

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Setiap organisasi atau perusahaan memiliki sistem yang menangani serangkaian aktifitas di dalam suatu sistem agar mencapai tujuan yang optimal. Dalam Laporan Tugas Akhir ini penulis membuat rancangan sistem dengan menggunakan Web.

Konsep dasar sistem ini menekankan sekumpulan elemen yang terdapat dalam suatu organisasi, perusahaan, aspek dan sudut pandang yang berbeda sesuai dengan keterangan fungsi dalam hal-hal yang berkaitan dengan sistem yang memiliki ciri dan karakteristik tertentu. Adapun uraian dari konsep dasar sistem akan dijelaskan dibawah ini.

2.1.1 Sistem

Secara umum suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen yang terorganisir, saling beinteraksi, saling ketergantungan satu sama lain dan terpadu untuk mencapai tujuan tertentu.

Menurut Hutahaeen (2015:2) mengemukakan bahwa “sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu.

Dari kutipan diatas, dapat disimpulkan bahwa sistem merupakan serangkaian elemen baik secara fisik maupun non fisik yang berkaitan dengan jaringan kerja yang saling berinterajsu, berhubungan, atau ketergantungan antara satu dengan yang lain

bertujuan untuk menangani dan menyelesaikan serangkaian kegiatan yang terjadi secara rutin dan berulang-ulang.

2.1.2 Karakteristik Sistem

Pembuat sistem harus memahami ciri-ciri atau karakteristik yang terdapat pada sekumpulan elemen yang ada, sebagai dasar pertimbangan dalam pembuatan sistem. Adapun karakteristik dari sistem yang dimaksud (Hutahaean, 2015:3) yaitu:

1. Komponen

Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batasan sistem (*boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

3. Lingkungan luar sistem (*environment*)

Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung sistem (*interface*)

Media yang menghubungkan sistem dengan subsistem yang lain disebut penghubung sistem atau interface. Penghubung ini memungkinkan

sumberdaya mengalir dari satu subsistem ke subsistem lain. Bentuk keluaran dari satu subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut. Dengan demikian, dapat terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan sistem (*input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*).

6. Keluaran sistem (*output*)

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini dapat menjadi masukan bagi subsistem yang lain seperti sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang menjadi input bagi subsistem lain.

7. Pengolah sistem

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

8. Sasaran sistem

Suatu sistem mempunyai tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

2.1.3 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang. Adapun klasifikasi terhadap sistem (Hutahaean, 2015:6) diuraikan sebagai berikut:

1. Sistem Abstrak dan Sistem Fisik

Sistem abstrak merupakan sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem telogi. Sedangkan sistem fisik diartikan sebagai sistem yang nampak secara fisik sehingga setiap makhluk dapat melihatnya, misalnya sistem komputer.

2. Sistem Alamiah dan Sistem Buatan Manusia

Sistem alamiah merupakan sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem tata surya, sistem galaksi, sistem reproduksi dan lain-lain. Sedangkan sistem buatan manusia merupakan sistem yang dirancang oleh manusia. Sistem buatan yang melibatkan interaksi manusia, misalnya sistem akuntansi, sistem informasi, dan lain-lain.

3. Sistem Deterministik dan Sistem Probabilistik

Sistem deterministik merupakan sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan, misalnya sistem komputer, adalah contoh sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang dijalankan. Sedangkan sistem probabilistik merupakan sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas, misalnya sistem manusia.

4. Sistem Terbuka dan Sistem Tertutup

Sistem terbuka merupakan sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Lebih spesifik dikenal juga yang disebut dengan sistem terotomasi, yang merupakan bagian dari sistem buatan manusia dan berinteraksi dengan kontrol oleh satu atau lebih komputer sebagai bagian dari sistem yang

digunakan dalam masyarakat modern. Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainnya, misalnya sistem kebudayaan manusia. Sedangkan sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya campur tangan dari pihak luar. Secara teoritis sistem tersebut ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup).

2.1.4 Sistem Informasi

Sistem memiliki tujuan untuk menghasilkan informasi yang nantinya informasi tersebut akan digunakan sebagai dasar pertimbangan untuk pencapaian di dalam suatu organisasi atau perusahaan.

