

BAB II

LANDASAN TEORI

2.1 Teori Umum

2.1.1 Pengertian Perancangan

Perancangan pada dasarnya telah di deskripsikan sebagai proses banyak langkah dimana representasi – representasi data dan struktur program, karakteristik – karakteristik antar muka, dan rincian prosedural diikhtisarkan dari hal – hal yang berkaitan dengan kebutuhan – kebutuhan informasi.

Menurut Jogiyanto dalam Anggraeni (2016:9) “Perancangan adalah suatu jaringan kerja yang saling berhubungan untuk menentukan bagaimana suatu sistem menyelesaikan apa yang mesti diselesaikan”.

Menurut Soetam Rizky (2011:140) “Perancangan adalah sebuah proses untuk mendefinisikan sesuatu yang akan dikerjakan dengan menggunakan teknik yang bervariasi serta di dalamnya melibatkan deskripsi mengenai arsitektur serta detail mengenai komponen dan juga keterbatasan yang akan dialami dalam proses pengerjaan nya”.

Kristanto (2013:61) “Perancangan sistem adalah suatu fase dimana diperlukan suatu keahlian perancangan untuk elemen-elemen komputer yang akan menggunakan sistem, yaitu pemilihan peralatan dan program komputer untuk sistem yang baru”.

Kegiatan yang dilakukan dalam tahap perancangan ini meliputi perancangan *output*, *input*, dan *file*.

a. Perancangan Keluaran (*Output*)

Perancangan keluaran bertujuan menentukan keluaran-keluaran yang akan digunakan oleh sistem. Keluaran tersebut berupa tampilan-tampilan layar, dan juga format dan frekuensi laporan yang diperlukan.

b. Perancangan Masukan (*Input*)

Perancangan masukan bertujuan menentukan data-data masukan, yang akan digunakan untuk mengoperasikan sistem. Data-data masukan tersebut dapat berupa formulir-formulir, faktur, dan lain-lain yang berfungsi memberikan data masukan bagi pemrosesan sistem. Pada tahap ini perlu juga ditentukan format data masukan agar sesuai dengan kebutuhan sistem.

2.1.2 Pengertian Sistem

Menurut Susanto (2013:45) “Sistem adalah kumpulan atau *group* dari sub sistem atau bagian maupun komponen apapun baik fisik ataupun *non* fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu”.

Menurut Jogiyanto dalam Andalia (2015:93) “Sistem adalah kumpulan dari elemen – elemen yang berinteraksi untuk mencapai suatu tujuan tertentu”.

Menurut Rudy Tantra (2012:1) “Sistem dapat diartikan sebagai entitas atau satuan yang terdiri dari dua atau lebih komponen atau subsistem (sistem yang lebih kecil) yang saling terhubung dan terkait untuk mencapai suatu tujuan”.

1. Klasifikasi Sistem

Menurut Sutabri (2012:22) “Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi dalam sistem tersebut”. Oleh karena itu, sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya :

- a. Sistem Abstrak (*Abstract System*) dan Sistem Fisik (*Physical System*) adalah Sistem abstrak merupakan sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Misalnya sistem teologi, yaitu sistem yang berupa pemikiran-pemikiran hubungan antara manusia dengan Tuhan. Sistem fisik merupakan sistem yang ada secara fisik. Misalnya sistem komputer, sistem akuntansi, dan sistem persediaan barang.
- b. Sistem Alamiah (*Natural System*) dan Sistem Buatan Manusia (*Human Made System*) adalah Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Misalnya sistem perputaran bumi. Sistem buatan manusia adalah sistem yang melibatkan interaksi manusia dengan mesin

yang disebut human machine system. Misalnya sistem informasi berbasis komputer.

- c. Sistem Tertentu (*Deterministic System*) dan Sistem Tak Tentu (*Probabilistic System*) adalah Sistem tertentu adalah sistem yang beroperasi dengan tingkah laku yang dapat diprediksi. Sebagai contoh adalah hasil pertandingan sepak bola. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas. Misalnya kematian seseorang.
- d. Sistem Tertutup (*Closed System*) dan Sistem Terbuka (*Open System*) adalah Sistem tertutup adalah sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya campur tangan dari pihak diuarnya. Secara teoritis sistem tertutup ini ada, tetapi pada kenyataannya tidak ada sistem yang benar-benar tertutup. Contohnya adalah sistem adat masyarakat Baduy. Sedangkan sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lainnya. Misalnya sistem musyawarah.

