujuan AI yang jelas, identifikasi use case prioritas, alokasi sumber daya yang tepat, dan pembentukan tim dengan kompetensi yang diperlukan. Dimensi manusia mencakup pengembangan keterampilan karyawan, manajemen perubahan, dan penciptaan budaya organisasi yang mendukung inovasi dan eksperimen dengan teknologi baru (Joshi et al., 2025). Kerangka kerja yang komprehensif ini memastikan bahwa integrasi AI tidak hanya menghadirkan teknologi baru, tetapi juga mengubah cara organisasi beroperasi, membuat keputusan, dan menciptakan nilai bagi stakeholder mereka (Chintam, 2025).

10.1.3 Arsitektur Sistem dan Integrasi Data

Arsitektur sistem untuk integrasi AI harus dirancang dengan mempertimbangkan skalabilitas, keandalan, keamanan, dan kemudahan pemeliharaan. Arsitektur modern mengintegrasikan berbagai komponen, termasuk *data lakes*, sistem *business intelligence*, mesin pembelajaran, dan antarmuka pengguna yang intuitif dalam ekosistem yang kohesif. Platform integrasi data berbasis cloud seperti Informatica dengan *CLAIRE AI engine* memungkinkan organisasi untuk mengelola data dari berbagai sumber, melakukan transformasi data yang cerdas, dan memastikan kualitas data secara otomatis (Pandurangan, 2025).

Integrasi data yang efektif memerlukan strategi yang mencakup identifikasi sumber data, definisi model data yang konsisten, pembangunan data pipelines yang reliable, dan implementasi monitoring untuk memastikan integritas data. Pendekatan cloud-native dengan arsitektur multi-cloud dan hybrid memungkinkan fleksibilitas dalam pemilihan platform dan mengurangi *vendor lock-in* (Jaiswal, 2025). Dengan mengintegrasikan teknologi *Internet of Things* (IoT) dan *edge computing*, organisasi dapat melakukan analisis data di tingkat lokal untuk respons real-time, sementara *cloud* menangani analisis kompleks dan penyimpanan data skala besar (Koneru, 2025).

10.1.4 Penyesuaian Strategi dengan Konteks Industri

Strategi integrasi AI harus disesuaikan dengan karakteristik unik setiap industri, termasuk kebutuhan regulasi, kompleksitas proses bisnis, dan tingkat kematangan digital organisasi. Dalam industri manufaktur, fokus utama sering pada prediksi pemeliharaan, optimisasi kualitas, dan otomasi proses produksi, dimana machine learning digunakan untuk menganalisis data sensor dari mesin dan memprediksi kebutuhan pemeliharaan sebelum terjadi kerusakan (Boiko et al., 2024). Dalam sektor kesehatan, integrasi AI berfokus pada peningkatan diagnosis akurat, optimisasi manajemen pasien, otomasi *revenue cycle management*, dan dukungan keputusan klinis (Dachepalli, 2024).

Dalam sektor keuangan, AI digunakan untuk deteksi fraud, analisis risiko kredit, dan personalisasi layanan pelanggan. Namun, penyesuaian dengan konteks industri memerlukan pemahaman

mendalam tentang tantangan spesifik industri, regulasi yang berlaku, dan harapan stakeholder. Organisasi yang berhasil adalah mereka yang dapat menyeimbangkan ambisi transformasional dengan realistis tentang kecepatan implementasi dan sumber daya yang diperlukan (Kotha et al., 2024).

10.2 Metodologi dan Pendekatan Implementasi

10.2.1 Proses Identifikasi dan Pemilihan Use Case

Langkah pertama dalam implementasi AI yang sukses adalah identifikasi dan pemilihan use case yang tepat. Proses ini memerlukan analisis mendalam tentang proses bisnis saat ini, identifikasi pain points, dan evaluasi potensi dampak dari implementasi AI. Use case yang ideal adalah mereka yang memiliki dampak bisnis yang signifikan, data yang tersedia dalam volume dan kualitas yang memadai, dan kompleksitas yang sesuai untuk pilot awal (Joshi et al., 2025). Pendekatan yang terstruktur melibatkan pembentukan tim lintas fungsi yang mencakup domain experts, data scientists, dan business analysts untuk mengevaluasi setiap kandidat use case.

Prioritisasi use case harus mempertimbangkan tidak hanya potensi ROI, tetapi juga kelayakan teknis, ketersediaan data, dan alignment dengan strategi bisnis jangka panjang organisasi. Banyak organisasi mulai dengan use case yang relatif sederhana namun memberikan nilai yang terukur, seperti prediksi demand dalam supply chain atau deteksi anomali dalam financial transactions,

untuk membangun momentum dan expertise sebelum menangani use case yang lebih kompleks (Subramanyam, 2025). Dengan memilih use case yang tepat dan mendokumentasikan pembelajaran dari setiap implementasi, organisasi dapat secara bertahap membangun kapabilitas AI mereka dan meningkatkan kesuksesan dalam implementasi selanjutnya.

10.2.2 Tahap Pengembangan dan Pelatihan Model

Pengembangan model AI adalah proses iteratif yang melibatkan persiapan data, pemilihan algoritma yang sesuai, pelatihan model, evaluasi, dan penyesuaian parameter untuk meningkatkan kinerja. Tahap persiapan data seringkali memakan waktu lebih lama dari yang diharapkan, mengingat kebutuhan untuk membersihkan data, mengatasi data yang hilang, dan memastikan keseimbangan data untuk training yang efektif. Pemilihan algoritma harus didasarkan pada karakteristik masalah, ukuran dan jenis data, dan trade-off antara akurasi dan interpretabilitas (Juyal et al., 2025).

Proses training model melibatkan penggunaan dataset training yang representative, validasi menggunakan cross-validation atau holdout set, dan tuning hyperparameter untuk mengoptimalkan kinerja. Evaluasi model harus mempertimbangkan berbagai metrik kinerja yang relevan dengan tujuan bisnis, seperti accuracy, precision, recall, F1-score untuk masalah klasifikasi, atau MAE, RMSE untuk masalah regresi. Selain metrik teknis, penting juga untuk mengevaluasi model dari perspektif bisnis, termasuk biaya false positives dan false negatives dalam konteks operasi aktual (Boiko et al., 2024). Model yang telah divalidasi kemudian

dideploy dalam lingkungan produksi dengan monitoring yang ketat untuk memastikan kinerja tetap konsisten dan merespons perubahan kondisi data.

10.2.3 Integrasi dengan Sistem Legacy dan Cloud

Integrasi AI dengan sistem existing merupakan salah satu tantangan terbesar dalam implementasi AI di organisasi besar yang memiliki landscape IT yang kompleks. Sistem legacy seringkali tidak dirancang untuk interoperabilitas modern dan memiliki arsitektur yang sulit untuk dimodifikasi. Pendekatan integrasi yang efektif melibatkan penggunaan *Application Programming Interfaces* (APIs), *Enterprise Service Bus*, dan *middleware* yang dapat menghubungkan sistem lama dengan platform AI dan cloud modern (Chintam, 2025). Dalam banyak kasus, organisasi mengadopsi strategi hybrid yang mempertahankan investasi di sistem legacy sambil secara bertahap bermigrasi ke infrastruktur cloud-native.

Integrasi dengan cloud computing membuka peluang untuk memanfaatkan skalabilitas, elastisitas, dan layanan managed AI yang ditawarkan oleh platform cloud utama seperti AWS, Azure, dan Google Cloud. Platform cloud menyediakan layanan pre-built untuk machine learning, *natural language processing*, computer vision, dan layanan lainnya yang dapat diintegrasikan ke dalam aplikasi enterprise. Namun, integrasi dengan cloud juga memerlukan perhatian khusus terhadap keamanan data, privacy, dan compliance dengan regulasi lokal dan industri (Kumari, 2024). Organisasi yang berhasil adalah mereka yang dapat mengembangkan strategi

integrasi yang fleksibel, scalable, dan aman yang memungkinkan mereka untuk memanfaatkan kekuatan sistem legacy sambil adopsi teknologi AI modern.

10.2.4 Manajemen Perubahan dan Adopsi Pengguna

Implementasi AI yang sukses bukan hanya tentang teknologi, tetapi juga tentang manajemen perubahan yang efektif dan adopsi oleh pengguna. Karyawan seringkali memiliki kekhawatiran tentang automation yang dapat menggantikan pekerjaan mereka, sehingga komunikasi yang jelas tentang bagaimana AI akan mengubah peran mereka dan menciptakan peluang baru sangat penting. Program pelatihan yang komprehensif harus dirancang untuk mempersiapkan karyawan dengan keterampilan yang diperlukan untuk bekerja dengan sistem berbasis AI dan memanfaatkan insights yang dihasilkan oleh model (Tarafdar, 2025).

Strategi adopsi pengguna yang efektif melibatkan keterlibatan early adopters yang dapat menjadi champion untuk teknologi baru, feedback loops yang konsisten untuk penyesuaian sistem berdasarkan kebutuhan pengguna, dan recognition untuk early wins dan success stories. Penciptaan budaya organisasi yang mendukung eksperimen, pembelajaran dari kegagalan, dan continuous improvement adalah fondasi untuk implementasi AI jangka panjang yang sukses. Organisasi harus juga mempertimbangkan implikasi etika dan fairness dari sistem AI, termasuk perlindungan terhadap bias algoritma dan transparansi dalam pengambilan keputusan otomatis (Apu, 2025).

10.3 Aplikasi AI dalam Proses Bisnis Utama

10.3.1 Otomasi Proses Bisnis Berbasis AI

Otomasi proses bisnis (*Business Process Automation* — BPA) berbasis AI melampaui otomasi berbasis aturan tradisional dengan menambahkan kemampuan pembelajaran, adaptasi, dan pengambilan keputusan yang cerdas. Kombinasi *Robotic Process Automation* (RPA) dengan AI menciptakan sistem yang dapat menangani tugas-tugas kompleks yang sebelumnya memerlukan intervensi manusia. Dalam fungsi keuangan, AI dapat digunakan untuk otomasi entry data, validasi dokumen, pemrosesan invoice, dan deteksi anomali dalam transaksi finansial dengan akurasi yang tinggi (Odogwu et al., 2023). Dalam supply chain, AI dapat dioptimalkan demand forecasting, perencanaan inventori, dan rute pengiriman berdasarkan berbagai faktor seperti historis penjualan, tren pasar, dan kondisi cuaca.

Hyperautomation, yang mengintegrasikan RPA, AI, dan proses mining, memungkinkan organisasi untuk mengotomatisasi workflow end-to-end yang kompleks melintasi berbagai sistem dan departemen. Penelitian menunjukkan bahwa implementasi AI-powered BPA dapat mengurangi biaya operasional hingga 40%, meningkatkan efisiensi hingga 50%, dan meningkatkan kepuasan pelanggan melalui respons yang lebih cepat dan akurat (Odogwu et al., 2023). Namun, kesuksesan otomasi berbasis AI memerlukan pemilihan proses yang tepat, data quality yang baik, dan

keseimbangan yang cermat antara otomasi dan involvement manusia untuk memastikan kontrol dan kualitas (Venkatachalapathi, 2024).

10.3.2 Sistem Pendukung Keputusan AI-Enhanced

Sistem pendukung keputusan yang ditingkatkan dengan AI memberikan kemampuan analitik yang jauh lebih powerful daripada sistem tradisional dengan mengintegrasikan predictive analytics, machine learning, dan data mining dalam satu platform yang user-friendly. Dalam manajemen supply chain, decision support systems dapat menganalisis jutaan kombinasi yang mungkin dari sourcing, production, dan distribution untuk merekomendasikan strategi optimal yang memaksimalkan efisiensi sambil meminimalkan risiko (Sharabati et al., 2024). Dalam healthcare, AI-driven decision support dapat membantu clinicians dalam diagnosis, treatment planning, dan patient risk stratification dengan menganalisis data dari multiple sources termasuk electronic health records, imaging studies, dan genetic information.