Menurut Hutahaean (2015:13) mengemukakan bahwa: Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan. Dari kutipan di atas disimpulkan bahwa sistem informasi merupakan kompilasi yang terdiri dari manusia, teknologi informasi, dan prosedur kerja yang saling memiliki kaitan, hubungan atau interaksi untuk melakukan pengolahan data menjadi informasi dalam suatu organisasi sebagai dasar dalam pengambilan keputusan untuk menentukan langkah di masa mendatang.

Komponen-komponen yang membangun sistem informasi ini dikenal dengan istilah blok bangunan (*building block*). Adapun uraian dari blok bangunan (Hutahaean, 2015:13) itu terdiri dari:

1. Blok masukan (*input block*)

Blok masukan merupakan blok yang bertugas dalam input data agar masuk ke dalam sistem informasi. Blok masukan bertugas dalam merekam data yang akan dimasukkan, biasanya berupa dokumen-dokumen dasar.

2. Blok model (*model block*)

Blok model terbentuk dari kombinasi prosedur, logika dan model matematik yang memproses data input dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Blok keluaran (*output block*)

Sistem informasi menghasilkan keluaran (*output*) yaitu informasi yang berkualitas dan berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok teknologi (*technology block*)

Teknologi digunakan merupakan kotak alat dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran berupa informasi dan membantu pengendalian dari sistem secara menyeluruh. Blok teknologi perangkat lunak (*software*) dan perangkat keras (*hardware*) yang dioperasikan oleh teknisi (*brainware*).

5. Blok basis data (*database block*)

Basis data (*database*) merupakan media untuk menyimpan data yang saling berhubungan satu sama lainnya, tersimpan di perangkat keras komputer dan dapat dipergunakan kembali, diperlukan perangkat lunak untuk memanipulasinya.

6. Blok kendali (*control block*)

Sistem informasi memiliki kontrol kendali untuk menanggulangi gangguan gangguan terhadap sistem apabila terlanjur terjadi kesalahan maka dapat langsung diantisipasi atau diatasi.

2.1.5 Pengertian Sistem Berorientasi Objek (OOP)

Menurut (Wibowo, 2015) Secara garis besar, bahasa pemrograman komputer adalah sebuah alat yang dipakai oleh para programmer komputer untuk menciptakan program aplikasi yang digunakan untuk berbagai macam keperluan. Pada tahap awal dikenal beberapa jenis bahasa pemrograman, bahasa ini berbasis teks dan berorientasi linear contohnya : Bahasa BASIC, Bahasa Clipper, Bahasa Pascal, Bahasa cobol. Pemrograman berorientasi objek atau object-oriented programming merupakan suatu pendekatan pemrograman yang menggunakan object dan class. Saat ini konsep OOP sudah semakin berkembang. Hampir semua programmer maupun pengembang aplikasi menerapkan konsep OOP. OOP bukanlah sekedar cara penulisan sintaks program yang berbeda, namun lebih dari itu. OOP merupakan cara pandang dalam menganalisa system dan permasalahan pemrograman. Dalam OOP, setiap bagian dari program adalah object. Sebuah object mewakili suatu bagian program yang akan diselesaikan.

2.1.6 Website

Untuk mengakses *internet* diperlukan suatu program aplikasi *Web Browser*. Program aplikasi ini mempunyai kemampuan untuk menampilkan suatu *Web Page* yang ditulis dalam format HTML. Dua contoh program aplikasi angkatan pertama adalah *Lynx* dan *Mosaic*. *Lynx* adalah *Web Browser* yang berbasis teks.

Kekurangannya adalah tidak bisa menampilkan suatu *Web Page* yang ditulis dalam format HTML. Mosaic adalah browser web mempopulerkan dikreditkan dengan World Wide Web. Itu juga sebelumnya klien untuk protokol seperti FTP, Usenet, dan Gopher. Yang bersih, mudah dipahami user interface, keandalan, Windows pelabuhan dan instalasi sederhana semua berkontribusi untuk membuat aplikasi yang membuka Web untuk masyarakat umum.