2. Karakteristik Sistem

Menurut Yasin (2012:261) “suatu sistem memiliki karakteristik tertentu, yang menjelaskan bahwa hal tersebut biasa dikatakan sebagai suatu sistem”. Adapun yang termasuk ke dalam karakteristik adalah sebagai berikut:

a. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama membentuk satu kesatuan. 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. Suatu sistem dapat mempunyai sistem yang lebih besar atau sering disebut “*super system*”.

b. 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.

c. Lingkup Luar Sistem (*enviromtent*)

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.

d. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem lain disebut penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya 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 integritas sistem yang membentuk satu kesatuan.

e. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan dan sinyal. Contohnya, di dalam suatu unit sistem komputer, "program" adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan "data" adalah *signal input* untuk diolah menjadi informasi.

f. Keluaran Sistem (*Output*)

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain seperti sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini dapat digunakan sebagai

masuk untuk pengambilan keputusan atau hal-hal lain yang menjadi *input* bagi sub sistem lain.

g. Pengolah Sistem (Proses)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran, contohnya sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

h. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat *deterministic*. 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 Informasi

A. Pengertian Informasi

Informasi dapat berupa data mentah, data tersusun, kapasitas sebuah saluran komunikasi dan lain sebagainya. Informasi merupakan suatu hal yang sangat penting didalam suatu organisasi.

Menurut Lucas dalam Hartono (2013:15) “Informasi sebagai data yang telah di tafsirkan agar memberikan makna tertentu bagi seseorang”.

Menurut Davis dalam Hartono (2013:15) “Informasi adalah data yang telah diolah menjadi suatu bentuk yang berguna bagi penerimannya dan memiliki nilai bagi pengambilan keputusan saat ini di masa yang akan datang”.

Berdasarkan beberapa pendapat yang dikemukakan di atas dapat ditarik kesimpulan bahwa informasi adalah fakta yang telah diolah dengan cara tertentu dan mempunyai arti berguna bagi penerimanya atau menggambarkan suatu kejadian nyata yang dapat dipahami dan dapat digunakan untuk pengambilan keputusan sekarang maupun untuk masa depan.

B. Tujuan Informasi

Tujuan dari sistem informasi adalah menghasilkan informasi (*Information*) dari bentuk data yang diolah menjadi bentuk yang berguna bagi para pemakainya. (Jogiyanto 2010:13). Tujuan sistem informasi terdiri dari :

1. Kegunaan (*Usefulness*) adalah Sistem harus menghasilkan informasi yang akurat, tepat waktu, dan relevan untuk pengambilan keputusan manajemen dan personil operasi di dalam organisasi.
2. Ekonomi (*Economic*) adalah Semua bagian komponen sistem termasuk laporan-laporan, pengendalian-pengendalian,

mesin-mesin harus menyumbang suatu nilai manfaat setidaknya tidaknya sebesar biaya yang dibutuhkan.

3. Keandalan (*Realibility*) adalah Keluaran sistem harus mempunyai tingkatan ketelitian yang tinggi dan sistem itu sendiri harus mampu beroperasi secara efektif bahkan pada waktu komponen manusia tidak hadir atau saat komponen mesin tidak beroperasi secara temporer.
4. Pelayanan Langganan (*Customer Service*) adalah Sistem harus memberikan pelayanan dengan baik atau ramah kepada para pelanggan. Sehingga sistem tersebut dapat diminati oleh para pelanggannya.
5. Kesederhanaan (*Simplicity*) adalah Sistem harus cukup sederhana sehingga terstruktur dan operasinya dapat dengan mudah dimengerti dan prosedurnya mudah diikuti.
6. Fleksibilitas (*Fleksibility*) adalah Sistem harus cukup fleksibel untuk menangani perubahan-perubahan yang terjadi, kepentingannya cukup beralasan dalam kondisi dimana sistem beroperasi atau dalam kebutuhan yang diwajibkan oleh organisasi.