Implementasi decision support systems AI-enhanced memerlukan perhatian khusus terhadap explainability, karena keputusan yang diambil berdasarkan rekomendasi sistem harus dapat dijelaskan dan dipahami oleh pengguna (Juyal et al., 2025). Pendekatan yang mengintegrasikan human judgment dengan machine intelligence, dimana AI menyediakan insights dan rekomendasi tetapi keputusan final tetap di tangan manusia, telah terbukti paling efektif dalam aplikasi kritis seperti healthcare dan financial services (Tarafdar, 2025). Dengan menggunakan techniques seperti SHAP (SHapley Additive exPlanations) dan

LIME (Local Interpretable Model-Agnostic Explanations), organizational dapat meningkatkan transparency dari AI models dan membangun trust dari end-users.

10.3.3 Personalisasi dan Customer Experience

AI memungkinkan personalisasi customer experience dalam skala yang sebelumnya tidak mungkin dicapai secara manual. Dalam e-commerce dan retail, machine learning algorithms dapat menganalisis perilaku pelanggan, preferensi, dan historical purchases untuk merekomendasikan produk yang paling relevan untuk setiap individual customer. Natural language processing dapat digunakan untuk memahami customer sentiment dari reviews dan social media, memungkinkan organisasi untuk mengidentifikasi issues dan opportunities dengan cepat (Gunda, 2025). Dalam customer service, AI-powered chatbots dan virtual assistants dapat menangani pertanyaan pelanggan dan menyelesaikan issues 24/7 tanpa ketergantungan pada availability manusia.

Personalisasi berbasis AI juga dapat diterapkan dalam marketing, dimana algorithms dapat mengidentifikasi target audience paling profitable, mengoptimalkan timing dan channel dari marketing communications, dan personalize message content untuk meningkatkan engagement dan conversion rates. Dalam financial services, AI dapat digunakan untuk personalize investment recommendations, insurance products, dan pricing berdasarkan profile dan risk tolerance individual customer (Badmus et al., 2024). Namun, personalisasi berbasis AI juga memerlukan attention terhadap privacy concerns dan ethical considerations, memastikan

bahwa data customer digunakan secara responsible dan dengan proper consent.

10.3.4 Maintenance Prediktif dan Quality Control

Dalam manufacturing dan maintenance-intensive industries, predictive maintenance powered by AI dapat mengurangi unplanned downtime, memperpanjang asset lifespan, dan mengoptimalkan maintenance costs. Machine learning algorithms dapat menganalisis sensor data dari equipment untuk mengidentifikasi patterns yang mengindikasikan potential failures sebelum mereka terjadi, memungkinkan maintenance teams untuk melakukan perbaikan secara proaktif daripada reaktif. Implementasi predictive maintenance pada fleet vehicles dapat mengurangi maintenance costs hingga 30% dan meningkatkan vehicle availability, sementara dalam manufacturing dapat mengurangi production downtime secara signifikan (Ravikumar, 2025).

Quality control berbasis AI menggunakan computer vision dan deep learning untuk inspect produk dengan akurasi yang superior terhadap inspeksi manual. Dalam aerospace manufacturing, AI-driven quality control dapat mendeteksi defects yang sangat kecil dalam composite materials dengan akurasi yang konsisten, memastikan keselamatan produk dan compliance dengan regulatory standards (Gunakala, 2025). Beyond manufacturing, AI dapat diaplikasikan untuk quality assurance dalam software development, clinical diagnostics, dan berbagai proses quality-critical lainnya. Kombinasi dari multiple inspection modalities dan ensemble

learning techniques dapat meningkatkan reliability dan robustness dari quality control systems (Apu, 2025).

10.4 Tantangan dan Strategi Mitigasi

10.4.1 Data Quality dan Governance Challenges

Data quality merupakan foundational challenge dalam implementasi AI, karena bahkan algoritma sophisticated hanya dapat menghasilkan insights berkualitas tinggi dari high-quality input data. Common data quality issues termasuk missing values, inconsistent data formats, duplicates, dan outliers yang dapat compromise model accuracy dan reliability. Establishing comprehensive data governance frameworks menjadi essential, dengan clear definitions tentang data ownership, data stewardship responsibilities, dan data quality standards (Baddam, 2025). Organizations harus invest dalam data cleaning dan preprocessing tools, serta dalam training data teams tentang best practices dalam data management.

Governance challenges juga mencakup issues sekitar data privacy, compliance dengan regulations seperti GDPR dan HIPAA, dan ethical use dari data. Organizations harus establish policies yang jelas tentang bagaimana data akan dikumpulkan, disimpan, diakses, dan digunakan dalam AI models, dengan particular attention terhadap sensitive data categories seperti personal health information atau financial data (S. & Zeebaree, 2025). Data retention policies harus balanced antara needs untuk sufficient historical data untuk training models dengan privacy considerations dan regulatory

requirements. Blockchain dan federated learning approaches menawarkan promising solutions untuk addressing data privacy concerns sambil enabling collaborative AI development across organizations (Katta, 2025).

10.4.2 Skills Gap dan Workforce Development

Implementasi AI yang sukses memerlukan multidisciplinary teams dengan expertise dalam machine learning, data engineering, domain knowledge, dan change management. Namun, shortage global dalam AI talent merupakan significant barrier untuk adoption. Organizations harus develop comprehensive talent strategies yang mencakup recruitment dari external talent pools, upskilling existing employees, dan development dari internal training programs (Apu, 2025). Universities dan training institutions harus juga adapt curricula mereka untuk ensure bahwa graduates dilengkapi dengan skills yang relevan untuk AI era.

Beyond technical skills, organizations harus juga develop soft skills seperti communication, collaboration, dan critical thinking yang essential untuk effective human-AI collaboration. Management dan leadership teams harus be trained untuk understand capabilities dan limitations dari AI, sehingga mereka dapat make informed decisions tentang AI adoption dan resource allocation (Joshi et al., 2025). Career paths dalam AI fields harus be designed untuk incentivize talented individuals untuk join organizations dan remain engaged dalam long-term. Many successful organizations have established AI centers of excellence yang serve sebagai internal

knowledge hubs, providing training, mentoring, dan best practice sharing untuk teams across organization.

10.4.3 Model Maintenance dan Drift Management

Deployed AI models tidak adalah static entities tetapi memerlukan continuous monitoring dan maintenance untuk ensure mereka tetap performant dalam production environments. Model *drift*, dimana model *performance degrades over time* karena changes dalam *underlying data distributions* atau *relationships*, merupakan *common challenge* dalam *machine learning systems* (Venkatasubramanian, 2018). *Organizations* harus *establish monitoring systems* yang *track model performance metrics* dan *automatically alert teams* ketika *performance falls below acceptable thresholds*. *Retraining strategies* harus *be implemented* untuk *refresh models* dengan *latest data*, dengan *careful attention* terhadap *timing* dan *triggering conditions* untuk *retraining*.

Governance frameworks untuk model management harus address questions tentang model versioning, audit trails, approval processes untuk model updates, dan rollback procedures dalam case of issues. MLOps (Machine Learning Operations) practices dapat help organizations manage complexity dari deploying dan maintaining multiple models across different systems (Boiko et al., 2024). Automated testing frameworks harus be developed untuk ensure bahwa model updates tidak introduce regressions atau unexpected behaviors. Continuous monitoring dan improvements, combined with strong governance dan clear documentation, dapat

help organizations maintain reliability dan performance dari AI systems jangka panjang.

10.4.4 Ethical Considerations dan Regulatory Compliance

Penggunaan AI dalam decision-making, terutama dalam high-stakes domains seperti healthcare, finance, dan criminal justice, raises significant ethical concerns tentang bias, fairness, transparency, dan accountability. Algorithmic bias dapat arise dari biased training data, biased feature selection, atau biased model architectures, leading to unfair treatment dari certain groups. Organizations harus implement bias testing and mitigation strategies, including diverse representation dalam training data, adversarial testing untuk identify biased behaviors, dan fairness-aware algorithms yang explicitly optimize untuk fairness metrics (Haldar et al., 2025). Explainability menjadi increasingly important, dengan regulatory requirements di banyak jurisdictions yang require organizations untuk be able to explain decisions made oleh AI systems (Narayanan, 2024).

Regulatory landscape untuk AI sedang rapidly evolving, dengan regulations seperti EU AI Act, various healthcare regulations, dan sector-specific requirements emerging di banyak countries. Organizations harus stay informed tentang regulatory requirements di jurisdictions dimana mereka operate dan implement compliance mechanisms ke dalam AI governance frameworks mereka. Transparency dalam AI development dan deployment, comprehensive documentation, dan regular audits dapat help organizations demonstrate compliance dengan regulatory

requirements (Joshi et al., 2025). Beyond regulatory compliance, organizations harus consider broader societal impacts dari AI systems mereka dan strive untuk develop and deploy AI dalam manner yang aligned dengan societal values dan norms.

10.5 Studi Kasus dan Best Practices

10.5.1 Transformasi Digital di Sektor Manufaktur

Sektor manufaktur telah menjadi salah satu early adopters dari AI, dengan implementations yang mencakup predictive maintenance, quality control, production optimization, dan supply chain management. Case studies dari leading manufacturers menunjukkan bahwa implementasi AI-driven decision support systems dapat menghasilkan significant improvements dalam production efficiency, product quality, dan cost reduction. Dalam manufacturing execution systems yang enhanced dengan AI, real-time data dari factory floor dapat dianalisis untuk identify bottlenecks, optimize resource allocation, dan predict potential quality issues sebelum produk selesai diproduksi (Boiko et al., 2024).

Integrasi IoT sensors dengan AI analytics memungkinkan manufacturers untuk develop predictive maintenance strategies yang significantly reduce unplanned downtime dan maintenance costs. AI juga dapat digunakan untuk supply chain optimization, dengan algorithms yang forecast demand, identify optimal suppliers, dan optimize logistics routes. Challenges dalam AI implementation di

manufacturing termasuk legacy equipment yang sulit diintegrasikan, workers yang resistant terhadap automation, dan high implementation costs. Successful implementations biasanya melibatkan phased approach yang dimulai dengan pilot projects, clear communication tentang expected benefits, dan investment dalam worker retraining programs (Ravikumar, 2025).

10.5.2 Implementasi AI dalam Healthcare Systems

Healthcare sector telah melihat dramatic improvements dalam diagnostic accuracy, patient outcomes, dan operational efficiency melalui AI implementations. Clinical decision support systems yang powered by AI dapat analyze patient data dan medical literature untuk recommend evidence-based treatment options, potentially improving clinical outcomes dan reducing medical errors. In medical imaging, AI algorithms dapat detect abnormalities dengan accuracy yang comparable atau bahkan superior terhadap experienced radiologists dalam certain conditions (Khwaji et al., 2024). In healthcare operations, AI dapat optimize patient scheduling, predict patient no-shows, dan allocate resources lebih efficiently.

Implementation challenges dalam healthcare termasuk strict regulatory requirements, data privacy concerns, need untuk clinical validation sebelum deployment, dan resistance dari clinicians yang concerned tentang replacement atau devaluation dari clinical judgment (Koneru, 2025). Successful healthcare AI implementations typically involve close collaboration antara clinical teams, data scientists, dan IT teams, dengan clear protocols untuk

ongoing monitoring dan validation. Business models untuk healthcare AI harus balance automation benefits dengan need untuk maintaining human oversight dalam critical clinical decisions, ensuring bahwa AI serves sebagai tool untuk augmenting clinical capabilities daripada replacing clinical judgment.

10.5.3 Enterprise Resource Planning dengan AI Integration

Modern ERP systems semakin mengintegrasikan AI capabilities untuk provide intelligent automation, predictive analytics, dan advanced decision support. SAP S/4HANA, Oracle Cloud ERP, dan competing platforms sekarang offer built-in AI capabilities untuk berbagai business functions termasuk finance, supply chain, human resources, dan manufacturing. AI-enhanced ERP systems dapat automate routine transactions, provide real-time insights tentang business performance, dan predict future trends untuk enable proactive decision making (Boiko et al., 2024). Dalam financial processes, AI dapat automate accounts payable dan receivable processes, detect fraud, dan provide real-time cash flow forecasting.

Implementasi AI dalam ERP context memberikan leverage dari existing data investments dan business process expertise yang organizations telah develop over years. Cloud-based ERP platforms dengan AI capabilities menyediakan scalability dan access ke latest technologies tanpa need untuk heavy infrastructure investments. Challenges dalam ERP AI implementation termasuk complexity dari ERP systems, customization requirements untuk specific business needs, dan learning curve untuk organizations yang adopting new

platforms. Best practices include phased implementation approach, extensive testing sebelum production deployment, dan ongoing training untuk users tentang bagaimana leverage AI features untuk improve performance mereka.