Internet berasal dari kata *Interconnection Networking* yang berarti kata jaringan yang saling berhubungan. Didalam mengatur integrasi dan komunikasi jaringan internet dibutuhkan sebuah *protocol* yang disebut TCP/IP (*Transmission Control Protocol/Internet Protocol*).

Secara umum *internet* dapat diartikan sebagai pertukaran informasi dan komunikasi. Semua informasi bisa didapatkan dengan mudah dan bebas di *internet* tanpa ada batasan.

2.1.7 Bahasa Pemrograman

Untuk mengembangkan proses perancangan pembuatan website penulis menjelaskan beberapa bahasa pemrograman yang dipakai jika membuat website. Diantaranya adalah:

1. HTML (*HyperText Markup Language*)

HTML akronim dari *Hypertext Markup Language*. Menurut (Destiana & Hadidah, 2016) “HTML (*Hypertext Markup Language*) merupakan sebuah kombinasi teks dan informasi yang dipakai untuk membuat sebuah halaman situs. HTML (*Hypertext Markup Language*) merupakan bahasa (kode) yang digunakan untuk membuat halaman web”.

Mengapa disebut *hypertext*, karena pemakai dapat membaca tampilan *website* tanpa harus baris demi baris, dapat meloncat dari satu baris ke baris yang lain. Disebut *markup language*, karena susunan instruksinya banyak menggunakan tanda-tanda yang disebut *tag*.

Halaman *web* yang biasa anda lihat ketika *browsing* di internet selalu ditulis menggunakan bahasa HTML. Walaupun halaman *web* tersebut dibuat dengan menggunakan bahasa seperti PHP, JSP, ASPX, hasil akhirnya yang ditampilkan oleh browser tetap adalah HTML murni.

2. PHP (*Personal Home Page*)

PHP singkatan dari *Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang berhasil *open source*. Menurut (Destiana & Hadidah, 2016) “PHP merupakan bahasa berbentuk *script* yang ditempatkan di dalam *server* baru kemudian di proses”. Adapun menurut (Imaniawan & Elsa, 2017) “PHP merupakan bahasa standar yang digunakan dalam dunia *website*, PHP adalah bahasa program yang berbentuk skrip yang diletakkan di dalam *server web*”.

Kemudian hasil pemrosesan dikirimkan kepada web browser klien. Bahasa pemrograman ini dirancang khusus untuk membentuk web dinamis. Artinya, pemrograman PHP dapat membentuk suatu tampilan berdasarkan permintaan terkini.

Secara khusus, PHP dirancang untuk membentuk aplikasi web dinamis. Artinya, ia dapat membentuk satu tampilan berdasarkan permintaan terkini. Misalnya, anda bisa menampilkan isi database ke halaman web. Pada prinsipnya PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP (*Active Server Page*), *Cold Fusion*, ataupun *Perl*. Namun, perlu diketahui bahwa PHP

sebenarnya bisa dipakai secara *common line*. Artinya, skrip PHP dapat dijalankan tanpa melibatkan *web server* maupun *browser*.

3. CSS (*Cascading Style Sheet*)

Menurut (Imaniawan & Elsa, 2017) “*Cascading Style Sheet* memiliki arti gaya menata halaman bertingkat, yang berarti setiap satu elemen yang telah diformat dan memiliki anak yang telah diformat, maka anak dari elemen tersebut secara otomatis mengikuti format elemen induknya”.

CSS mempunyai 2 bagian utama yaitu *selectors* dan deklarasi. Yang dimaksud *selectors* biasanya element HTML yang ingin diubah, sedangkan deklarasi biasanya terdiri dari properti dan nilai. Properti sendiri adalah atribut *style* yang di ingin diubah, dan setiap properti memiliki nilai.

2.1.8 Basis Data

Dalam pembuatan suatu aplikasi, para pembuat aplikasi atau programmer menggunakan basis data yang digunakan untuk pengolahan data atau penataan file-file yang ada dan digunakan kembali sesuai dengan kebutuhan aplikasi tersebut.

