

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

2.1.1 Pengertian Wisuda

Wisuda adalah upacara peneguhan atau pelantikan bagi seseorang yang telah menempuh pendidikan. Di kalangan akademik, wisuda merupakan penanda kelulusan mahasiswa yang telah menempuh masa belajar pada suatu universitas. Biasanya prosesi wisuda diawali prosesi masuknya senat universitas yang terdiri dari rektor dan para pembantu rektor dengan dekan-dekannya guna mewisuda para calon wisudawan. Dalam menyelenggarakan wisuda, tiap perguruan tinggi memiliki agenda yang tidak sama. Ada yang dilakukan setiap tahun, tetapi ada juga yang setiap semester, menyesuaikan kalender akademik.

Pada umumnya, calon wisudawan mengenakan pakaian yang sudah ditentukan, seperti pakaian pria menggunakan hem putih dan celana hitam bersepatu hitam, pakaian wanita menggunakan kebaya tradisional tipis dengan kain batik dan bagian luarnya mengenakan toga.

2.1.2 Pengertian Sistem

Secara sederhana suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsu, komponen atau variabel-variabel yang terorganisir, saling berinteraksi, saling ketergantungan satu sama lainnya dan terpadu. Sistem juga merupakan suatu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang berinteraksi untuk mencapai tujuan.

Ada beberapa pakar yang berlainan pendapat dalam mendefinisikan suatu sistem namun tetap pada satu tujuan.

“sistem adalah kumpulan/grup dari subsistem/bagian/komponen apapun, baik fisik ataupun nonfisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu” (Djahir dan Pratita, 2015). Sedangkan menurut (Mulyani, 2017) menyatakan bahwa “sistem bisa diartikan sebagai sekumpulan sub sistem, komponen yang saling bekerja sama dengan tujuan yang sama untuk menghasilkan output yang sudah ditentukan sebelumnya”.

“sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu” (Jeperson Hutahaeen, 2015)

2.1.3 Karakteristik Sistem

Menurut (Hutahaeen, 2015) Suatu sistem mempunyai karakteristik antara lain:

1. Komponen Sistem (*Component*)

Suatu Sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat-sifat sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batas Sistem (*Boundary*)

Batas Sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luar.

3. Lingkungan Luar Sistem (*Environment*)

Lingkungan Luar Sistem (*Environment*) dari suatu sistem apapun diluar batas dari sistem yang mempengaruhi operasi sistem.

4. Penghubung Sistem (*Interface*)

Penghubung Sistem (*Interface*) merupakan media penghubung antara satu sub sistem dengan sub sistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu sub sistem ke sub sistem yang lainnya.

5. Masukan Sistem (*Input*)

Masukan Sistem (*Input*) adalah energi yang dimasukkan ke dalam sistem.

Masukan dapat berupa *maintenance input* dan *signal input*.

6. Keluaran Sistem (*Output*)

Keluaran Sistem (*Output*) adalah hasil dari energi yang diolah dan disiklasifikasikan menjadi keluaran yang berguna.

7. Pengolahan Sistem

Pengolahan sistem adalah bagian yang bertugas mengolah atau mengubah masukan menjadi keluaran.

8. Sasaran Sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*).

Sasaran dari sistem sangat menentukan sekali maukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem dan dikatakan berhasil jika mengenai sasaran dan tujuannya.

2.1.4 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut:

1. Sistem Abstrak dan Sistem Fisik

Sistem abstrak adalah sistem yang berupa pemikiran ide-ide yang tidak tampak secara fisik. Sedangkan Sistem Fisik adalah merupakan sistem yang ada secara fisik.

2. Sistem Alamiah dan Sistem Buatan Manusia

Sistem Alamiah adalah sistem yang terjadi melalui proses alam dan tidak dibuat manusia. Sedangkan Sistem Buatan Manusia merupakan sistem yang dirancang oleh manusia dan melibatkan interaksi antara manusia dengan mesin.