10.5.4 Intelligent Automation dalam Layanan Pelanggan

Chatbots dan virtual assistants yang powered by AI dan natural language processing telah revolutionize customer service dengan providing immediate responses terhadap customer inquiries, reducing wait times, dan improving customer satisfaction. Advanced implementations menggunakan conversational AI yang dapat understand context, handle complex inquiries, dan seamlessly escalate ke human agents ketika necessary (Joy, 2025). These systems dapat be integrated dengan CRM systems untuk provide personalized responses based on customer history dan preferences, leading ke enhanced customer experience.

Beyond simple FAQ answering, intelligent customer service systems dapat perform complex tasks seperti troubleshooting problems, processing transactions, dan providing recommendations. Integration dengan backend systems seperti order management, inventory management, dan billing systems enable end-to-end automation dari customer interactions. Challenges dalam implementing intelligent customer service termasuk ensuring natural dan helpful conversational experience, handling edge cases dan unusual inquiries, dan maintaining data security dan privacy dalam context dari sensitive customer information (Gunda, 2025). Successful implementations typically involve continuous refinement

berdasarkan customer feedback, regular updates untuk improve accuracy dan coverage, dan clear handoff processes untuk escalation ke human agents (Devi et al., 2025).

10.6 Latihan Soal

1. Jelaskan bagaimana paradigma integrasi AI dalam sistem enterprise berbeda dari otomasi berbasis aturan tradisional. Apa keuntungan utama dari pendekatan AI-driven dan bagaimana hal ini mengubah cara organisasi membuat keputusan?
2. Analisis tahapan implementasi AI dalam organisasi, mulai dari identifikasi use case hingga deployment dan maintenance model. Apa saja tantangan utama di setiap tahap dan bagaimana organisasi dapat mengatasinya?
3. Diskusikan bagaimana AI dapat diintegrasikan dengan sistem legacy sambil melakukan migrasi gradual ke infrastruktur cloud-native. Apa strategi terbaik untuk memastikan business continuity selama proses integrasi?
4. Jelaskan peran data governance dan data quality dalam kesuksesan implementasi AI. Bagaimana organisasi dapat establish comprehensive frameworks untuk memastikan bahwa data digunakan secara responsible dan dalam compliance dengan regulatory requirements?
5. Analisis implikasi etika dan sosial dari implementasi AI dalam bisnis, termasuk concerns tentang bias algoritma, job displacement, dan transparency. Bagaimana organisasi dapat

develop dan deploy AI secara responsible sambil memaksimalkan business benefits?

PROFILE PENULIS



Goldie Gunadi, S.Kom., M.Kom. lahir di Jakarta pada 6 Juli 1978 dan berdomisili di Jakarta Selatan. Ia memiliki minat dalam bidang membaca, olahraga, dan bermain game yang mendukung pengembangan wawasan serta keseimbangan aktivitas sehari-hari.

Melalui pemikiran dan pengalamannya, Goldie Gunadi menekankan pentingnya budaya membaca sebagai kunci untuk memperluas pengetahuan dan membuka wawasan. Ia percaya bahwa semakin banyak seseorang membaca, semakin luas pula pemahaman dan pengalaman yang diperoleh dalam kehidupan.



Ahmad Pudoli, S.Kom., M.Kom. lahir di Tangerang pada 4 Maret 1991 dan berdomisili di Jakarta Selatan, DKI Jakarta. Ia memiliki latar belakang di bidang teknologi informasi dan pemrograman, khususnya dalam pengembangan ilmu komputer dan bahasa pemrograman.

Melalui karya dan pengajarannya, Ahmad Pudoli berfokus pada pengenalan dasar-dasar pemrograman, khususnya bahasa C sebagai fondasi penting dalam memahami logika dan struktur program modern. Ia berharap pembaca dapat tidak hanya memahami konsep teknis, tetapi juga menumbuhkan rasa ingin tahu dan semangat untuk terus mengeksplorasi dunia pemrograman.



Dewi Kusumaningsih, S.Kom., M.Kom. lahir di Jakarta pada 10 Desember 1984 dan berdomisili di Jakarta Selatan. Ia memiliki minat dalam kegiatan memasak, mengajar, dan traveling yang turut memperkaya pengalaman serta wawasan hidupnya.

Sebagai akademisi di bidang teknologi informasi, Dewi Kusumaningsih berkomitmen dalam dunia pendidikan dan pengembangan ilmu pengetahuan. Melalui karya dan pengajarannya, ia berharap dapat memberikan manfaat serta menambah wawasan bagi para pembaca.



Sirwan, S.Pd., M.Pd. lahir di Sarroanging pada 10 Maret 1992. Ia menempuh pendidikan S1 Pendidikan Teknik Elektronika konsentrasi Komputer di Universitas Negeri Makassar dan melanjutkan S2 Pendidikan Teknologi dan Kejuruan di Universitas Negeri Yogyakarta.

Dalam perjalanan kariernya, Sirwan telah berpengalaman sebagai dosen di berbagai perguruan tinggi, di antaranya Universitas Nani Bili Nusantara Sorong, Universitas Teknologi Akba Makassar, serta Universitas Negeri Makassar. Saat ini, ia juga aktif sebagai tutor online di Universitas Terbuka. Melalui pengabdian dan karya akademiknya, ia berkomitmen untuk mengembangkan ilmu pengetahuan di bidang teknologi dan pendidikan, serta meyakini bahwa sebaik-baiknya ilmu adalah ilmu yang bermanfaat bagi orang lain.



Shinta Widyaningtyas, S.TP., M.T. lahir di Malang pada 14 Juni 1993 dan berdomisili di Malang, Jawa Timur. Ia memiliki minat dalam bidang menyanyi, memasak, dan membuat kue yang menjadi bagian dari keseharian serta pengembangan kreativitasnya.

Sebagai akademisi, Shinta Widyaningtyas berkomitmen dalam pengembangan ilmu pengetahuan di bidang teknologi. Melalui karya dan pemikirannya, ia berharap dapat memberikan manfaat bagi orang lain serta berkontribusi dalam penyebaran ilmu yang bermanfaat bagi masyarakat luas.



Julfikar Mawansyah, S.Kom., MT. lahir di Bima pada 3 Januari 1993. Ia menempuh pendidikan S1 Teknik Informatika, S2 Teknik Sipil, dan saat ini sedang menempuh pendidikan S3 Teknik Elektro dan Informatika (ongoing). Dengan latar belakang multidisiplin, ia memiliki kompetensi luas di bidang teknologi dan rekayasa sistem.

Dalam perjalanan kariernya, Julfikar Mawansyah berpengalaman sebagai Full Stack Developer, Data Scientist, dan Project Manager. Saat ini, ia berprofesi sebagai dosen serta aktif dalam pengembangan ilmu pengetahuan di bidang teknologi informasi. Melalui karya dan pengabdianya, ia berharap dapat menjadi sumber inspirasi, berbagi pengetahuan, serta memberikan kontribusi yang bermanfaat bagi dunia akademik maupun praktik kehidupan nyata.



Machrus Ali, ST., MT., IPM. lahir di Lamongan pada 6 Juni 1971. Ia menyelesaikan pendidikan S1 di Universitas Darul Ulum Jombang dan S2 di Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Dengan latar belakang pendidikan tersebut, ia memiliki kompetensi yang kuat di bidang teknik elektro dan pengembangan ilmu keteknikan.

Dalam perjalanan kariernya, Machrus Ali telah menjabat berbagai posisi akademik, di antaranya Ketua Jurusan Teknik Elektro (2015–2019), Kepala LPPM Universitas Darul Ulum (2020–2024), serta Koordinator Pengelola Jurnal Universitas Darul Ulum (2024–2028). Saat ini, ia berprofesi sebagai dosen dan aktif dalam pengembangan penelitian serta publikasi ilmiah. Melalui pengabdian dan karyanya, ia berpegang pada nilai bahwa ilmu pengetahuan akan meninggikan derajat orang-orang yang beriman dan berilmu.



Eka Wulansari Fridayanthie, M.Kom. lahir di Garut pada 20 Agustus 1982. Ia menempuh pendidikan S1 di Universitas Komputer Indonesia dan S2 di Universitas Nusa Mandiri, dengan fokus pada bidang teknologi informasi.

Dalam perjalanan kariernya, Eka Wulansari Fridayanthie pernah menjadi dosen di Universitas Komputer Indonesia pada tahun 2004–2005 dan sejak tahun 2006 hingga sekarang aktif sebagai dosen di Universitas Bina Sarana Informatika. Melalui pengabdian di dunia pendidikan, ia terus berkomitmen dalam pengembangan ilmu pengetahuan dan teknologi serta menginspirasi pembaca untuk tidak mudah menyerah dan selalu bersemangat dalam belajar dan berkarya.

DAFTAR PUSTAKA

- Abase, A. H., Hassan, H. A., & Omara, F. (2025). From text to knowledge: A hybrid framework for enhanced semantic analysis using language models and heuristics. *IEEE Proceedings*, 11313208, 1–12. <https://doi.org/10.1109/ICICIS66182.2025.11313208>
- Abimannan, S., El-Alfy, E. M., Chang, Y. S., Hussain, S., Shukla, S., & Satheesh, D. (2023). Ensemble multifeatured deep learning models and applications: A survey. *IEEE Access*, 11, 122859–122880.
- Adenekan, O. A., Solomon, N. O., Simpa, P., & Obasi, C. (2024). Enhancing manufacturing productivity: A review of AI-driven supply chain management optimization and ERP systems integration. *International Journal of Multidisciplinary Engineering Research and Development*, 6(5), 1126-1145. <https://doi.org/10.51594/ijmer.v6i5.1126>
- Agrawal, R. R., Shinde, S. R., Rohatgi, D., & Khaitan, S. (2025). Comparative analysis of artificial intelligence and machine learning: Concepts, methodologies, applications, and societal impact. *International Journal of Applied Mathematics*, 38(10S), 1071.
- Ahmadi, M. N., O'Neil, M., Fragala-Pinkham, M., Lennon, N., & Trost, S. (2018). Machine learning algorithms for activity

- recognition in ambulant children and adolescents with cerebral palsy. *Journal of NeuroEngineering and Rehabilitation*, 15(1), 456. <https://doi.org/10.1186/s12984-018-0456-x>
- Akbar, S., Khan, S., & Amin, M. Y. (2025). Capital structure and speed of adjustment of shariah-compliant firms: Does the shariah screening criteria influence the results? *Journal of Islamic Accounting and Business Research*, 16(7). <https://doi.org/10.1108/jiabr-07-2024-0240>
- Al Bataineh, A., Kaur, D., & Jalali, S. M. J. (2022). Multi-layer perceptron training optimization using nature inspired computing. *IEEE Access*, 10, 41086–41102. <https://doi.org/10.1109/access.2022.3164669>
- Ali, R., Hussain, A., Nazir, S., Khan, S., & Khan, H. U. (2023). Intelligent decision support systems—An analysis of machine learning and multicriteria decision-making methods. *Applied Sciences*, 13(22), 12426. <https://doi.org/10.3390/app132212426>
- Alkhudhayr, A. (2024). Deep learning in the era of big data: Foundations, advances, applications, challenges, and future directions. *International Journal of Advanced Research*, 18581.
- Alom, M. Z., Carminati, B., & Ferrari, E. (2018). Detecting spam accounts on Twitter. In *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks*

- Analysis and Mining (pp. 1027-1031). <https://doi.org/10.1109/ASONAM.2018.8508495>
- Alviano, M., Grillo, L., Lo Scudo, F., & Reiners, L. A. R. (2025). Integrating answer set programming and large language models for enhanced structured representation of complex knowledge in natural language. *IJCAI Proceedings*, 482, 1–10. <https://doi.org/10.24963/ijcai.2025/482>
- Andrade-Arenas, L., Celis, D. H., & Arias, C. Y. (2025). The crucial role of artificial intelligence in addressing climate change. *International Journal of Advances in Intelligence Informatics*, 14(1), 1-11. <https://doi.org/10.11591/ijai.v14.i1.pp1-11>
- Apu, K. U. (2025). AI-driven data analytics and automation: A systematic literature review of industry applications. *Sustainable Development and Monitoring Initiative*, 2(1), 9-45. <https://doi.org/10.71292/sdmi.v2i01.9>
- Aravindan, A. (2024). A comprehensive survey on multi-agent cooperative decision-making: Scenarios, approaches, challenges and perspectives. *ArXiv Preprint ArXiv:2503.13415*.
- Aslan, B., Salam, R. B., & Li, S. (2018). Automatic detection of cyber security related accounts on online social networks: Twitter as an example. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (pp. 1642-1644). <https://doi.org/10.1145/3217804.3217919>

