

## BAB II

### LANDASAN TEORI

#### 2.1. Tinjauan Jurnal

Pada tahapan ini, dilakukan peninjauan kembali berdasarkan jurnal dan penelitian-penelitian yang sebelumnya telah dilakukan, diantaranya:

Penelitian dari Nadya Andhika Putri(2018) yang berjudul “Sistem Pakar Untuk Mengidentifikasi Kepribadian Siswa Menggunakan Metode *Certainty Factor* dalam Mendukung Pendekatan Guru”, menyebutkan bahwa salah satu faktor yang dapat menghambat siswa dalam memahami materi yaitu proses belajar mengajar yang kurang menarik. Keterampilan seorang guru dalam memahami kepribadian siswanya dapat berdampak baik sehingga dapat menentukan perlakuan dalam mengoptimalkan pengembangan bakat, minat dan potensi yang dimiliki. Penggunaan metode *Certainty Factor* dalam penelitian ini bertujuan untuk proses mengidentifikasi kepribadian siswa berdasarkan bobot dari nilai tingkat keyakinan dan solusi perbaikannya, sehingga dapat diperoleh solusi bagi setiap guru dalam melakukan pendekatan kepada siswanya guna menyampaikan materi pembelajaran secara efektif.

Selanjutnya, penelitian yang berjudul “Sistem Pakar Hukum Pidana Pencurian Menggunakan Algoritma *Fuzzy Decision Table* Berbasis Android” oleh Agung Wibowo, Rival Afrian, dan Saeful Bahri (2018) yang mengimplementasikan konsep sistem pakar di bidang hukum pidana materiil, khususnya dalam kasus pencurian sesuai kitab undang-undang yang berlaku. Hasil penelitian ini berupa aplikasi android yang dapat digunakan sebagai referensi konsultasi pada hukum kasus pencurian.

Kemudian Imran Djafar, Alders Paliling dan Asrul Syam (2019) dengan judul “Implementasi Metode *Certainty Factor* Untuk Aplikasi Kebijakan Sistem Pointing Monitoring Pelanggaran Siswa Pada SMA Negeri 1 Baras Mamuju Utara” yang memanfaatkan metode tersebut dalam pembangunan aplikasi untuk memantau perilaku perkembangan siswa. Dengan adanya penggunaan aplikasi tersebut, diharapkan dapat meningkatkan kinerja guru BK khususnya dalam memantau kegiatan siswa serta dapat merekapitulasi hasil monitoring siswa sebagai referensi dimasa akan datang.

Terakhir, Agustinus Gunawan Siregar (2019) dalam penelitiannya yang berjudul “Sistem Pakar Menentukan Tingkat Depresi Pekerja Yang di PHK Menggunakan Metode *Certainty Factor*”, menjelaskan bahwa penelitian ini

dilakukan sebagai salah satu upaya konsultasi bagi para pekerja dalam menentukan deteksi dini tingkat depresi akibat PHK. Penggunaan metode *Certainty Factor* untuk mengidentifikasi dan menganalisis gejala depresi yang dialami sehingga dapat ditemukan solusi yang tepat dalam mengatasi depresi tersebut.

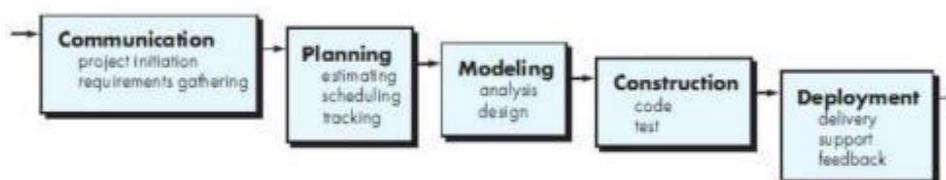
Berdasarkan penelitian-penelitian tersebut, maka akan dilaksanakan penelitian sejenis seperti yang telah dilakukan oleh Agung Wibowo, Rival Afrian, dan Saeful Bahri yaitu memanfaatkan penerapan ilmu komputer dengan konsep sistem pakar pada sektor bidang hukum, namun yang berbeda disini adalah pembahasan hukum yang digunakan yakni berupa undang-undang dalam perlindungan hak-hak dasar pekerja/buruh dan penggunaan metode *Certainty Factor* seperti 3 penelitian yang telah dijelaskan sebelumnya, serta diimplementasikan dalam pemrograman berbasis web supaya dapat memperoleh informasi yang cepat dan fleksibel dalam upaya konsultasi bagi para pekerja/buruh.

## 2.2. Konsep Dasar Program

### A. Teori Model *Waterfall*

Menurut Pressman (2015:42) menyimpulkan bahwa:

Model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun software (*Linear Sequential Model*) dan termasuk ke dalam model generik pada rekayasa perangkat lunak yang pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE).



Sumber : Pressman (2015:42)

**Gambar II.1.**

**Fase-Fase dalam *Waterfall***

1. *Communication*

Proses komunikasi sangat diperlukan demi memahami dan mencapai tujuan yang diinginkan konsumen. Hasil dari proses ini yaitu berupa inisialisasi proyek, seperti melakukan analisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta identifikasi fitur dan fungsi software.