3. Sistem Deterministik dan Sistem Probalistik

Sistem Deterministik adalah sistem yang tingkah lakunya dapat diprediksi. Sedangkan Sistem Probalistik merupakan sistem yang kondisi masa depannya tidak dapat diprediksi, karena mengandung unsur probabilitas.

4. Sistem Terbuka dan Sistem Tertutup

Sistem Terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya. Sedangkan Sistem Tertutup merupakan sistem yang tidak berhubungan dan tidak dipengaruhi oleh lingkungan luarnya.

2.1.5 Pengertian Informasi

Informasi adalah data-data yang telah diolah sehingga dapat berguna bagi siapa saja yang membutuhkan. Informasi dapat direkam atau dikirim. Para ahli memiliki banyak arti lain tentang informasi. Informasi bisa dikatakan sebagai pengetahuan yang didapatkan dari belajar, pengalaman atau instruksi. (Effendy & Hasan, 2016)

Lalu menurut Kusriani dalam (Effendy & Hasan, 2016) menjelaskan bahwa “Informasi dapat didefinisikan sebagai hasil dari olahan data menjadi

sebuah bentuk yang bermakna yang digunakan untuk pengambilan keputusan”.

Sedangkan menurut Bodnar dalam (Effendy & Hasan, 2016) berpendapat bahwa “Informasi adalah data yang diolah sehingga dapat dijadikan dasar untuk mengambil keputusan yang tepat”.

2.1.6 Pengertian Sistem Informasi

Menurut Sutabri dalam (Lubis, 2016) menyatakan bahwa:
Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.

Menurut Jogiyanto dalam (Effendy & Hasan, 2016) menerangkan bahwa “Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan”.

Menurut Azar Susanto dalam (Effendy & Hasan, 2016) mengemukakan bahwa “Sistem informasi merupakan kumpulan dari subsistem apapun baik fisik ataupun nonfisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai suatu tujuan yaitu mengolah data menjadi informasi yang berarti dan berguna”.

Menurut Tata Sutabri dalam (Siregar et al., 2019) sistem informasi terdiri dari komponen-komponen antara lain sebagai berikut:

a. Blok Masukan (*input block*)

Mewakili data yang masuk kedalam sistem informasi. Input yang dimaksud adalah metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

b. Blok Model (*model block*)

Blok ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

c. Blok Keluaran (*output block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkat manajemen serta semua pemakai sistem.

d. Blok Teknologi (*technology block*),

Teknologi merupakan “*tool box*” dalam sistem informasi.

e. Blok Basis Data (*database block*),

Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu sama lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya.

f. Blok Kendali (*control block*)

Banyak hal yang dapat merusak sistem informasi, seperti bencana alam, api, temperatur, air, debu, kecurangankecurangan, kegagalan-kegagalan sistem itu sendiri, ketidak efisienan, sabotase, dan lain sebagainya.

2.1.7 Website

Pada dasarnya website adalah kepanjangan dari *World Wide Web* (WWW). Informasi WWW ini disimpan pada web server untuk dapat diakses dari jaringan browser terlebih dahulu, seperti *Internet Explorer* atau *Mozilla Firefox*.

Menurut Kustiyaningsih dan Devie Rosa berpendapat bahwa “Jenis website dapat dikategorikan menjadi dua yaitu web statis dan web dinamis”

(Mahdiati, 2016). Web statis adalah web yang menampilkan informasi-informasi yang sifatnya statis (tetap). Disebut statis karena pengguna tidak dapat berinteraksi dengan web tersebut. Sedangkan web dinamis adalah web yang menampilkan informasi serta dapat berinteraksi dengan pengguna. Web dinamis memungkinkan pengguna untuk berinteraksi menggunakan form sehingga dapat mengolah informasi yang ditampilkan.