- Azizov, D., Manzoor, M. A., Bojkovic, V., Wang, Y., Wang, Z., Iklassov, Z.,... & Liang, S. (2024). A decade of deep learning: A survey on the magnificent seven. *ArXiv Preprint ArXiv:2412.16188*.
- Badmus, O., Rajput, S., Arogundade, J. B., & Williams, M. H. (2024). AI-driven business analytics and decision making. *World Journal of Advanced Research and Reviews*, 24(1), 3093-3120. <https://doi.org/10.30574/wjarr.2024.24.1.3093>
- Baena, E., Testolina, P., Polese, M., Aliaga, S., Benincasa, A., Koutsonikolas, D.,... & Melodia, T. (2025). Agentic semantic control for autonomous wireless space networks: Extending Space-O-RAN with MCP-driven distributed intelligence. *ArXiv Preprint ArXiv:2506.10925*.
- Beck, M., Scarlata, C., Fortson, L., Lintott, C., Simmons, B., Galloway, M. A.,... & Marshall, P. J. (2018). Integrating human and machine intelligence in galaxy morphology classification tasks. *Monthly Notices of the Royal Astronomical Society*, 476(4), 5516-5534. <https://doi.org/10.1093/mnras/sty503>
- Berenzweig, A., Logan, B., Ellis, D. P. W., & Whitman, B. (2003). A large-scale evaluation of acoustic and subjective music similarity measures. *Proceedings of the 4th International Society for Music Information Retrieval Conference*, 8, 1-8. <https://doi.org/10.7916/d87m0jkn>

- Berhane, T. M., Lane, C., Wu, Q., Autrey, B., Anenkhonov, O., Chepinoga, V., & Liu, H. (2018). Decision-tree, rule-based, and random forest classification of high-resolution multispectral imagery for wetland mapping and inventory. *Remote Sensing*, 10(4), 580. <https://doi.org/10.3390/rs10040580>
- Bhatya, T. J., Salomi, J., Jothi, B., Pal, S., Taqi, I., Islam, D., Raghuram, K. S., & Soni, M. (2024). IoT and AI synergy in society: Robotics and automation perspectives. *Proceedings of the 2024 IEEE International Conference on Computing, Communication and Automation*, 1-8. <https://doi.org/10.1109/IC3I61595.2024.10828680>
- Bhende, M., Joshi, S., Sainath, C. G., Surendarkumar, S., Pasha, M. Y., & Parasa, G. (2025). Multi-agent AI systems for decentralized decision-making. *2025 Global IT Conference (GITCON)*.
- Bodner, A. D., Tepsich, A. S., Spolski, J. N., & Pourteau, S. (2024). Convolutional Kolmogorov-Arnold networks. *arXiv Preprint arXiv:2406.13155*.
- Boiko, R., Aulin, V., Tykhyi, A., Karpushyn, S., & Koval, O. (2024). Integration of artificial intelligence into decision support systems in optimizing production processes of a machine-building enterprise. *Science and Innovation*, 10(41), 95-103. [https://doi.org/10.32515/2664-262x.2024.10\(41\).2.95-103](https://doi.org/10.32515/2664-262x.2024.10(41).2.95-103)
- Bukhsh, M., Ali, M. S., Ashraf, M., Alsubhi, K., & Chen, W. (2022). An interpretation of long short-term memory recurrent neural

- network for approximating roots of polynomials. *IEEE Access*, 10, 36839–36857.
- Capra, M., Bussolino, B., Marchisio, A., Masera, G., Martina, M., & Shafique, M. (2020). Hardware and software optimizations for accelerating deep neural networks: Survey of current trends, challenges, and the road ahead. *IEEE Access*, 8, 225383–225406. <https://doi.org/10.1109/ACCESS.2020.3039858>
- Castillo-Zuiga, I., Rosas, F. J. L., Rodríguez-Martínez, L., Arteaga, J. M., López-Veyna, J., & Rodríguez-Díaz, M. A. (2020). Internet data analysis methodology for cyberterrorism vocabulary detection, combining techniques of big data analytics, NLP and semantic web. *International Journal of Semantic Web and Information Systems*, 16(1), 104–128. <https://doi.org/10.4018/ijswis.2020010104>
- Chang, S. F. (2002). The holy grail of content-based media analysis. *IEEE Multimedia*, 9(4), 6–10. <https://doi.org/10.1109/93.998041>
- Chen, J., Yang, H., Liu, Z., & Joe-Wong, C. (2026). The five Ws of multi-agent communication: Who talks to whom, when, what, and why--A survey from MARL to emergent language and LLMs. *ArXiv Preprint*.
- Chen, L., Li, S., Bai, Q., Yang, J., Jiang, S., & Miao, Y. (2021). Review of image classification algorithms based on convolutional neural networks. *Remote Sensing*, 13(22), 4712.

- Chen, R., Wan, F., & Yu, H. (2022). A survey of knowledge representation learning based on structure and semantics. *ACM Transactions on Knowledge Discovery from Data*, 3546607, 1–18. <https://doi.org/10.1145/3546607.3546621>
- Chen, S., & Harrington, M. (2025). Multi-agent reinforcement learning for dynamic resource allocation in real-time bidding platforms. *Marine Resource Economics Information Service*, 154.
- Chen, Y., Ge, X., Yang, S., Hu, L., Li, J., & Zhang, J. (2023). A survey on multimodal knowledge graphs: Construction, completion and applications. *Mathematics*, 11(8), 1815. <https://doi.org/10.3390/math11081815>
- Chiroma, F., Liu, H., & Haig, E. (2018). Suicide-related text classification with Prism algorithm. In *Proceedings of the 2018 10th International Conference on Machine Learning and Computing* (pp. 358-363). <https://doi.org/10.1109/ICMLC.2018.8527032>
- Chollet, F. (2016). Xception: Deep learning with depthwise separable convolutions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1610–1619.
- Chowdhury, M. E. H., Rahman, T., Khandakar, A., Mazhar, R., Kadir, M. A., Mahbub, Z. B.,... & Islam, M. T. (2020). Can AI help in screening viral and COVID-19 pneumonia? *IEEE Access*, 8, 132665-132676. <https://doi.org/10.1109/access.2020.3010287>

- Chrimawaty, B. E., Emilia, O., Rahayu, G., & Ana, I. (2022). Clinical reasoning pattern used in oral health problem solving—A case study in Indonesian undergraduate dental students. *BMC Medical Education*, 22(1), 808. <https://doi.org/10.1186/s12909-022-03808-7>
- Chugh, A., Sharma, V. K., Kumar, S., Nayyar, A., Qureshi, B., Bhatia, M. K., & Jain, C. (2021). Spider monkey crow optimization algorithm with deep learning for sentiment classification and information retrieval. *IEEE Access*, 9, 20341-20357. <https://doi.org/10.1109/access.2021.3055507>
- Collins, K., Umamaheswarareddy, C. (2025). AI-enabled process automation in enterprise application integration: Bridging legacy systems and cloud-native platforms. *Journal of Computer Science and Technology Studies*, 7(8), 97-125. <https://doi.org/10.32996/jcsts.2025.7.8.97>
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297. <https://doi.org/10.1007/BF00994018>
- Dachepalli, V. (2024). A smarter ERP: How artificial intelligence is reshaping enterprise workflows. *Journal of Artificial Intelligence and Management*, 2(2), 265-285. <https://doi.org/10.55124/jaim.v2i2.265>
- Dameruppula, S. (2025). The evolution and impact of artificial intelligence and machine learning: A technical overview. *International Journal of Advance Research, Science and Technology*, 24609.

- Dannels, S. R., Kumar, P., Kumar, R., Sinha, U., & Mehta, S. (2025). AI at the edge: Cloud-edge synergy. *International Journal for Innovative Science and Research Technology*, 25(5), 967. <https://doi.org/10.38124/ijisrt/25may967>
- Daoud, H. G., & Bayoumi, M. (2019). Efficient epileptic seizure prediction based on deep learning. *IEEE Transactions on Biomedical Circuits and Systems*, 13(5), 798-808. <https://doi.org/10.1109/TBCAS.2019.2929053>
- Das, M., Mandal, S., Roy, P., Roy, A., Mukherjee, A., Mahala, S., & Das, A. (2025). Design and implementation of multi-agent systems: A Lang Graph-based approach. 2025 International Conference on Emerging Innovative Technology (ICoEIT).
- De Alban, J. D. T., Connette, G. M., Oswald, P., & Webb, E. L. (2018). Combined Landsat and L-Band SAR data improves land cover classification and change detection in dynamic tropical landscapes. *Remote Sensing*, 10(2), 306. <https://doi.org/10.3390/rs10020306>
- Deb, K., Lu, Z., Kropp, I., Hernandez-Suarez, J. S., Hussein, R., Miller, S., & Nejadhashemi, A. (2023). Minimizing expected deviation in upper level outcomes due to lower level decision making in hierarchical multiobjective problems. *IEEE Transactions on Evolutionary Computation*, 27(3), 645-659. <https://doi.org/10.1109/TEVC.2022.3172302>
- Deist, T. M., Dankers, F., Valdes, G., Wijsman, R., Hsu, I., Oberije, C.,... & Lambin, P. (2018). Machine learning algorithms for outcome prediction in (chemo)radiotherapy: An empirical

- comparison of classifiers. *Medical Physics*, 45(7), 3894-3907. <https://doi.org/10.1002/mp.12967>
- Dematheis, F., Walter, M., Lang, D., Antwerpen, M. H., Scholz, H., Pfalzgraf, M.-T.,... Wölfel, R. (2022). Machine learning algorithms for classification of MALDI-TOF MS spectra from phylogenetically closely related species *Brucella melitensis*, *Brucella abortus* and *Brucella suis*. *Microorganisms*, 10(8), 1658. <https://doi.org/10.3390/microorganisms10081658>
- Devi, C., Balasubramanian, K., & Rajeswari, N. (2025). AI-powered supply chain management in manufacturing: A data-driven approach to materials logistics. *Proceedings of the 2025 IEEE International Conference on Robotics and Manufacturing*, 1-10. <https://doi.org/10.1109/ROMA66616.2025.11155325>
- Devi, K. V. R., S, S. B., Lakhanpal, S., Kalra, R., Sethi, V. A., & Thajil, S. K. (2024). A review: Swarm robotics: Cooperative control in multi-agent systems. *E3S Web of Conferences*, 505, 03013.
- Dey, M. K., Dey, S., & Das, D. (2024). Indian stock price prediction using optimal gated recurrent unit (GRU) network. *2024 IEEE International Conference on Advanced Electronics and Communications (AREIS)*, 2559–2570.
- Dhivya, M. H., Shiba Raj, K., & Surendarkumar, S. (2025). Embodied and multi-agent reinforcement learning: Advances, challenges and opportunities. *International*

Journal of Innovative Science and Research Technology,
10(3).