1. Pengertian Basis Data

Basis data berguna sebagai tempat pengolahan informasi yang sangat penting dalam upaya menciptakan suatu aplikasi yang terintegrasi. Menurut Lubis (2016:2) mengemukakan bahwa “basis data merupakan gabungan file data yang dibentuk dengan hubungan/relasi yang logis dan dapat diungkapkan dengan catatan serta bersifat independen”. Sedangkan, menurut Sukamto dan Shalahuddin (2015:43), “sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”. Dapat ditarik kesimpulan bahwa basis data merupakan kumpulan

data yang diolah menjadi informasi dan dapat digunakan kembali jika dibutuhkan dalam suatu organisasi atau perusahaan.

2. Aplikasi Basis Data

Aplikasi basis data sering digunakan oleh para pembuat aplikasi sebagai media pengolahan basis data. Aplikasi basis data yang digunakan penulis dalam perancangan basis data yaitu MySQL. :

a. MySQL

Menurut (Destiana & Hadidah, 2016) “MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional RDBMS (*Relational Database Management System*) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*)”.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah.

Sebagai *software* DBMS, MySQL memiliki sejumlah fitur seperti yang di jelaskan di bawah ini:

1) *Multiplatform*

MySQL tersedia pada beberapa *platform* (*Windows, Linux, Unix* dan Lain-lain).

2) Andal cepat dan mudah digunakan

MySQL tergolong sebagai *database server* (*server* yang melayani terhadap permintaan *database*) yang dapat menangani *database* yang

besar dengan kecepatan tinggi, mendukung banyak sekali fungsi untuk mengakses *database*, dan sekaligus mudah untuk digunakan. Berbagai *tool* pendukung juga tersedia (walupun dibuat oleh pihak lain). Perlu diketahui MySQL dapat menangani sebuah tabel yang berukuran dalam *terabyte* (1 *terabyte* = 1024 *Gigabyte*). Namun, ukuran sesungguhnya bergantung kepada batasan sistem operasi. Sebagai contoh : pada sistem Solaris 9/10, batasan ukuran file sebesar 16 *terabyte*.

3) Jaminan keamanan akses

MySQL mendukung keamanan *database* dengan berbagai kriteria pengaksesan. Sebagai gambaran, dimungkinkan untuk mengatur *user* tertentu agar bisa mengakses data yang bersifat rahasia, sedangkan *user* lain tidak boleh.

4) Dukungan MySQL

Seperti tersirat dalam namanya, MySQL mendukung perintah SQL (*Structured Query Language*). Sebagaimana diketahui, SQL merupakan standar dalam pengaksesan *database relasional*. Pengetahuan akan SQL akan memudahkan siapapun untuk menggunakan MySQL.

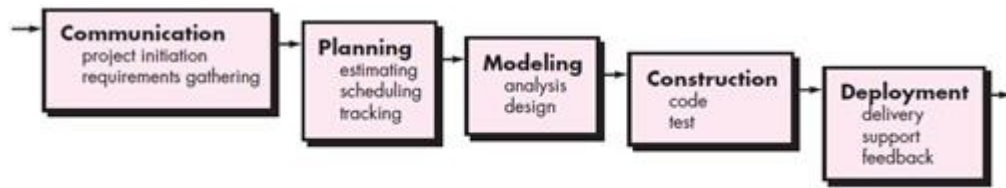
b. PhpMyAdmin

Menurut (Saryoko, 2015) “PhpMyAdmin merupakan aplikasi berbasis web yang dikembangkan menggunakan bahasa pemrograman PHP . Melalui PhpMyAdmin, *user* dapat melakukan perintah *query* tanpa harus mengetikkan seperti pada MS DOS. Perintah tersebut misalnya administrasi user dan *privileges*, *export* dan *import database*, manajemen *database*, manajemen *table* dan struktur *table*. PhpMyAdmin sangat *user friendly*, sehingga mudah untuk digunakan walaupun oleh pengguna baru”.

Setiap RDBMS (*Relation Database Management System*) seperti *Oracle*, *SQL Server*, *MySQL* dan lain-lain, pasti memiliki *tool* yang dapat digunakan untuk mempermudah pengoprasian *database*. *Oracle* memiliki *TOAD*. *SQL Server* memiliki *Enterprise Manager* dan *SQL Query Analyzer*. Sedangkan *MySQL* memiliki *tool* atau aplikasi yang disebut *PhpMyAdmin*.