Gambar II.1
website

2.1.8 Internet

Menurut Siregar dalam (Sutopo, Cahyadi, & Arifin, 2016) mendefinisikan: *Internet* atau *Interconnected Networking* merupakan dua komputer atau lebih yang saling berhubungan membentuk jaringan komputer hingga meliputi jutaan komputer di dunia, yang saling berinteraksi dan bertukar informasi, terhadap masalah sosial misalnya dengan menggunakan alat-alat bantu online untuk mencapai bisnis elektronik (e-commerce), kepemilikan informasi dan interaksi dengan masyarakat. (Sutopo et al., 2016)

Menurut Siregar dalam (Sutopo et al., 2016) ada beberapa fasilitas dari internet yang sering kita temui bahkan digunakan antara lain:

a. E-mail

Atau surat elektronik adalah suatu sarana untuk mengirim dan menerima surat atau pesan dengan format digital melalui jalur jaringan komputer dan internet.

b. *World Wide Web (WWW)*

adalah suatu ruang informasi di internet yang digunakan oleh pengenalan global yang disebut dengan Pengidentifikasi Sumber Ragam / URL untuk mengenali sumber daya berguna. World Wide Web berisi kumpulan peladen web dari berbagai seluruh bagian dunia untuk menyediakan informasi dan data yang bisa digunakan bersama.

c. *Newsgroup*

adalah suatu tempat penyimpanan yang biasanya dipakai oleh *usenet* (sistem diskusi internet yang tersebar) untuk berbagai pesan yang dikirimkan dari banyak pengguna pada lokasi-lokasi yang berlainan. Kadang istilah ini membingungkan, sebab kebanyakan orang memakai newsgroup untuk diskusi kelompok. Perangkat lunak yang dipakai untuk membaca newsgroup yaitu *Newsreader*.

d. Telnet

Merupakan suatu protokol client-server yang memungkinkan kita untuk mengakses komputer dari jarak jauh. Telnet yang biasa disingkat TN ini menjadi semacam emulasi terminal yang memakai protokol untuk mengakses komputer secara remote. Fungsinya yakni memungkinkan kita untuk mengakses sebuah komputer secara remote access atau pengendalian jarak jauh. Ia merupakan program yang memungkinkan komputer anda dijadikan terminal dari komputer satu ke komputer lainnya, tentunya dengan koneksi jaringan internet.

e. *Chat*

Adalah suatu feature atau suatu program di internet untuk berkomunikasi langsung sesama pengguna internet yang sedang online / yang sedang samasama menggunakan internet. Komunikasi ini dapat berupa teks (*text chat*) ataupun suara (*voice chat*).

f. *File Transfer Protocol (FTP)*

Adalah suatu protokol yang berfungsi untuk tukar-menukar *file* dalam suatu network yang menggunakan TCP koneksi bukan UDP. Dua hal yang penting dalam FTP adalah *FTP Server* dan *FTP Client*. FTP server adalah suatu server yang menjalankan software yang berfungsi untuk memberikan layanan tukar menukar *file* dimana server tersebut selalu siap memberikan layanan FTP apabila mendapat permintaan (*request*) dari *FTP client*.



Gambar II.2
Internet (gmail)

2.1.9 *Web Server*

Web server adalah software yang memberikan layanan data yang berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan browser web dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML. Fungsi

utama sebuah web server adalah untuk mentransfer berkas atas permintaan pengguna melalui protokol komunikasi yang telah ditentukan. Hal ini disebabkan sebuah halaman web dapat terdiri atas berkas teks, gambar, video dan yang lainnya. Server HTTP Apache atau *server web* / WWW Apache adalah *server web* yang dapat dijalankan di banyak sistem operasi (Unix, BSD, Linux, Microsoft Windows, dan Novell Netware serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web / WWW ini menggunakan HTTP. (Febrian, Kusriani, & Arief, 2016)



Gambar II.3
Web server (apache, microsoft)