- Ding, Y., Jiang, W., Liu, S., Jing, Y., Guo, J., Wang, Y., Zhang, J., Wang, Z., Liu, Z., Du, B., Liu, X., & Tao, D. (2025). Dynamic parallel tree search for efficient LLM reasoning. arXiv preprint arXiv:2502.16235, 1-15. <https://doi.org/10.48550/arXiv.2502.16235>
- Djoumbou-Feunang, Y., Eisner, R., Knox, C., Chepelev, L., Hastings, J., Owen, G.,... & Wishart, D. S. (2016). ClassyFire: Automated chemical classification with a comprehensive, computable taxonomy. *Journal of Cheminformatics*, 8(3), 174.
- Duanmu, H., Li, X., Yuan, Z., Zheng, S., Duan, J., Zhang, X., & Lin, D. (2025). MxMoE: Mixed-precision quantization for MoE with accuracy and performance co-design. arXiv Preprint, 505799. <https://doi.org/10.48550/arXiv.2505.05799>
- Dudek, G., Peka, P., & Smyl, S. (2021). A hybrid residual dilated LSTM and exponential smoothing model for midterm electric load forecasting. *IEEE Transactions on Neural Networks and Learning Systems*, 31(5), 1494–1505.
- Duggal, R., Prakash, G., & Das, D. (2025). Frequency-based deep learning model for beam damage detection. *Structural Control and Health Monitoring*, 2025, 2495134.
- E., F. M. D., Julil, M. S., Mehedy, M. T. J., Saeed, M., Snigdha, E. Z., Khan, M. N., & Khan, N. (2025). Optimizing revenue cycle management in healthcare: AI and IT solutions for

- business process automation. *Turkish Journal of Advanced Engineering and Technology*, 7(3), 14-35. <https://doi.org/10.37547/tajet/volume07issue03-14>
- Egbuhuzor, N. S., Ajayi, A., Akhigbe, E. E., & Agbede, O. O. (2022). AI in enterprise resource planning: Strategies for seamless SaaS implementation in high-stakes industries. *International Journal of Social Sciences and Entrepreneurship Review*, 1(1), 81-95. <https://doi.org/10.54660/ijsser.2022.1.1.81-95>
- Ehtesham, A., Singh, A., Gupta, G. K., & Kumar, S. (2025). A survey of agent interoperability protocols: Model context protocol (MCP), agent communication protocol (ACP), agent-to-agent protocol (A2A), and agent network protocol (ANP). *ArXiv Preprint ArXiv:2505.02279*.
- Eker, E., Kayri, M., Ekinci, S., & Izci, D. (2023). Comparison of swarm-based metaheuristic and gradient descent-based algorithms in artificial neural network training. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 12(1), 1–20. <https://doi.org/10.14201/adcaij.29969>
- El Hajoui, A. (2025). A student performance prediction using RNNs models with variety of optimization techniques in deep learning. *International Journal of Information and Education Technology*, 15(6), 2331–2345.
- Elavarasri, K. (2025). Salesforce classic as well as lightning automation using tosa automation and tosa AI-powered

- Salesforce engine. *Journal of Information Technology*, 14(2), 22055-22075. <https://doi.org/10.26634/jit.14.2.22055>
- Elshenawy, A., Mohammed, A., & Hamouda, S. (2025). The evolution of deep learning: Models, applications, and future directions. 2025 International Conference on Microelectronics and Communications (MIUCC), 11196834.
- Elshewey, A. M., Shams, M. Y., El-Rashidy, N., Elhady, A. M., Shohieb, S. M., & Tarek, Z. (2023). Bayesian optimization with support vector machine model for Parkinson disease classification. *Sensors*, 23(4), 2085. <https://doi.org/10.3390/s23042085>
- Enguehard, J., O'Halloran, P., & Gholipour, A. (2019). Semi-supervised learning with deep embedded clustering for image classification and segmentation. *IEEE Access*, 7, 86888-86900. <https://doi.org/10.1109/ACCESS.2019.2891970>
- Espina-Romero, L. C., Gutiérrez Hurtado, H., Ros Parra, D. E., Vílchez Praela, R. A., Talavera-Aguirre, R., & Ochoa-Díaz, A. (2024). Challenges and opportunities in the implementation of AI in manufacturing: A bibliometric analysis. *Science*, 6(4), 60-82. <https://doi.org/10.3390/sci6040060>
- Ezhilarasi, K., & Kalavathy, G. (2018). Literature survey: Analysis on semantic web information retrieval methodologies. *Proceedings of the 2018 International*

- Conference on Education and Environmental Studies, 18, 119-127. <https://doi.org/10.2991/PECTEAM-18.2018.18>
- Fachrizal, F., Al-Khowarizmi, A., & Nugroho, O. (2026). Load balancing optimization through multi-label classification of network traffic using deep learning in distributed computing systems. *Engineering, Technology & Applied Science Research*, 14, 567.
- Ferrag, M., Tihanyi, N., & Debbah, M. (2025). From LLM reasoning to autonomous AI agents: A comprehensive review. *ArXiv Preprint ArXiv:2504.19678*.
- Gadde, A. (2025). AI-enhanced knowledge management systems in enterprises: Transforming organizational intelligence. *World Journal of Advanced Research and Reviews*, 26(2), 1913. <https://doi.org/10.30574/wjarr.2025.26.2.1913>
- Gaikwad, S. R., Bontha, M. R., Devi, S., & Dumbre, D. (2024). Improving clinical preparedness: Community health nurses and early hypoglycemia prediction in type 2 diabetes using hybrid machine learning techniques. *Public Health Nursing*, 41(5), 13440. <https://doi.org/10.1111/phn.13440>
- Galbusera, F. (2025). Machine learning in spine research and clinical practice. *EFORT Open Reviews*, 10(6), 023. <https://doi.org/10.1302/1358-992x.2025.6.023>
- Gandolfo, M. E., Brarda, M. E. B., Tamargo, L. H., & García, A. J. (2024). Argument-based cooperative decision-making for autonomous vehicles. *Proceedings of the 2024 ACM*

SIGMOD/PODS International Conference on Management of Data, 3696463.

- Gao, Y., Zhang, Z., Zhang, X., Xu, L., Yu, Y., & Tang, M. (2019). Deep convolutional neural networks for WCE abnormality detection: CNN architecture, region proposal and transfer learning. *IEEE Access*, 7, 30314–30325.
- Ghalibaf, A. K., Nazari, E., GholianAval, M., & Tara, M. (2019). Comprehensive overview of computer-based health information tailoring: A systematic scoping review. *BMJ Open*, 9(1), 021022. <https://doi.org/10.1136/bmjopen-2017-021022>
- Ghane, S., Sawant, R., Supe, G., & Pichad, C. (2024). LangchainIQ: Intelligent content and query processing. *International Journal of Modern Trends in Science and Technology*, 3(3), 360-375. <https://doi.org/10.47992/ijmts.2581.6012.0360>
- Ghimire, D., Kil, D., & Kim, S. (2022). A survey on efficient convolutional neural networks and hardware acceleration. *Electronics*, 11(6), 945.
- Ghosh, S., Dasgupta, A., & Swetapadma, A. (2019). A study on support vector machine based linear and non-linear pattern classification. *Proceedings of the 2nd International Conference on Intelligent Systems, Modelling and Simulation*, 1–7. <https://doi.org/10.1109/iss1.2019.8908018>
- Goldstein, E., Keidar, D., Yaron, D., Shachar, Y., Blass, A., Charbinsky, L.,... & Eldar, Y. C. (2020). COVID-19 classification of X-ray images using deep neural

- networks. *European Radiology*, 31(2), 1-9. <https://doi.org/10.1007/s00330-021-08050-1>
- Gui, J., Chen, T., Zhang, J., Cao, Q., Sun, Z., Luo, H., & Tao, D. (2024). A survey on self-supervised learning: Algorithms, applications, and future trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48, 1–21.
- Gullipalli, T., Murali, K., & Peri, S. (2025). An improved taylor hyperbolic tangent and sigmoid activations for avoiding vanishing gradients in recurrent neural nets. *International Arab Journal of Information Technology*, 22(5), 15.
- Gunakala, K. K. (2025). AI-driven automation for aerospace manufacturing: Enhancing quality control through integrated systems. *Journal of Computer Science and Technology Studies*, 7(3), 73-95. <https://doi.org/10.32996/jcsts.2025.7.3.73>
- Gunasekara, N., Pfahringer, B., Gomes, H. M., & Bifet, A. (2025). Gradient boosted bagging for evolving data stream regression. *Data Mining and Knowledge Discovery*, 39, 65. <https://doi.org/10.1007/s10618-025-01147-x>
- Gupta, A., & Agarwal, P. (2024). Enhancing sales forecasting accuracy through integrated enterprise resource planning and customer relationship management using artificial intelligence. *Proceedings of the 2024 IEEE International Conference on AI and Internet of Things*, 1-8. <https://doi.org/10.1109/AIIoT58432.2024.10574785>

- Gupta, G. P., & Kulariya, M. (2016). A framework for fast and efficient cyber security network intrusion detection using Apache Spark. *Procedia Computer Science*, 93, 824-831. <https://doi.org/10.1016/j.procs.2016.07.238>
- Gupta, R. D., Biswas, D., Paul, A., Kapali, S., & Khan, M. S. (2024). Empowering digital transformation: The strategic role of artificial intelligence in enterprise innovation. *European Journal of Theoretical and Applied Sciences*, 2(6), 16-45. [https://doi.org/10.59324/ejtas.2024.2\(6\).16](https://doi.org/10.59324/ejtas.2024.2(6).16)
- Gupta, S., Jain, S., Katiyar, N., Verma, S., Tiwari, M., Awasthi, P., & Singh, P. (2025). Human-robot interaction interface design using computer vision and natural language processing. *Semantic Journal*, 9–103.
- Gupta, S., Kishan, B., & Gulia, P. (2024). Comparative analysis of predictive algorithms for performance measurement. *IEEE Access*, 12, 60640–60665. <https://doi.org/10.1109/ACCESS.2024.3372082>
- Haldar, U., Rahman, M. M., Miah, M. A., Uddin, M., & Yousuf, S. B. (2025). Measuring the ROI of AI implementation in enterprise systems: A managerial perspective on MIS and analytics integration. *Proceedings of the 2025 IEEE International Conference on Electronics and Information Technology*, 1-10. <https://doi.org/10.1109/ICECIE66637.2025.11363846>
- Hamilton, K., Nayak, A., Boi, B., & Longo, L. (2022). Is neuro-symbolic AI meeting its promises in natural language

- processing? A structured review. *Semantic Web Journal*, 223228, 1–22. <https://doi.org/10.3233/sw-223228>
- Hammad, M. M. (2024). Artificial neural network and deep learning: Fundamentals and theory. *ArXiv*, 2408.16002.
- Hanz, T., Ali, M., Sadman, Z., & Qasem, A. (2025). Autotuning CNN workloads on the edge: A hybrid approach with cross-domain embeddings. *IEEE Proceedings*, 65507, 62. <https://doi.org/10.1109/COMPSAC65507.2025.00062>
- Harrison, J. (2009). Handbook of practical logic and automated reasoning. *Journal of Logic and Computation*, 19(6), 1105–1107. <https://doi.org/10.1093/logcom/exp045>
- Hasan, M. K., Alam, M. A., Elahi, M. E., Roy, S., & Wahid, S. R. (2020). CVR-Net: A deep convolutional neural network for coronavirus recognition from chest radiography images. *arXiv preprint arXiv:2007.12073*.
- Hassan, C. A. U., Khan, M. S., & Shah, M. A. (2018). Comparison of machine learning algorithms in data classification. In *Proceedings of the 2018 International Conference on Advances in Computing and Communication Engineering* (pp. 153-157). <https://doi.org/10.23919/IConAC.2018.8748995>
- Hassan, Y. A., & Zeebaree, S. R. M. (2025). Big data cloud computing and AI-driven digital marketing in enterprise systems. *Engineering and Technology Journal*, 10(4), 28-62. <https://doi.org/10.47191/etj/v10i04.28>

- Hatomono, A. P. (2024). Digital zakat in the era of Islamic social finance: A systematic review of economic impact and institutional transformation. *Sharia Contemporary Review*, 1(3), 59.
- Hidri, A., Alsaif, S., Alahmari, M., Alshehri, E., & Hidri, M. S. (2025). Opinion mining and analysis using hybrid deep neural networks. *Technologies*, 13(5), 175. <https://doi.org/10.3390/technologies13050175>
- Horcas, J., Galindo, J., Heradio, R., Fernández-Amors, D., & Benavides, D. (2021). Monte Carlo tree search for feature model analyses: A general framework for decision-making. *Proceedings of the 25th ACM International Systems and Software Product Line Conference*, 1, 1-12. <https://doi.org/10.1145/3461001.3471146>
- Hossain, M. A., Arafat, M. S., Desai, K., Akter, S., & Asha, A. I. (2025). The economic impact of AI-driven remote patient monitoring: A business intelligence perspective on healthcare cost optimization. *Business Intelligence and Society Review*, 6(5), 3-35. <https://doi.org/10.55640/business/volume06issue05-03>
- Hu, S., Fang, Z., Fang, Z., Deng, Y., Chen, X., Fang, Y., & Kwong, S. (2024). AgentsCoMerge: Large language model empowered collaborative decision making for ramp merging. *IEEE Transactions on Mobile Computing*, 3564163.