2. *Planning*

Tahapan ini menjelaskan mengenai estimasi tugas-tugas teknis yang akan dilakukan, resiko yang dapat terjadi, sumber daya yang diperlukan dalam pembuatan sistem, produk kerja yang ingin dihasilkan, penjadwalan, dan *tracking* proses pengerjaan sistem.

3. *Modeling*

Fase *Modeling* yaitu tahap pemodelan arsitektur yang berfokus pada perancangan struktur data, arsitektur software, tampilan interface, dan algoritma program untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.

4. *Construction*

Tahapan *Construction* adalah proses interpretasi dari bentuk desain menjadi kode atau bentuk/bahasa yang dapat dibaca oleh mesin. Setelah pengkodean selesai, pengujian terhadap sistem dan *coding* dilakukan yang bertujuan untuk menemukan dan memperbaiki kesalahan yang mungkin terjadi.

5. *Deployment*

Tahapan ini merupakan tahapan implementasi software ke konsumen, *maintenance* software secara berkala, perbaikan dan evaluasi software, dan pengembangan software berdasarkan *feedback* yang diberikan supaya sistem

dapat tetap berjalan dan berkembang sesuai dengan fungsinya (Pressman, 2015:17).

## B. Teori Certainty Factor

Faktor kepastian (*Certainty Factor*) diperkenalkan oleh Shortlife Buchanan yang merupakan nilai parameter klinis untuk menunjukkan besarnya kepercayaan pada pembuatan MYCIN (Djafar, Paliling dan Syam, 2019:3). Metode ini biasanya digunakan pada suatu masalah yang jawabannya tidak pasti (Halim dan Hansun, 2015:16). Metode *Certainty Factor* memanfaatkan penggunaan suatu nilai untuk mengasumsikan derajat keyakinan seorang pakar terhadap suatu data (Darnita dan Muntahanah, 2018:177). Metode tersebut didefinisikan dalam persamaan berikut ini (Suryadi dan Murdani, 2017:23):

$$CF[h,e] = MB[h,e] - MD[h,e] \dots \dots \dots (1)$$

Dimana :

CF[h,e] : Faktor Kepastian.

MB[h,e] : Ukuran kepercayaan terhadap hipotesis h , jika diberikan evidence e ( antara 0 dan 1 ).

MD[h,e] : Ukuran ketidakpercayaan terhadap evidence h,jika diberikan evidence e (antara 0 dan 1).

Berikut ini kondisi beberapa kombinasi *Certainty Factor* terhadap berbagai kondisi (Siregar, 2019:62):

1. CF untuk kaidah dengan premis tunggal (*single premis rules*):

$$CF (H,E) = CF (E) * CF (rule) = CF (user) * CF (pakar)$$

2. CF untuk kaidah dengan premis majemuk (*multiple premis rules*)

$$CF (A \text{ AND } B) = \text{Minimum } (CF (a), CF (b) * CF (rule)$$

$$CF (A \text{ OR } B) = \text{Maksimum } (CF (a), CF (b) * CF (rule)$$

3. CF untuk kaidah dengan kesimpulan yang serupa (similarly concluded ruler)

$$\text{CF combine (CF1, CF2)} = \text{CF1} + \text{CF2} * (1 - \text{CF1})$$

Kelebihan dari metode ini yaitu cocok digunakan untuk sistem pakar yang mengukur sesuatu yang kepastian atau ketidakpastiannya dengan perhitungan yang hanya berlaku untuk sekali hitung saja sehingga mudah diterapkan dan hanya dapat mengolah dua data sehingga keakuratannya terjaga (Putri, 2018:81).

#### C. Web Server

Web Server merupakan software dalam server yang berfungsi untuk menerima permintaan berupa halaman web melalui HTTP atau HTTPS dari klien dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML (Marisa, 2017:142).

#### D. Website

Website merupakan salahsatu media penting dimana yang dapat mencari semua berbagai informasi dan merupakan salah satu bentuk media massa yang dipublikasi melalui jaringan internet serta dapat diakses dimanapun dan kapanpun (Manik, Salamah, Susanti, 2017:478).

#### E. PHP

PHP atau *Hypertext Preprocessor* yaitu suatu bahasa pemrograman yang digunakan untuk menginterpretasikan *line* kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang bersifat *server-side* (Supono dan Putratama, 2016:3).

#### F. MySQL

MySQL adalah sebuah perangkat lunak yang terdapat didalam sistem manajemen basis data atau biasa disebut DBMS (*Database Management System*)

yang multithread, multiuser dengan sekitar 6 juta instalasi di seluruh dunia (Dhika, Isnain, Tofan, 2019:107).

G. Perundang-Undangan Hak Pekerja/Buruh

1. Pasal 78 ayat (2) Undang-Undang Nomor 13 Tahun 2003 Tentang ketenagakerjaan, menjelaskan sebagai berikut:

“Pengusaha yang mempekerjakan pekerja/buruh melebihi waktu kerja sebagaimana dimaksud dalam ayat (1) wajib membayar upah kerja lembur.”