2.1.10 Basis Data

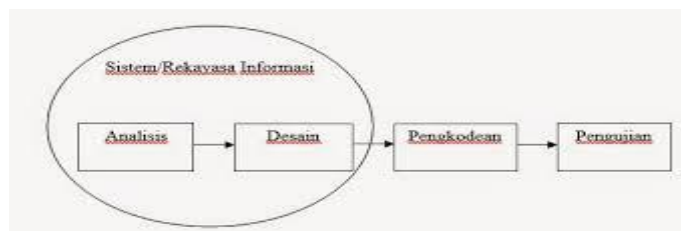
Menurut (Fathansyah, 2015) “Basis data terdiri dari 2 kata yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat berkumpul. Sedangkan Data adalah representasi fakta dunia nyata mewakili suatu objek seperti manusia (pegawai, siswa, pembelian pelanggan), barang hewan, peristiwa, konsep, keadaan, dan sebagian yang berwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya”.

2.1.11 Model Pengembangan Perangkat Lunak

Menurut (A.S & Shalahuddin, 2018) menyatakan bahwa “Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential*

linier) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, dan tahap pendukung (*support*)”.

Berikut adalah gambar model air terjun (*waterfall*) menurut Rosa A.S & M.Shalahuddin:



Sumber: (A.S & Shalahuddin, 2018)

Gambar II.4

Model air terjun (*waterfaal*)

Penjelasan tahapan-tahapan dari model air terjun (*waterfall*) ini adalah sebagai berikut:

a. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

b. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap

selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

c. Pembuatan Kode Program

Pada tahap ini penulis tidak melakukan pengkodean program, karena penulis hanya merancang dan membuat desain menggunakan *software HTML* dan *CSS*, Setelah desain dibuat kedalam sistem pengolahan data sebagai rancangan program agar hasilnya sesuai dengan desain yang telah dibuat.

d. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

e. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

Model air terjun (*waterfall*) sangat cocok digunakan. Kebutuhan pelanggan sudah sangat dipahami dan kemungkinan terjadinya perubahan kebutuhan selama pengembangan perangkat lunak kecil. Hal positif dari model air terjun (*waterfall*) adalah struktur tahap pengembangan sistem jelas, dokumentasi dihasilkan di setiap tahap pengembangan, dan sebuah tahap dijalankan setelah tahap sebelumnya selesai

dijalankan (tidak ada tumpang tindih dalam pelaksanaan tahap). (A.S & Shalahuddin, 2018).

2.2 Teori Pendukung

Berikut ini adalah teori-teori pendukung yang digunakan dalam pembuatan website.

A. ERD (*Entity Relationship Diagram*).

1. Definisi ERD

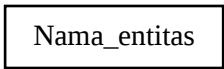
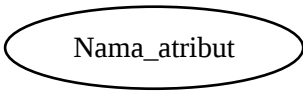
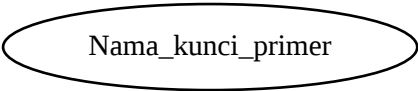
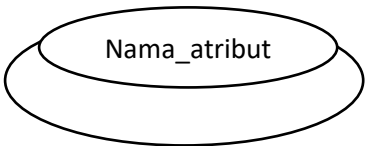
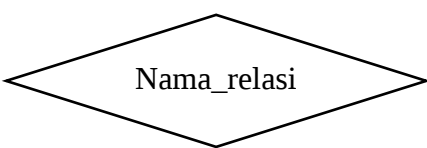
Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram* (ERD). Menurut (A.S & Shalahuddin, 2018) mengemukakan bahwa “ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD”.

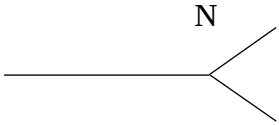
OODBMS merupakan *database management system* (DBMS) yang mendukung pemodelan dan pembuatan data sebagai objek.