C. Nilai Informasi

Parameter untuk mengukur nilai sebuah informasi (*value of information*) ditentukan dari dua hal pokok yaitu manfaat (*benefit*) dan biaya (*cost*). Namun, dalam kenyataannya informasi yang biaya untuk mendapatkannya tinggi belum tentu memiliki manfaat yang tinggi pula.

Menurut Sutarman (2013:14) nilai dari informasi ditentukan oleh empat hal yaitu :

1. Memperoleh pemahaman dan manfaat.
2. Mendapatkan pengalaman.
3. Mengakumulasi proses pembelajaran sehingga dapat diduplikasikan dalam pemecahan masalah atau proses bisnis tertentu.
4. Mengekstrak implikasi kritis dan merefleksikan pengalaman masa lampau yang menyediakan pengetahuan yang terorganisasi dengan nilai yang tinggi. Nilai ini bisa menghindari seseorang manajer dari membuat kesalahan yang sama dilakukan oleh manajer lain.

Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya untuk mendapatkannya dan sebagian besar informasi tidak dapat tepat ditaksir keuntungannya dengan satuan nilai uang, tetapi dapat ditaksir nilai efektivitasnya. (Mulyanto, 2012 : 247).

D. Kualitas Informasi

Menurut Delone Mc Lean dalam Setiawan (2016:2), indikator-indikator yang mendukung kualitas informasi adalah sebagai berikut :

1. Kelengkapan (*Completeness*)

Suatu informasi yang dihasilkan oleh sistem informasi dapat di katakan berkualitas jika informasi yang di hasilkannya lengkap. Informasi yang lengkap ini sangat dibutuhkan oleh pengguna dalam pengambilan keputusan. Informasi yang lengkap ini mencakup seluruh informasi yang di butuhkan pengguna.

2. Relevansi (*Relevance*)

Kualitas informasi dikatakan relevan apabila informasi tersebut mempunyai manfaat bagi penggunanya. Sesuatu dikatakan relevan jika memiliki hubungan, berkaitan, atau berguna secara langsung.

3. Tepat (*Accurate*)

Sebuah informasi dapat dikatakan akurat jika informasi tersebut tidak bias atau menyesatkan, bebas dari kesalahan-kesalahan dan harus jelas mencerminkan maksudnya. Ketidakakuratan sebuah informasi dapat terjadi karena sumber informasi atau Data mengalami gangguan atau kesengajaan sehingga merusak atau merubah data-data asli

tersebut. Informasi yang dihasilkan oleh sistem informasi harus akurat karena sangat berguna bagi pengguna dalam hal pengambilan keputusan. Informasi yang akurat harus terbebas dari kesalahan-kesalahan. Akurat juga informasi tersebut harus jelas dengan kata lain harus mencerminkan maksud dari informasi yang disediakan oleh sistem informasi.

4. Ketepatan Waktu (*Timeliness*)

Informasi yang tepat waktu sangat diperlukan sehingga informasi yang datang kepada penerima tidak terlambat. Dengan kata lain untuk informasi yang terlambat menjadikan informasi tersebut sudah tidak memiliki nilai lagi, informasi yang tepat waktu menjadi landasan dalam pengambilan keputusan yang cepat, jika informasi tersebut terlambat maka keputusan yang diambilpun menjadi terlambat.

5. Format

Maksudnya agar memudahkan pengguna untuk memahami informasi yang disediakan oleh sistem informasi mencerminkan kualitas informasi yang baik. Jika informasi yang disajikan dalam bentuk yang tepat maka informasi yang dihasilkan dianggap berkualitas, tujuannya untuk memudahkan pengguna.

2.1.4 Sistem Informasi

Sistem informasi adalah suatu sistem yang menyediakan informasi untuk manajemen pengambilan keputusan atau kebijakan dan menjalankan operasional dari kombinasi orang-orang, teknologi informasi dan prosedur-prosedur yang terorganisasi. Atau sistem informasi diartikan sebagai kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi untuk mendukung operasi dan manajemen. Sedangkan dalam arti luas, sistem informasi diartikan sebagai sistem informasi yang sering digunakan menurut kepada interaksi antara orang, proses, algoritmik, data dan teknologi.