- Huang, T. (2025). Deep learning techniques and applications. *Proceedings of International Conference on Education and Software Engineering*, 2025, 25828.
- Indhuja, R., Sivakami, G., Sowndarya Devi, S., & Sowndarya, A. K. (2025). Solar power prediction using LSTM & RNN. *International Journal of Scientific Research and Technology*, 25, 1047.
- Iqbal, S., Qureshi, A. N., Li, J., & Mahmood, T. (2023). On the analyses of medical images using traditional machine learning techniques and convolutional neural networks. *Archives of Computational Methods in Engineering*, 30, 2365–2387.
- Islam, S., Elmekki, H., Elsebai, A., Bentahar, J., Drawel, N., Rjoub, G., & Pedrycz, W. (2023). A comprehensive survey on applications of transformers for deep learning tasks. *arXiv Preprint arXiv:2306.07303*.
- Jabbar, A., Iqbal, S., Tamimy, M. I., Rehman, A., Bahaj, S. A., & Saba, T. (2023). An analytical analysis of text stemming methodologies in information retrieval and natural language processing systems. *IEEE Access*, 11, 122340-122370. <https://doi.org/10.1109/access.2023.3332710>
- Jadhav, N., Sugandhi, R., Pawar, R., Shirke, S., & Nalavade, J. E. (2025). Variance adaptive optimization for the deep learning applications. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 15(7), 1023677.

- Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering. *ACM Computing Surveys*, 31(3), 264-323. <https://doi.org/10.1145/331499.331504>
- Jaiswal, I. A. (2025). Integrating AI into enterprise Java applications for secure high performance and scalable systems. *International Journal of Computer Engineering and Science*, 4(086), 1-25. <https://doi.org/10.22399/ijcesen.4086>
- Jawad, Z. N., & Balzs, V. (2024). Machine learning-driven optimization of enterprise resource planning (ERP) systems: A comprehensive review. *BMC Research Notes*, 17(460), 1-20. <https://doi.org/10.1186/s43088-023-00460-y>
- Jaya, H., Sabran, S., Idris, M. M., Djawad, Y. A., Ilham, A., & Ahmar, A. S. (2018). *Kecerdasan Buatan (Cetakan Pertama)*. Fakultas MIPA Universitas Negeri Makassar.
- Jian, J., Chen, Y., Li, Q., Li, H., Zheng, X., & Han, C. (2024). Decision-making method of multi-UAV cooperate air combat under uncertain environment. *IEEE Journal of Miniaturization for Air and Space Systems*, 5(2), 3378726-3378745. <https://doi.org/10.1109/JMASS.2024.3378726>
- Jiang, P., Xu, X., Tao, H., Zhao, L., & Zou, C. (2022). Convolutional-recurrent neural networks with multiple attention mechanisms for speech emotion recognition. *IEEE Transactions on Cognitive and Developmental Systems*, 14(4), 1408–1419. <https://doi.org/10.1109/TCDS.2021.3123979>

- Jin, Y., Li, P., & Zhang, W. (2018). Hybrid macro/micro level backpropagation for training deep spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 29(6), 2670–2682.
- Jinguji, A., Sato, S., & Nakahara, H. (2018). An FPGA realization of a random forest with k-means clustering using a high-level synthesis design. *IEICE Transactions on Information and Systems*, 101(2), 521-531. <https://doi.org/10.1587/transinf.2017rcp0006>
- Jordan, M. I. (2004). Graphical models. *Statistical Science*, 19(1), 140-155. <https://doi.org/10.1214/0883423040000000026>
- Joshi, G., Walambe, R., & Kotecha, K. (2021). A review on explainability in multimodal deep neural nets. *IEEE Access*, 9, 59800–59821.
- Joshi, R. (2025). Comparative analysis of LSTM and GRU neural network architectures for plasma discharge prediction using tokamak diagnostic time series data. 2025 IEEE International Conference on Communications and Information Networks (CICN), 11368018.
- Joshi, R., Singh, V., & Sehgal, U. (2025). A systematic review of artificial intelligence enabled data-driven decision making in management and IT. *Journal of New Sciences*, 13(1), 9260-9295. <https://doi.org/10.63682/jns.v13i1.9260>
- Joy, M. (2025). Agentic workflows in healthcare: Advancing clinical efficiency through AI integration. *Journal of Computer*

- Science and Engineering Technology Studies, 7(8), 112396-112420. <https://doi.org/10.32628/cseit25112396>
- Juyal, P., Manukonda, P., Saratchandran, D., Trehan, A., Shah, K. N., Rao, C., & Katru. (2025). The role of artificial intelligence in enhancing decision-making in enterprise information systems. *Journal of Information Systems and Enterprise Management*, 10(3), 371-405. <https://doi.org/10.52783/jisem.v10i3s.371>
- Kamau, E., & Myllynen, T. (2025). A predictive auto-scaling framework for microservices in distributed systems: A cost-performance optimization approach for U.S. enterprises. *International Journal of Advanced Information and Emerging Technologies*, 6(2), 113–133. <https://doi.org/10.54660/ijalet.2025.6.2.113-133>
- Karim, M. R., & Saha, R. (2025). A framework-based meta-analysis of artificial intelligence-driven ERP solutions for circular and sustainable supply chains. *Journal of Business and Working Environment Systems*, 2(49), 1-45. <https://doi.org/10.63125/jbws2e49>
- Katta, T. B. (2025). Federated learning for enterprise data integration: Examining the application of federated learning to integrate AI models without centralizing enterprise data. *Journal of Information Systems and Enterprise Management*, 10(63), 13989-14025. <https://doi.org/10.52783/jisem.v10i63s.13989>

- Ke, Z., Jiao, F., Ming, Y., Nguyen, X. P., Xu, A., Long, D. X., Li, M., Qin, C., Wang, P., Savarese, S., Xiong, C., & Joty, S. (2025). A survey of frontiers in LLM reasoning: Inference scaling, learning to reason, and agentic systems. *arXiv preprint arXiv:2504.09037*, 1-80. <https://doi.org/10.48550/arXiv.2504.09037>
- Kensert, A., Harrison, P., & Spjuth, O. (2018). Transfer learning with deep convolutional neural networks for classifying cellular morphological changes. *bioRxiv*, 345728.
- Khan, D., Patankar, A., & Kakar, J. (2026). Optimizing hourly air quality index forecasting: A particle swarm optimization-enhanced hybrid approach combining convolutional and recurrent neural networks. *International Journal of Electrical and Computer Engineering*, 16(1), 333–341.
- Khan, R., & Nayak, A. (2025). AI-optimized communication protocols for intelligent control systems. *2025 International Conference on Artificial Intelligence Applications and Innovations (ICAAIC)*, 30.
- Khodayar, M., Liu, G., & Wang, J. (2020). Deep learning in power systems research: A review. *CSEE Journal of Power and Energy Systems*, 7(2), 209-220. <https://doi.org/10.17775/cseejpes.2020.02700>
- Khwaji, A. H., Daghriri, M. A., & Mohammed Awaji, F. A. (2024). The impact of laboratory automation and AI on healthcare delivery: A systematic review. *Health Science and Nursing*, 10(2), 6184-6215. <https://doi.org/10.53555/hsn.v10i2.6184>

- Kila, H. K. (2025). The transformative impact of AI and machine learning in enterprise software testing: A focus on SAP and Salesforce. *World Journal of Advanced Research and Reviews*, 26(2), 1736-1765. <https://doi.org/10.30574/wjarr.2025.26.2.1736>
- Kirichenko, L., Radivilova, T., & Bulakh, V. (2018). Machine learning in classification time series with fractal properties. *Data*, 4(1), 5. <https://doi.org/10.3390/data4010005>
- Kleinberg, J. (1999). Authoritative sources in a hyperlinked environment. *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1, 668-677. <https://doi.org/10.1145/324133.324140>
- Knipper, R. A., Mishty, K., Sadi, M., Shubhra, K., & Santu, K. M. (2024). SNNLP: Energy-efficient natural language processing using spiking neural networks. *arXiv Preprint arXiv:2401.17911*.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90.
- Kumar, A. (2025). AI and edge computing: Real-time collaboration in distributed systems. *World Journal of Advanced Engineering and Technology Studies*, 15(1), 214. <https://doi.org/10.30574/wjaets.2025.15.1.0214>
- Kumar, K. D., Dash, S., & Ganiya, A. R. K. (2025). Ensemble deep learning technique for robust seizure detection using integrating convolutional and recurrent neural networks with

- advanced optimization techniques. *International Journal of Engineering Research and Technology*, 14.
- Kumar, R., & Pamucar, D. (2025). A comprehensive and systematic review of multi-criteria decision-making (MCDM) methods to solve decision-making problems: Two decades from 2004 to 2024. *Sustainability and Development Mapping*, 21(2), 24-56. <https://doi.org/10.31181/sdmap21202524>
- Latha, P., & Kumaresan, P. (2025). Deep learning model optimization for crop prediction and recommendation using Harris hawks optimization. *Environmental Research Letters*, 20, 2515–7620.
- Lei, J., Hu, L., Bu, Y., & Liu, J. (2025). Understanding teams and productivity in information retrieval research: Academia, industry, and cross-community collaborations. *Journal of Data and Information Science*, 2025(1), 51-78. <https://doi.org/10.2478/jdis-2025-0051>
- Lei, Y., Lian, J., Zhu, B., Wang, D., & Wang, W. (2025). A communication approach for cooperation of multi-agent reinforcement learning based on value decomposition. 2025 Chinese Control Conference (CCC).
- Li, H. (2021). Research on big data analysis data acquisition and data analysis. *Proceedings of 2021 International Conference on Artificial Intelligence and Big Data Computing*, 41, 198-203. <https://doi.org/10.1109/CAIBDA53561.2021.00041>
- Li, H., & Deng, L. (2015). Deep sentence embedding using long short-term memory networks: Analysis and application to

- information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4), 694–707. <https://doi.org/10.1109/TASLP.2016.2520371>
- Li, M. (2024). Comprehensive review of backpropagation neural networks. *Journal of Advances in Information Technology*, 15(1), 1–15. <https://doi.org/10.54097/51y16r47>
- Li, M., Zhu, Y., Shen, Y., & Angelova, M. (2022). Clustering-enhanced stock price prediction using deep learning. *World Wide Web*, 25, 1183–1210.
- Li, S., He, X., Xu, X., Zhao, T., Song, C., & Li, J. (2023). Weapon-target assignment strategy in joint combat decision-making based on multi-head deep reinforcement learning. *IEEE Access*, 11, 3324193–3324210. <https://doi.org/10.1109/ACCESS.2023.3324193>
- Li, Y. (2018). Deep reinforcement learning: An overview. In *Proceedings of the 2018 International Symposium on Intelligent Systems and Informatics* (pp. 371–376). https://doi.org/10.1007/978-3-319-94463-0_9
- Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 32(12), 6999–7019.
- Lim, S., Kim, S., Yu, J., Lee, S., Chung, J., & Yu, Y. (2025). VisEscape: A benchmark for evaluating exploration-driven decision-making in virtual escape rooms. *arXiv preprint*

- Liu, S., Chen, Y., Xu, K., & Lin, J. (2022). Emotional analysis of evaluation discourse in business English translation based on language big data mining of public health environment. *Frontiers in Public Health*, 10, 981182. <https://doi.org/10.3389/fpubh.2022.981182>
- Liu, S., Liang, Z., Lyu, X., & Amato, C. (2025). LLM collaboration with multi-agent reinforcement learning. *ArXiv Preprint ArXiv:2508.04652*.
- Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., & Zhang, C. (2017). Learning efficient convolutional networks through network slimming. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2736–2745.
- Lu, Z., Whalen, I., Dhebar, Y. D., Deb, K., Goodman, E., Banzhaf, W., & Boddeti, V. N. (2019). Multiobjective evolutionary design of deep convolutional neural networks for image classification. *IEEE Transactions on Evolutionary Computation*, 25(1), 2–13.
- Luan, K., Wu, F., & Xu, Y. (2025). The analysis of deep learning-based football training under intelligent optimization technology. *Computer Science and Information Systems*, 24, 42.
- Lundervold, A., & Lundervold, A. (2018). An overview of deep learning in medical imaging focusing on MRI. *Zeitschrift für*