Di dalam ERD memiliki beberapa aliran notasi yang dikembangkan. Di antaranya adalah notasi Chen yang dikembangkan oleh Peter Chen, notasi Barker yang dikembangkan oleh Richard Barker, Ian Palmer, dan Harry Ellis, dan juga notasi Crow’s Foot serta beberapa notasi lain. Namun yang paling banyak digunakan adalah notasi dari Chen. (A.S & Shalahuddin, 2018).

Berikut adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen: (A.S & Shalahuddin, 2018)

Tabel II.1.
Simbol-Simbol erd notasi Chen

SIMBOL	KETERANGAN
Entitas / Entity 	Entitas merupakan data inti yang akan disimpan; bakal table pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
Atribut 	Field atau kolom data yang butuh disimpan dalam suatu entitas.
Atribut kunci primer 	Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).
Atribut multinilai / multivalue 	Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.

Asosiasi / association 	Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki multiplicity kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan one to many.
---	---

Sumber : (A.S & Shalahuddin, 2018)

2. Komponen ERD

Menurut Fatta dalam (Fauzi, Erniawati, & Setyawan, 2019) mengemukakan bahwa “Didalam ERD terdapat komponen-komponen khusus untuk menggambarkan elemen-elemen ERD”. Antara lain yaitu:

a. Entitas

Entitas adalah objek yang harus menampilkan beberapa kali event untuk menjadi sebuah entitas.

b. Atribut

Atribut adalah informasi dari sebuah entitas yang merupakan kata benda dan hanya digunakan oleh organisasi yang dimasukan dalam model.

c. Identifier

Identifier adalah satu atau lebih atribut yang dapat menjadi identifier entitas secara unik mengidentifikasi setiap anggota dari entitas.

d. Kardinalitas

Kardinalitas adalah hubungan antara suatu *instance* pada entitas dapat berelasi dengan instance lain di instance yang berbeda.

e. Modalitas

Modalitas dapat dikatakan suatu instance dari entitas anak dapat ada tanpa suatu relasi dengan instance dari entitas induk atau tidak.

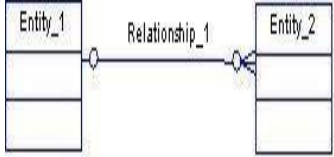
3. Pengertian Logical Record Structured (LRS)

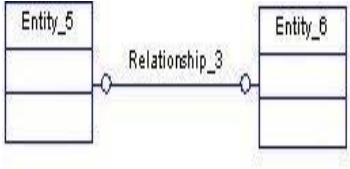
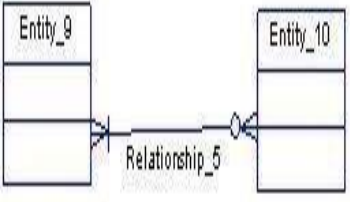
Menurut Supardi berpendapat bahwa “LRS merupakan hasil dari Entity Relationship (ER) beserta atributnya sehingga bisa terlihat hubungan-hubungan antara entitas”(Suryanto, 2016).

Dalam kaitannya dalam konversi ke LRS, maka perubahan yang terjadi adalah mengikuti aturan-aturan berikut ini: (Hairil & Widjaja, 2018)

- a. Setiap entitas akan diubah ke dalam bentuk kotak.
- b. Sebuah relationship kadang di satukan dalam sebuah kotak bersama entitas jika dihubungkan yang terjadi pada ERD memiliki tingkat hubungan *one to many* (relasi bersatu dengan *cardinality*) atau tingkat hubungan *one to one* (relasi bersatu dengan *cardinality* yang paling membutuhkan referensi), dan kadang sebuah relasi dipisah dalam sebuah kotak tersendiri jika tingkat hubungannya *many to many*.