2.1.9 Model Pengembangan Perangkat Lunak

Disiplin ilmu yang penulis pilih sebagai dasar untuk menentukan pembuatan aplikasi berdasarkan model pengembangan perangkat lunak yaitu model air terjun (*waterfall*). Menurut Pressman (2015:42), model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun software. Nama model ini sebenarnya adalah “*Linear Sequential Model*”. Model ini sering disebut juga dengan “*classic life cycle*” atau metode *waterfall*. Model ini termasuk ke dalam model generic pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam *Software Engineering (SE)*. Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Fase-fase dalam *Waterfall Model* menurut referensi Pressman :



Gambar II.1 Waterfall Pressman (Pressman, 2015:42)

1. *Communication* (Project Initiation & Requirements Gathering)

Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan customer demi memahami dan mencapai tujuan yang ingin dicapai. Hasil dari komunikasi tersebut adalah inisialisasi proyek, seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi software. Pengumpulan data-data tambahan bisa juga diambil dari jurnal, artikel, dan internet.

2. *Planning* (Estimating, Scheduling, Tracking)

Tahap berikutnya adalah tahapan perencanaan yang menjelaskan tentang estimasi tugas-tugas teknis yang akan dilakukan, resiko-resiko yang dapat terjadi, sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan, dan tracking proses pengerjaan sistem.

3. *Modeling* (Analysis & Design)

Tahapan ini adalah tahap perancangan dan permodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur software, tampilan interface, dan algoritma program. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.

4. *Construction* (Code & Test)

Tahapan Construction ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk/bahasa yang dapat dibaca oleh mesin. Setelah pengkodean selesai, dilakukan pengujian terhadap sistem dan juga kode yang sudah dibuat. Tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki.

5. *Deployment* (Delivery, Support, Feedback)

Tahapan Deployment merupakan tahapan implementasi software ke customer, pemeliharaan software secara berkala, perbaikan software, evaluasi software, dan pengembangan software berdasarkan umpan balik yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya. (Pressman, 2015:17)

2.2. **Teori Pendukung**

Berikut ini beberapa teori pendukung yang digunakan dalam membangun website, diantaranya adalah:

2.2.1 *Entity Relationship Diagram (ERD)*

1. **Pengertian ERD**

Entity Relationship Diagram (ERD) merupakan sebuah cara untuk menggambarkan sebuah basis data yang menggunakan simbol-simbol beserta hubungan antara simbol-simbol tersebut.

Menurut Sukamto dan Shalahuddin (2015:53) “ERD adalah bentuk paling awal dalam melakukan perancangan basis data relasional. Jika menggunakan OODMBS maka perancangan ERD tidak perlu dilakukan”

2. Logical Record Structure.

Menurut (Imaniawan & Elsa, 2017) “LRS merupakan transformasi dari penggambaran ERD dalam bentuk yang lebih jelas dan mudah untuk dipahami” Dua aturan tersebut yaitu:

- a. Setiap *entity* akan diubah ke bentuk sebuah kotak dengan nama *entity* berada diluar kotak dan atribut berada di dalam kotak.
- b. Sebuah relasi kadang disatukan dalam sebuah kotak bernama *entity*, kadang dipisah dalam sebuah kotak tersendiri.
- c. Aturan pokok di tersebut akan sangat dipengaruhi oleh elemen yang menjadi titik perhatian utama pada langkah transformasi yaitu *cardinality*/kardinalitas.

3. Elemen-Elemen Diagram

- a. Entitas (*entity*)

Entity menunjukkan obyek-obyek dasar yang terkait didalam sistem obyek dasar dapat berupa orang, benda atau hal lain yang keterangannya perlu disimpan dalam basis data (Sukamto dan Shalahuddin, 2015:50).

- b. Atribut (*Attribute*)

Atribut sering juga disebut sebagai properti, merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan sebagai basis data (Sukamto dan Shalahuddin, 2015:50)

c. Atribut kunci primer

Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record yang diinginkan; biasanya berupa id (Sukamto dan Shalahuddin, 2015:50).

d. Atribut multinilai (*multivalue*)

Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu (Sukamto dan Shalahuddin, 2015:51).

e. Relasi (*relation*)

Relasi atau hubungan adalah kejadian atau transaksi yang terjadi diantara dua entity yang keterangannya (Sukamto dan Shalahuddin, 2015:51).

f. Asosiasi (*assocation*)

Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki multiplicity (Sukamto dan Shalahuddin, 2015:51).