- Medizinische Physik, 28(4), 320–340. <https://doi.org/10.1016/j.zemedi.2018.11.002>
- Luo, X. (2025). Constructing text-centric knowledge graphs based on deep learning: A focus on named entity recognition and relation extraction. *SPIE Digital Library*, 3075576, 1–12. <https://doi.org/10.1117/12.3075576>
- Mahmud, M. I., Mamun, M., & Abdelgawad, A. (2023). A deep analysis of brain tumor detection from MR images using deep learning networks. *Algorithms*, 16(4), 176.
- Manda, C. (2024). The PTSA framework: An enterprise architecture for autonomous AI agents. *International Journal of Computer Science and Engineering Technology*, 4(06).
- Mangalika, U. (2024). Object recognition to content based image retrieval: A study of the developments and applications of computer vision. *Journal of Computing and Natural Science*, 4(5), 005.
- Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to information retrieval. *Foundations and Trends in Information Retrieval*, 1(1), 1–135. <https://doi.org/10.1561/15000000001>
- Manso, D. F., Parnell, G. S., Pohl, E., & Belderrain, M. C. (2024). Gaps in strategic problem-solving methods: A systematic literature review. *Journal of Multi-Criteria Decision Analysis*, 31(1), 1828–1855. <https://doi.org/10.1002/mcda.1828>

- Martinez, O., & Sharma, D. (2025). Semantic web technologies for knowledge graph construction and querying. *International Journal of Advanced Computing and Technology Education*, 13(1), 94. <https://doi.org/10.65521/ijacte.v13i1.94>
- Mashragi, M., Petri, I., & Rana, O. F. (2025). FPGA-accelerated fast machine learning for heterogeneous edge systems. *IEEE EDGE Proceedings*, 67623, 29. <https://doi.org/10.1109/EDGE67623.2025.00029>
- Maxwell, A. E., Warner, T., & Fang, F. (2018). Implementation of machine-learning classification in remote sensing: An applied review. *International Journal of Remote Sensing*, 39(9), 2784-2817. <https://doi.org/10.1080/01431161.2018.1433343>
- Meier, F. (2020). Social network analysis as a tool for data analysis and visualization in information behaviour and interactive information retrieval research. *Proceedings of the 2020 ACM SIGIR Forum*, 54(1), 18-25. <https://doi.org/10.1145/3343413.3378018>
- Mienye, I. D., & Swart, T. G. (2024). A comprehensive review of deep learning: Architectures, recent advances, and applications. *Information*, 15(12), 755. <https://doi.org/10.3390/info15120755>
- Mienye, I. D., & Swart, T. G. (2024). Recurrent neural networks: A comprehensive review of architectures, variants, and applications. *Information*, 15(9), 517. <https://doi.org/10.3390/info15090517>

- Mienye, I. E., Jere, N., Obaido, G., Mienye, I. D., & Aruleba, K. (2024). Deep learning in finance: A survey of applications and techniques. *AI*, 5(4), 101. <https://doi.org/10.3390/ai5040101>
- Mohamed, A., Lee, H., Borgholt, L., Havtorn, J. D., Edin, J., Igel, C.,... & Shinji, W. (2022). Self-supervised speech representation learning: A review. *IEEE Journal of Selected Topics in Signal Processing*, 16(6), 1068–1085.
- Mohd Noor, M. H., & Ige, A. (2024). A survey on state-of-the-art deep learning applications and challenges. *arXiv Preprint arXiv:2402.12959*.
- Mohideen, A., Reddy, I. S., Ajay, M., Reddy, K., Kumar, M. P., & Dinesh, T. C. (2025). A hybrid approach of LSTM and ARIMA for enhancing numerical weather prediction accuracy using deep learning. 2025 IEEE International Conference on Intelligent Computing and Robotics Analytics (ICIRCA), 11089776.
- Monti, F., Boscaini, D., Masci, J., Rodol, E., Svoboda, J., & Bronstein, M. M. (2017). Geometric deep learning on graphs and manifolds using mixture model CNNs. 2017 IEEE International Conference on Computer Vision (CVPR), 5115–5124.
- Nagendran, M., Chen, Y., Lovejoy, C. A., Gordon, A., Komorowski, M., Harvey, H.,... & Maruthappu, M. (2020). Artificial intelligence versus clinicians: Systematic review of design,

- reporting standards, and claims of deep learning studies. *BMJ*, 368, m689. <https://doi.org/10.1136/bmj.m689>
- Nangi, P. R., & Settipi, S. (2023). A cloud-native serverless architecture for event-driven, low-latency, and AI-enabled distributed systems. *International Journal of Research and Reviews in Engineering*, 4(4), 113. <https://doi.org/10.63282/3050-922x.ijeret-v4i4p113>
- Nascimento, N., Alencar, P. S. C., & Cowan, D. D. (2023). GPT-in-the-loop: Adaptive decision-making for multiagent systems. *arXiv preprint arXiv:2308.10435*, 1-20. <https://doi.org/10.48550/arXiv.2308.10435>
- Navigli, R., Blloshmi, R., & Martinez Lorenzo, A. C. (2022). BabelNet meaning representation: A fully semantic formalism to overcome language barriers. *AAAI Proceedings*, 36(11), 21490. <https://doi.org/10.1609/aaai.v36i11.21490>
- Nayak, B., Prasanna, V., Ambhika, C., M, A., Maheswari, B., R, S., & Nalini, M. (2023). A comprehensive survey of deep learning: Advancements, applications, and challenges. *International Journal on Recent and Innovation Trends in Computing and Communication*, 11(8), 190–208.
- Neeraj, Mathew, J., Behera, R., & Panthakkalakath, Z. E. (2020). A deep learning framework for COVID outbreak prediction. *arXiv preprint arXiv:2007.15531*.
- Neumuth, T., & Mintz, Y. (2021). Performance analysis of machine learning algorithms for big data classification: ML and AI-

- based algorithms for big data analysis. *International Journal of E-Healthcare Information Systems*, 12(3), 1–22. <https://doi.org/10.4018/IJEHMC.20210701.OA4>
- Nikunj Gupta, L., Twardecka, J. Z. H., Milzman, J., Kannan, R., & Prasanna, V. K. (2025). TIGER-MARL: Enhancing multi-agent reinforcement learning with temporal information through graph-based embeddings and representations. *ArXiv Preprint ArXiv:2511.08832*.
- Noor, M. H. M., & Ige, A. (2024). A survey on state-of-the-art deep learning applications and challenges. *arXiv Preprint. arXiv:2402.12959*.
- Özmuş, H. C., Çavdur, F., & Özmütlü, S. (2006). Automatic new topic identification in search engine transaction logs. *Journal of Information & Knowledge Management*, 5(2), 127-137. <https://doi.org/10.1108/10662240610673727>
- P, N., & Gomathi, N. (2025). Optimized amalgam CNNLSTMGRU framework for robust stroke recognition in IoT-based clinical environments. 2025 *IEEE International Conference on Information and Communication Networks (ICICNIS)*, 11315522.
- Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., Song, X., & Ward, R. (2015). Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4), 694-707. <https://doi.org/10.1109/TASLP.2016.2520371>

- Panapakidis, I. P., & Christoforidis, G. C. (2018). Optimal selection of clustering algorithm via multi-criteria decision analysis (MCDA) for load profiling applications. *Applied Sciences*, 8(2), 237-255. <https://doi.org/10.3390/app8020237>
- Panda, P., & Bhatia, V. (2018). Role of artificial intelligence (AI) in public health. *Indian Journal of Community & Family Medicine*, 4(2), 1442.
- Pandey, S., Pakray, P., & Manna, R. (2024). Quantum classifier for natural language processing applications. *Cybernetics and Systems*, 28(2), 5016.
- Park, S., Liu, X., Gong, Y., & Choi, E. (2024). Ensembling large language models with process reward-guided tree search for better complex reasoning. *arXiv preprint arXiv:2412.15797*. <https://doi.org/10.48550/arXiv.2412.15797>
- Parsanasab, E., Ahmadipour, A., & Mehraeen, E. (2025). Utilization of ontology to develop artificial intelligence systems in the healthcare industry. *Healthcare Informatics Research*, 31(4), 320. <https://doi.org/10.4258/hir.2025.31.4.320>
- Patel, K., Patil, A., Shourya, A., Malviya, R. K., & Solanki, M. (2022). Deep learning for computer vision: A brief overview of YOLO. *International Journal of Advanced Research in Science, Communication and Technology*, 3943.
- Patil, M., & Lokhande, V. (2025). Model context protocol (MCP): Enabling scalable AI data integration. *International Journal of Multidisciplinary Research and Analysis*, 7(02).

- Pereira, R. F., & Meneguzzi, F. (2024). Empowering BDI agents with generalised decision-making. *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, 3635637.
- Pileggi, S. F. (2024). Ontology in hybrid intelligence: A concise literature review. *Future Internet*, 16(8), 268. <https://doi.org/10.3390/fi16080268>
- Prakash, V., & Shukla, A. K. (2025). An AI-enabled privacy-preserving federated learning framework of hybrid BiLSTM-RNN-CNN for deep intelligent intrusion detection in fog computing in the IoT domain. *Concurrency and Computation: Practice and Experience*, 70291.
- Prathi, S., Fathima, F., Rumaiza, S. M., & Elavarasi, M. (2025). Ontologies and artificial intelligence: Bridging semantic gaps in machine understanding. *IEEE Proceedings*, 11280578, 1–8. <https://doi.org/10.1109/ICIDCA66325.2025.11280578>
- Pushpavalli, M., Dhanya, D., Kulkarni, M., Jasmine, R. R., Umarani, B., Reddy, M. R.,... & Yadav, A. S. (2024). Enhancing electrical power demand prediction using LSTM-based deep learning models for local energy communities. *International Journal of Electrical Power & Energy Systems*, 160, 2316246.
- Qiu, Z., Yao, T., & Mei, T. (2017). Learning spatio-temporal representation with pseudo-3D residual networks. 2017

- IEEE International Conference on Computer Vision (ICCV), 4694–4702.
- Rahman, B. H. A., Subri, N. I., & Hanafi, A. G. (2025). Determinants of fintech adoption among family takaful agents: A TAM-based secondary data analysis. *International Journal of Research in Institutional and Social Studies*, 91(1), 126.
- Rane, J., Kaya, Ö., Mallick, S. K., & Rane, N. L. (2024). Artificial intelligence, machine learning, and deep learning in cloud, edge, and quantum computing: A review of trends, challenges, and future directions. *Data Science and Engineering*, 1–20. https://doi.org/10.70593/978-81-981271-0-5_1
- Rao, M. V. S., Bhargavi, I., Abhishek, A., Gopi, P., & Kiran, C. (2022). Deep fake face detection using long short-term memory with deep learning approach. *Journal of Image Processing and Image Recognition Study*, 21, 28–36.
- Rashed, B. M., & Popescu, N. (2021). Machine learning techniques for medical image processing. *Proceedings of the 2021 IEEE 8th International Conference on E-Health and Bioengineering*, 65–72. <https://doi.org/10.1109/EHB52898.2021.9657673>
- Rauf, H. T., Lali, M., Zahoor, S., Shah, S. J. H., Rehman, A., & Bukhari, S. A. C. (2019). Visual features based automated identification of fish species using deep convolutional neural networks. *Computers and Agriculture*, 165, 105075.