2. Pasal 79 ayat (1) Undang-Undang Nomor 13 Tahun 2003 Tentang ketenagakerjaan, menjelaskan sebagai berikut:

“Pengusaha wajib memberi waktu istirahat dan cuti kepada pekerja/buruh.”

3. Pasal 80 Undang-Undang Nomor 13 Tahun 2003 Tentang ketenagakerjaan, menjelaskan sebagai berikut:

“Pengusaha wajib memberikan kesempatan yang secukupnya kepada pekerja/buruh untuk melaksanakan ibadah yang diwajibkan oleh agamanya.”

4. Pasal 86 ayat (1) Undang-Undang Nomor 13 Tahun 2003 Tentang ketenagakerjaan, menjelaskan sebagai berikut:

“Setiap pekerja/buruh mempunyai hak untuk memperoleh perlindungan atas: keselamatan dan kesehatan kerja; moral dan kesusilaan; dan perlakuan yang sesuai dengan harkat dan martabat manusia serta nilai-nilai agama.

5. Pasal 90 ayat (1) Undang-Undang Nomor 13 Tahun 2003 Tentang ketenagakerjaan, menjelaskan sebagai berikut:

“Pengusaha dilarang membayar upah lebih rendah dari upah minimum sebagaimana dimaksud dalam Pasal 89.”

## H. Peralatan Pendukung Sistem (*Tools System*)

### 1. ERD (*Entity Relationship Diagram*)

ERD (*Entity Relationship Diagram*) adalah bentuk paling awal dalam melakukan perancangan basis data resional (Rosa A.S dan Salahudin, 2015:50). ERD juga merupakan sebuah gambar diagram yang dapat menunjukkan informasi yang dibuat, disimpan dan digunakan dalam sistem bisnis (Dhika, Isnain, Tofan, 2019:107).

### 2. LRS (*Logical Record Structure*)

Menurut Tabrani (2014:35) menyebutkan bahwa: *Logical Record Structure* terdiri dari link-link diantara tipe record. Link ini menunjukkan arah dari satu tipe record lainnya. Banyak link dari LRS yang diberi tanda field-field yang kelihatan pada kedua link tipe record. Penggambaran LRS mulai dengan menggunakan model yang dimengerti. Dua metode yang dapat digunakan, dimulai dengan hubungan kedua model yang dapat dikonverensikan ke LRS, metode yang lain dimulai dengan ER-diagram dan langsung dikonversikan ke LRS.

### 3. UML (*Unified Modeling Language*)

Menurut Rosa A.S dan Shalahuddin (2013:137) mengemukakan bahwa” *Unified Modeling Language* (UML) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung”. UML merupakan bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek dengan pemodelan yang digunakan untuk penyederhanaan permasalahan permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami (Djafar, Paliling dan Syam, 2019:5)

Dalam penulisan laporan ini, ada beberapa diagram UML yang digunakan diantaranya *use case diagram*, *activity diagram*, *class diagram*, *sequence diagram*, *component diagram* dan *deployment diagram*.

a. *Use Case Diagram*

*Use Case Diagram* bersifat statis dan menunjukkan fungsionalitas suatu sistem atau kelas tentang bagaimana sistem ini memahami bagaimana sistem seharusnya bekerja dan berinteraksi dengan dunia luar, contohnya penyusunan sebuah daftar pada layanan kesehatan (Prameswari dan Anwar, 2018:270).

b. *Activity Diagram*

*Activity diagram* bersifat dinamis dan merupakan diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis (Siregar, Siregar dan Melani, 2018:114).

c. *Class Diagram*

*Class diagram* merupakan visualisasi kelas-kelas dari suatu sistem dan tipe diagram yang paling banyak dipakai karena dapat memperlihatkan relasi antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain (*logical view*) dari suatu sistem (Prameswari dan Anwar, 2018:271). Didalam setiap *class* terdapat 3 area utama yaitu nama, atribut, dan operasi yang berfungsi sebagai identitas, karakteristik fungsi objek.

d. *Sequence Diagram*

*Sequence diagram* merupakan salah satu jenis diagram UML yang menggambarkan interaksi antar objek di dalam dan disekitar sistem, termasuk pengguna, display, dan sebagainya berupa message yang digambarkan terhadap waktu (Syarif dan Nugraha, 2020:65).

e. *Component Diagram*

Component diagram menggambarkan struktur fisik dari kode, pemetaan pandangan logis dari kelas proyek untuk kode aktual di mana logika ini dilaksanakan (Irawan dan Simargolang, 2018:78).

f. *Deployment Diagram*

*Deployment diagram* merupakan diagram yang menunjukkan gambaran dari arsitektur fisik perangkat lunak, perangkat keras, dan artefak dari sistem dan dapat dianggap sebagai ujung spektrum dari kasus penggunaan, serta menggambarkan bentuk fisik dari sistem yang bertentangan dengan gambar konseptual dari pengguna dan perangkat berinteraksi dengan sistem (Siregar, Siregar dan Melani, 2018:114).



**UNIVERSITAS**