Multiplicity yang dimaksud mengalami perubahan yang mengikuti aturan-aturan sebagai berikut :

- 1) Setiap entitas akan diubah kebentuk kotak.
- 2) Sebuah atribut relasi disatukan dalam sebuah kotak bersama entitas jika hubungan yang terjadi pada diagram-ER 1:M (relasi bersatu dengan cardinality M) atau tingkat hubungan 1:1 (relasi bersatu dengan cardinality yang paling membutuhkan referensi).

3) Sebuah relasi dipisah dalam sebuah kotak tersendiri (menjadi entitas baru)

jika

tingkat hubungannya M:M (many to many) dan memiliki foreign key sebagai primary key yang diambil dari kedua entitas yang sebelumnya saling berhubungan.

2.2.2 *Unified Modeling Language (UML)*

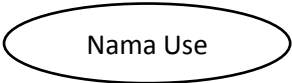
Menurut Rosa dan Shalahuddin (2014:133), berpendapat bahwa UML (*Unified Modeling Language*) adalah “Salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisa & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”. Sedangkan Mulyani (2016:48) mengatakan UML (Unified Modeling Language) adalah “Sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem”. Dari beberapa penjelasan teori tersebut dapat disimpulkan bahwa UML (Unified Modeling Language) adalah bahasa yang sering digunakan untuk membangun sebuah sistem perangkat lunak dengan melakukan penganalisaan desain dan spesifikasi dalam pemrograman berorientasi objek. UML (Unified Modeling Language) memiliki diagram-diagram yang digunakan dalam pembuatan aplikasi berorientasi objek, diantaranya (Rosa dan Shalahuddin, 2014:155):

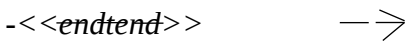
1. *Use Case Diagram*

Use Case Diagram merupakan pemodelan untuk melakukan (*behavior*) system informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol-simbol yang ada pada diagram *use case* (Rosa dan Shalahuddin, 2014:156):

Tabel II.1

Use Case Diagram

Symbol	Deskripsi
<i>Use Case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama Use Case
Aktor/ <i>actor</i>  <i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informaasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang: biasanya dinyatakan menggunakan kata benda diawal <i>frase</i> nama actor
Asosiasi / <i>association</i> 	Komunikasi antara aktor dan use case yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor

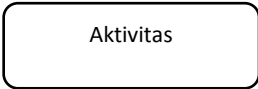
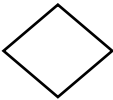
Ekstensi/ <i>extend</i> 	Relasi <i>use case</i> tambahan kesebuah <i>use case</i> dinamakan <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>intherince</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.
Generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari laainnya.