- Reddy Deva, M. R., & Jain, N. (2025). Utilizing Azure automated machine learning and XGBoost for predicting cloud resource utilization in enterprise environments. *IEEE Proceedings*, 11102235. <https://doi.org/10.1109/NETCRYPT65877.2025.11102235>
- Rehman, S. U., Tu, S., Rehman, O., Huang, Y., Magurawalage, C. S., & Chang, C. (2018). Optimization of CNN through novel training strategy for visual classification problems. *Entropy*, 20(4), 290. <https://doi.org/10.3390/e20040290>
- Ren, Z., Zhou, S., Liu, D., & Liu, Q. (2025). Physics-informed neural networks: A review of methodological evolution, theoretical foundations, and interdisciplinary frontiers toward next-generation scientific computing. *Applied Sciences*, 15(14), 8092. <https://doi.org/10.3390/app15148092>
- Reyhanmulky, M., Supriani, I., & Al-Ghifary, M. S. (2024). What drives millennials' and Gen Z's intentions to participate in cash waqf linked sukuk? Perspectives from Islamic altruism. *International Journal of Islamic Economics and Finance*, 7(2), 20703.
- Sadigova, N. (2025). The role of artificial intelligence in modern-term formation. *PalArch's Journal of Archaeology of Egypt/Egyptology*, 115, 26. <https://doi.org/10.22178/pos.115-26>
- Salamon, J., & Bello, J. P. (2016). Deep convolutional neural networks and data augmentation for environmental sound

- classification. *IEEE Signal Processing Letters*, 24(11), 1701–1705.
- Samek, W., Montavon, G., Lapuschkin, S., Anders, C. J., & Müller, K. R. (2021). Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, 109(3), 247–278. <https://doi.org/10.1109/jproc.2021.3060483>
- Sarraf, S., & Tofighi, G. (2016). Classification of Alzheimer's disease using fMRI data and deep learning convolutional neural networks. *arXiv Preprint arXiv:1603.08631*.
- Serratore, G., & Clunis, J. (2025). The integration of artificial intelligence and ontologies: Transformations in knowledge representation and application. *Advances in Knowledge Organization*, 7(1), 95643. <https://doi.org/10.7152/nasko.v7i1.95643>
- Shen, L., Sun, Y., Yu, Z., Ding, L., Tian, X., & Tao, D. (2024). On efficient training of large-scale deep learning models. *ACM Computing Surveys*, 56(11), 1–47.
- Shin, H. C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I.,... & Summers, R. M. (2016). Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Transactions on Medical Imaging*, 35(5), 1285–1298.
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 60. <https://doi.org/10.1186/s40537-019-0197-0>

- Shorten, C., Khoshgoftaar, T. M., & Furht, B. (2021). Deep learning applications for COVID-19. *Journal of Big Data*, 8, 18.
- Shrivastava, A., & Dubey, R. (2018). Classification of spam mail using different machine learning algorithms. In *Proceedings of the 2018 International Conference on Advanced Computation and Telecommunication* (pp. 1-8). <https://doi.org/10.1109/ICACAT.2018.8933787>
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv Preprint arXiv:1409.1556*.
- Sitokonstantinou, V., Papoutsis, I., Kontoes, C., Lafarga Arnal, A., Armesto Andrs, A. P., & Garraza Zurbano, J. A. (2018). Scalable parcel-based crop identification scheme using Sentinel-2 data time-series for the monitoring of the common agricultural policy. *Remote Sensing*, 10(6), 911. <https://doi.org/10.3390/rs10060911>
- Srinivasu, P., SivaSai, J. G., Ijaz, M. F., Bhoi, A. K., Kim, W., & Kang, J. J. (2021). Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM. *Sensors*, 21(8), 2852.
- Suganuma, M., Kobayashi, M., Shirakawa, S., & Nagao, T. (2020). Evolution of deep convolutional neural networks using Cartesian genetic programming. *Evolutionary Computation*, 28(1), 141–163.
- Sugil, A. J. (2024). A comprehensive survey of deep learning and its applications in advancing artificial intelligence. *International*

- Journal of Advance Research in Computer Science, 15(2), 7052.
- Sumanth, V., Rao, P., & Kumar, S. R. (2024). A survey on computer vision architectures for large scale image classification using deep learning. *International Journal of Advanced Computer Science and Applications*, 12(10), 13.
- Sure, T. A. R., Saigurudatta, P. V., Kapoor, S., Kandula, S. T. R., Choudhury, A., & Devendran, P. D. (2025). The role of natural language processing in developing intelligent knowledge repositories. *IEEE Proceedings*, 11101416, 1–12. <https://doi.org/10.1109/IAICT65714.2025.11101416>
- Suthanthran, S. S., Pavithra, S., Dineshkumar, P., Sellamuthu, K., Vasundradevi, P., & Geetha, K. (2025). Improving efficiency in machine learning: A deep dive into optimization, data handling, and hardware utilization. *IEEE Proceedings*, 11040327. <https://doi.org/10.1109/AIMLA63829.2025.11040327>
- Tabares-Soto, R., Orozco-Arias, S., Romero-Cano, V. A., Bucheli, V. S., Rodríguez-Sotelo, J., & Jiménez-Varón, C. F. (2020). A comparative study of machine learning and deep learning algorithms to classify cancer types based on microarray gene expression data. *PeerJ Computer Science*, 6, e270. <https://doi.org/10.7717/peerj-cs.270>
- Tarafdar, R. (2025). Embodied and multi-agent reinforcement learning: Advances, challenges and

- opportunities. *International Journal of Innovative Science and Research Technology*, 10.
- Tathipelly, R. (2025). Artificial intelligence and deep learning: Trends and applications. *International Journal of Scientific Research and Engineering Management*, 43336.
- Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T., & Maida, A. (2018). Deep learning in spiking neural networks. *Neural Networks*, 111, 47–63. <https://doi.org/10.1016/j.neunet.2018.12.002>
- Tejeda, Y. G., & Mayer, H. A. (2024). Deep learning with convolutional neural networks: A compact holistic tutorial with focus on supervised regression. *Machine Learning and Knowledge Extraction*, 6(4), 132. <https://doi.org/10.3390/make6040132>
- Thotapalli, R. (2024). Deep learning for computer vision: A brief overview of YOLO. *International Journal of Advanced Research in Science, Communication and Technology*, 3943.
- Todera, M. (2025). Artificial intelligence for sustainability: A systematic review and critical analysis of AI applications, challenges, and future directions. *Sustainability*, 17(17), 8049. <https://doi.org/10.3390/su17178049>
- Trovo, H., Mamede, H. S., Trigo, P., & Santos, V. (2025). Artificial intelligence in recruitment: A multivocal review of benefits, challenges, and strategies. *Engineering Studies Journal*, 9(6), 030. <https://doi.org/10.28991/esj-2025-09-06-030>

- Vakharia, V., Shah, M., Nair, P., Borade, H., Sahlot, P., & Wankhede, V. A. (2023). Estimation of lithium-ion battery discharge capacity by integrating optimized explainable-AI and stacked LSTM model. *Batteries*, 9(2), 125.
- Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer-Verlag.
- Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, 2018, 7068349. <https://doi.org/10.1155/2018/7068349>
- Vyas, B. (2023). Probabilistic reasoning models in artificial intelligence: Exploring probabilistic reasoning models and their applications in solving complex AI problems. *Journal of Artificial Intelligence Research*, 3(1), 14–24. <https://doi.org/10.5281/zenodo.7865965>
- Wang, C., Anisuzzaman, D., Williamson, V., Dhar, M. K., Rostami, B., Niezgoda, J., Gopalakrishnan, S., & Yu, Z. (2020). Fully automatic wound segmentation with deep convolutional neural networks. *Scientific Reports*, 10, 78799.
- Wang, D., Wan, B., Qiu, P., Su, Y., Guo, Q., & Wu, X. (2018). Artificial mangrove species mapping using Pleiades-1: An evaluation of pixel-based and object-based classifications with selected machine learning algorithms. *Remote Sensing*, 10(2), 294. <https://doi.org/10.3390/rs10020294>
- Wang, J. (2024). Studies on signal processing and visual computing applications of deep learning. 2024 IEEE International

- Conference on Advanced Machine Learning Technologies and Applications (ICAML), 64299.
- Wang, L., Hu, L., Gu, J., Wu, Y., Hu, Z., He, K., & Hopcroft, J. (2018). Towards understanding learning representations: To what extent do different neural networks learn the same representation. *ArXiv*, 1810.11910.
- Wang, L., Pan, H., & Cai, Z. (2023). Novel hybrid model coupling WOA with BiLSTM neural networks and attention mechanisms. *2023 International Conference on Intelligent Public Management, Medicine and Health (ICIPCA)*, 10258008.
- Wu, D., Guan, Q., Fan, Z., Deng, H., & Wu, T. (2023). AutoML with parallel genetic algorithm for fast hyperparameters optimization in efficient IoT time series prediction. *IEEE Transactions on Industrial Informatics*, 19(2), 1411–1421.
- Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., & Zhang, C. (2020). Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 753–763). <https://doi.org/10.1145/3394486.3403118>
- Xu, K., Shen, X., Yao, T., Tian, X., & Mei, T. (2018). Greedy layer-wise training of long short term memory networks. *2018 IEEE International Conference on Multimedia and Expo (ICMEW)*, 1–6.

- Xue, Y., Jiang, P., Neri, F., & Liang, J. (2021). A multi-objective evolutionary approach based on graph-in-graph for neural architecture search of convolutional neural networks. *International Journal of Neural Systems*, 31(7), 2150035.
- Yamini, B., Prasanna, V., Ambhika, C., M, A., Maheswari, B., R, S., & Nalini, M. (2023). A comprehensive survey of deep learning: Advancements, applications, and challenges. *International Journal on Recent and Innovation Trends in Computing and Communication*, 11(8), 190–208.
- Yan, B., Zhang, X., Zhang, L., Zhang, L., Zhou, Z., Miao, D., & Li, C. (2025). Beyond self-talk: A communication-centric survey of LLM-based multi-agent systems. *ArXiv Preprint ArXiv:2502.14321*.
- Yang, Y., Zhuang, Y., & Pan, Y. (2021). Multiple knowledge representation for big data artificial intelligence: Framework, applications, and case studies. *Frontiers in Information Technology and Electronic Engineering*, 22(12), 463. <https://doi.org/10.1631/FITEE.2100463>
- Yao, J. F., Huang, Y. H. J., Yang, C. Y., & Hwang, M. S. (2024). Deep learning applications. 2024 IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPACS), 62486.
- Yin, L., Fuentes, A., Yoon, S., Kim, S. R., & Park, D. J. (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors*, 17(9), 2022.

- Zhang, J., & Lyu, L. (2024). A spacecraft onboard autonomous task scheduling method based on hierarchical task network–timeline. *Aerospace*, 11(5), 350. <https://doi.org/10.3390/aerospace1105035>
- Zhang, J., Ye, L., & Lai, Y. (2023). Stock price prediction using CNN-BiLSTM-attention model. *Mathematics*, 11(9), 1985. <https://doi.org/10.3390/math11091985>
- Zhang, K., Zuo, W., Chen, Y., Meng, D., & Zhang, L. (2017). Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7), 3142–3155.
- Zhang, W., Çerban, O., Sun, J., & Guo, Y. (2023). IPPT4KRL: Iterative post-processing transfer for knowledge representation learning. *Machine Learning and Knowledge Extraction*, 5(1), 004. <https://doi.org/10.3390/make5010004>
- Zhang, Y., Saxe, K., & Nori, H. (2024). The evolution of artificial intelligence: Machine learning and deep learning trends. *IEEE Transactions on Emerging Topics in Computing*, 12(4), 789–805.
- Zhang, Y., Zhou, Y., & Fujita, H. (2024). Distributed multi-agent reinforcement learning for cooperative low-carbon control of traffic network flow using cloud-based parallel optimization. *IEEE Transactions on Intelligent Transportation Systems*, 3452430.
- Zhang, Z., & Sun, G. (2019). Caffeine: Toward uniformed representation and acceleration for deep convolutional neural

networks. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 37(11), 2411–2424.

Zuazua, E. (2025). Machine learning and control: Foundations, advances, and perspectives. ArXiv Preprint ArXiv:2510.03303.

Buku ajar yang berjudul **Kecerdasan Buatan dan Pengembangan Sistem Cerdas** membahas secara komprehensif konsep dasar hingga penerapan kecerdasan buatan (Artificial Intelligence) dalam berbagai bidang kehidupan modern. Buku ini menguraikan prinsip kerja AI, mulai dari pengolahan data, pembelajaran mesin (machine learning), hingga pengambilan keputusan otomatis yang digunakan dalam sistem cerdas. Pembaca juga diperkenalkan pada berbagai algoritma dan teknik yang menjadi fondasi dalam pengembangan teknologi berbasis kecerdasan buatan.

Selain aspek teoritis, buku ini juga menekankan penerapan praktis dalam pengembangan sistem cerdas seperti Internet of Things (IoT), robotika, sistem rekomendasi, dan otomasi industri. Dengan pendekatan yang sistematis dan aplikatif, buku ini membantu pembaca memahami bagaimana kecerdasan buatan dapat diimplementasikan untuk meningkatkan efisiensi dan efektivitas berbagai sistem. Buku ini diharapkan menjadi referensi bagi masyarakat umum.