Tabel II.2.
Simbol-simbol LRS

No	SIMBOL	NAMA	KETERANGAN
1.		<i>One to Many</i>	Tingkat Hubungan satu ke banyak adalah sama dengan banyak ke satu. Tergantung dari arah

			mana hubungan tersebut dilihat.
2.		<i>One to One</i>	Tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai hubungan dengan satu kejadian pada entitas yang kedua atau sebaliknya.
3.		<i>Many to Many</i>	Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya. Baik dilihat dari sisi entitas yang

			pertama, maupun dilihat dari sisi kedua.
--	--	--	---

Sumber: (Fauzi, Erniawati, & Setyawan, 2019)

B. UML (*Unified Modelling Language*)

1. Definisi UML (*Unified Modelling Language*)

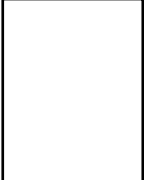
Menurut (Sukamto dan Shalahuddin (2015:154), 2019) “*Unified Model Language* (UML) merupakan Bahasa *visual* untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan”.

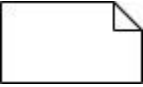
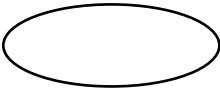
“UML adalah sebuah bahasa pemodelan yang menjadi standar dalam industri *software* untuk pembayangan, merancang, dan mengarsipkan sistem perangkat lunak yang akan dibuat”.

Dari pengertian UML (*Unified Modeling Language*) dapat disimpulkan UML (*Unified Modeling Language*) adalah bahasa pemodelan untuk sistem atau perangkat lunak dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung yang berparadigma berorientasi objek.

Tabel II.3.
Simbol (*Unified Modelling Language*) UML

NO	SIMBOL	NAMA	KETERANGAN
1.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .

2.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>)
3.		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
5.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

6.		<i>Include</i>	Menspesifikasikan bahwa use case sumber secara eksplisit.
7.		<i>Extend</i>	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
8.		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi
9.		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang

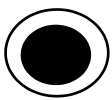
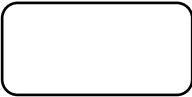
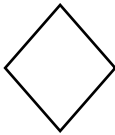
			terukur bagi suatu aktor.
10.		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).

2. Activity Diagram

Activity Diagram merupakan rancangan aliran aktivitas atau aliran kerja dalam sebuah sistem yang akan dijalankan. *Activity Diagram* juga digunakan untuk mendefinisikan atau mengelompokkan aluran tampilan dari sistem tersebut. *Activity Diagram* memiliki komponen dengan bentuk tertentu yang dihubungkan dengan tanda panah. Panah tersebut mengarah ke-urutan aktivitas yang terjadi dari awal hingga akhir.

Tabel II.4
Simbol *activity diagram*

NO	SIMBOL	NAMA	KETERANGAN
1.		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
2.		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.

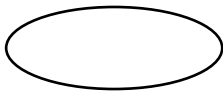
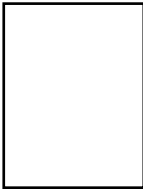
3.		<i>Actifty Final Node</i>	Bagaimana objek dibentuk dan diakhiri.
4.		<i>Actifty</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
5.		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.
6.		<i>Decision</i>	Untuk Menggambarkan test kondisi untuk memastikan bahwa <i>control flow</i> atau <i>object flow</i> mengalir lebih ke satu jalur, jumlah jalur sesuai yang diinginkan.

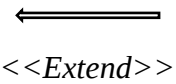
3. Use Case Diagram

Use Case Diagram yaitu salah satu jenis diagram pada UML yang menggambarkan interaksi antara sistem dan aktor, *use case diagram* juga dapat mendeskripsikan tipe interaksi antara si pemakai sistem dengan sistemnya.

Tabel II.5
Simbol *use case diagram*

NO	SIMBOL	NAMA	KETERANGAN
1.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .

2.		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
3.		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
5.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
6.	 <<include>>	<i>Include</i>	

			Menspesifikasikan bahwa use case sumber secara eksplisit.
7.		<i>Extend</i>	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
8.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>)

Sumber : Mila Wati Hartono (2016)

4. *Classs Diagram*

Diagram kelas atau Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak atau programmer dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas.

Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut: (A.S & Shalahuddin, 2018)

a. Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

b. Kelas yang menangani tampilan sistem (*view*)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

c. Kelas yang diambil dari pendefinisian use case (*controller*)

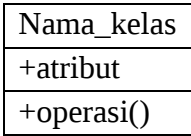
Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

d. Kelas yang diambil dari pendefinisian data (*model*)

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data. Semua tabel yang dibuat di basis data dapat dijadikan kelas, namun untuk tabel dari hasil relasi atau atribut multi value pada ERD dapat dijadikan kelas tersendiri dapat juga tidak asalkan pengaksesannya dapat dipertanggungjawabkan atau tetap ada di dalam perancangan kelas.

Berikut adalah simbol-simbol yang ada pada diagram kelas: (A.S & Shalahuddin, 2018).

Tabel II.6
Simbol *class diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem.
Antarmuka / <i>interface</i>  <i>nama_interface</i>	Sama dengan konsep interface dalam pemrograman berorientasi objek.
Asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semuabagian (whole-part).

5. *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki

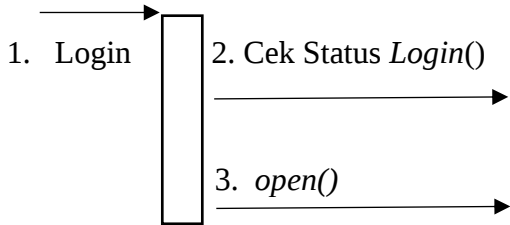
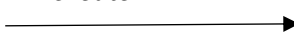
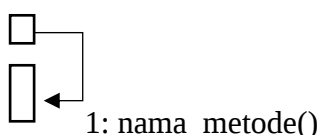
nama_*interface* kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

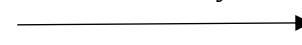
Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Berikut adalah simbol-simbol yang ada pada *diagram sequence*

Tabel II.7
Simbol *sequence diagram*

Simbol	Deskripsi
Aktor  Nama Aktor Atau Nama_Aktor Tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
Garis hidup / <i>lifeline</i> 	Menyatakan kehidupan suatu objek.
Objek Nama objek : nama kelas	Menyatakan objek yang berinteraksi pesan.

Waktu Aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya:  Maka cek Status <i>Login()</i> dan <i>open()</i> dilakukan di dalam metode <i>login()</i> Aktor tidak memiliki waktu aktif.
Pesan tipe <i>create</i> <<create>> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe <i>call</i> 1: nama_metode() 	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri,  arah panah mengarah pada objek yang memiliki operasi / metode maka operasi / metode yang dipanggil harus ada pada diagram kalian sesuai dengan kelas objek yang berinteraksi.
Pesan tipe <i>send</i> 1: masukan 	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
Pesan tipe <i>return</i> 1: Keluaran	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.

	
Pesan tipe <i>destroy</i> <<<i>Destroy</i>>> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.

Sumber : (A.S & Shalahuddin, 2018)

C. Pengujian

Dalam testing dan implementasi sistem dikenal 2 metode pengujian yang populer, yaitu Pengujian *Black-Box Testing* dan Pengujian *White-Box Testing*.

1. *Black-Box Testing*

Menurut Rosa dan Salahuddin (2015:275) “*Black-box* testing yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program”.

2. *White-box testing*

Menurut Pressman (2015:588) “Pengujian white box (Pengujian Kotak Putih), terkadang disebut juga pengujian kotak kaca (glass box testing), merupakan sebuah filosofi perencanaan test case (uji kasus) yang menggunakan struktur control yang dijelaskan sebagai bagian dari perancangan perangkat komponen untuk menghasilkan test case”.