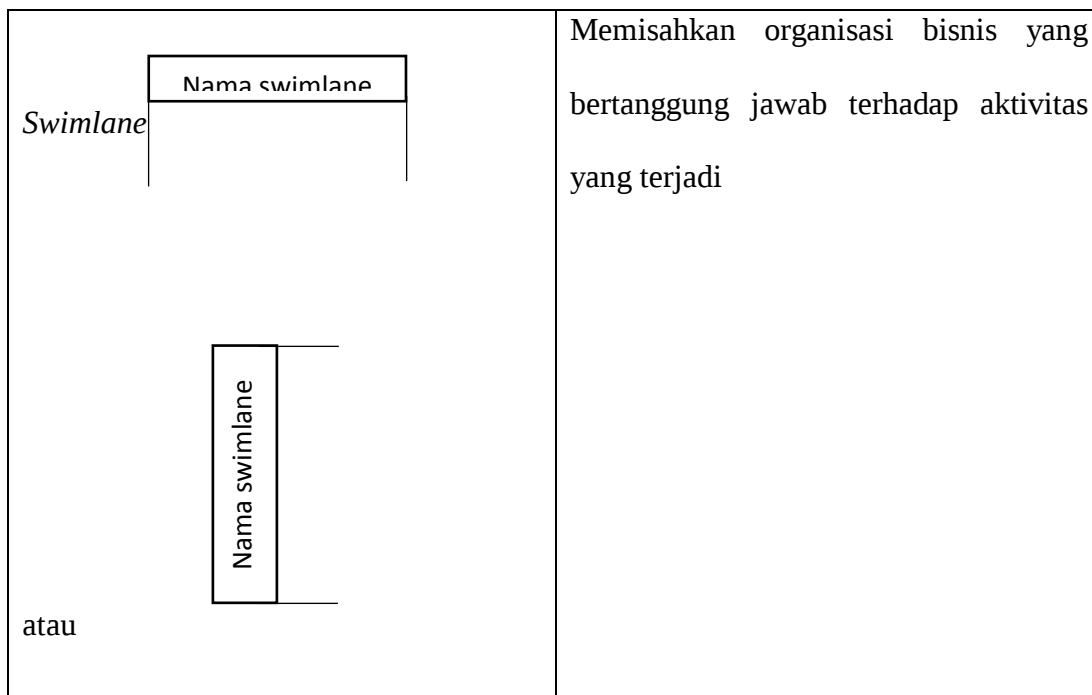
Sumber : Rosa dan Shalahudin (2014:156)

2. Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Perludiperhatikan bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas (Rosa dan Shalahuddin, 2014:162):

Tabel II.2
Activity Diagram

Simbol/	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir



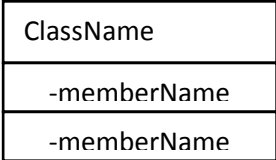
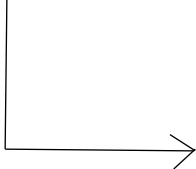
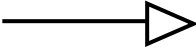
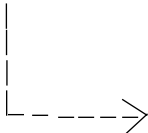
Sumber : Rosa dan Shalahuddin (2014:162)

3. Class Diagram

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.

Berikut adalah simbol-simbol yang ada pada diagram Kelas (Rosa dan Shalahuddin, 2014:141) :

Tabel II.3
Class Diagram

Simbol	Deskripsi
Kelas 	Kelas pada struktur system
Antarmuka/ <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi/ <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
Asosiasi berarah/ <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umumkhusus)
Kebergantungan/ <i>dependensi</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

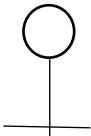
Sumber : Rosa dan Shalahuddin (2014:162)

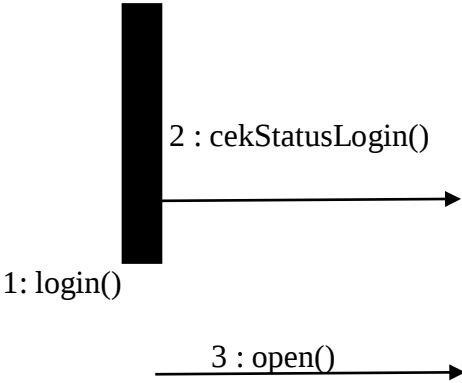
4. *Sequence Diagram*

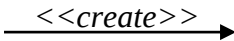
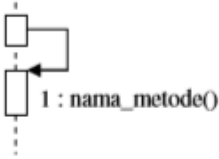
Rosa dan M. Shalahudin (2014:165), diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dengan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki 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 dalam diagram sekuen sehingga semakin banyak use case yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Tabel II.4

Tabel Sequence Diagram

Symbol	Deskripsi
Aktor/ <i>actor</i>   <i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar 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 diawal <i>frase</i> nama actor
Garis hidup / <i>lifeline</i> 	Menyatakan kehidupan suatu objek
Objek <u>Nama objek : nama kelas</u>	Menyatakan objek yang berinteraksi Pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah → sebuah tahapan yang dilakukan di dalamnya, misalnya 

	Maka cekStatusLogin() dan open() dilakukan didalam metode login(). 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,  1 : nama_metode() Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi

Pesan tipe <i>send</i> 1 : masukkan 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/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> <<destroy>> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>destroy</i>

Sumber : Rosa dan Shalahuddin (2014:162)