Menurut Rudy Tantra (2012:2) “Suatu cara yang terorganisir untuk mengumpulkan masukan dan memproses data, menyimpannya, mengelola, mengontrol dan melaporkannya sehingga dapat mendukung suatu perusahaan atau organisasi untuk mencapai tujuan. Sistem informasi sendiri dapat bersifat formal maupun informal, sistem informasi akuntansi, produksi dan penjualan merupakan contoh dari sistem informasi formal yang memang secara resmi memiliki tanggung jawab untuk menghasilkan informasi yang akurat, sedangkan sistem informasi informal adalah kebalikannya, berasal dari bagian-bagian organisasi yang tidak secara resmi memberikan informasi”.

Menurut Kenneth dalam Jane (2011:15) “Sistem Informasi secara teknis dapat didefinisikan sebagai sekumpulan komponen yang saling berhubungan, mengumpulkan, memproses, menyimpan dan mendistribusikan informasi untuk menunjang pengambilan keputusan dan pengawasan dalam suatu organisasi”.

Informasi merupakan hal yang sangat penting bagi sebuah manajemen, informasi dapat diperoleh dari sebuah sistem informasi. Dalam bentuk kompleks, sistem informasi melibatkan banyak pemakai yang tersebar diberbagai tempat yang berjauhan. Oleh karena itu sistem informasi sangat mendukung terhadap proses produksi, kualitas, manajemen, pengambilan keputusan, dan pemecahan masalah.

2.1.5 Pengertian Aplikasi

Menurut Nazrudin Safaat (2019:9) “Perangkat lunak aplikasi adalah suatu sub kelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna.

Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna, contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut suatu paket atau suite aplikasi (*application suite*), contohnya adalah *Microsoft Office*, dan *Open Office.org*, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi – aplikasi dalam suatu paket biasanya memiliki antar muka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan setiap aplikasi. Sering kali, aplikasi ini memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna.

2.1.6 Pengertian *Event Organizer*

Event atau acara adalah hal atau pokok yang akan dilakukan secara terencana untuk suatu tujuan. Acara ini dilaksanakan atas dasar rencana yang telah disiapkan sebelumnya dan tidak spontan. Setidaknya ada beberapa hal yang perlu diperhatikan di acara manajemen, seperti: perencanaan yang matang, panitia, desain, jadwal acara. Sama seperti organisasi dan serikat siswa pada umumnya.

Event Organizer adalah sebuah organisasi profesional dimana didalamnya terdapat sekumpulan orang-orang yang menyelenggarakan sebuah pertunjukan yang dimana didalamnya terdapat banyak pembagian tugas tiap anggota yang mengacu kepada sebuah tujuan yang sama. (Maganada dan Wijaya 2010:2)

Menurut Noor (2009:51) “*event* adalah peristiwa yang memperingati hal-hal penting dalam seluruh kehidupan manusia, baik secara individu maupun dalam kelompok yang berkaitan dengan adat, budaya, tradisi dan agama, yang diadakan untuk tujuan tertentu dan yang melibatkan lingkungan masyarakat pada waktu tertentu”.

2.2 Teori Khusus

2.2.1 *Flowchart* (Bagan Alir)

A. Pengertian *Flowchart*

Bagan alir (*flowchart*) Menurut Romney & Steinbart (2012:56) bagan alir (*flowchart*) “adalah suatu teknik analitis yang digunakan untuk menggambarkan beberapa aspek pada system informasi ke dalam suatu cara yang jelas, ringkas, dan logis”.

Simbol *flowchart* menurut Romney & Steinbart (2012:57) dibagi ke dalam empat kategori berikut :

1. *Input/output symbols*, merupakan perangkat atau media yang menyediakan input atau mencatat output dari proses operasi.
2. *Processing symbols*, menunjukkan tipe perangkat apa yang digunakan untuk memproses data atau mengindikasikan kapan sebuah proses dilakukan secara manual.
3. *Storage symbols*, menunjukkan perangkat yang digunakan untuk menyimpan data yang tidak sedang digunakan oleh sistem.

4. *Flow and miscellaneous symbols*, mengindikasikan aliran data dan barang, serta mewakili operasi dimana flowchart diawali atau diakhiri, dimana keputusan dibuat, dan kapan memberikan penjelasan tambahan pada *flowchart*.



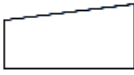
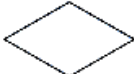
B. Jenis – Jenis *Flowchart*

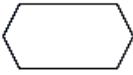

Jenis – jenis *Flowchart* menurut Jogyanto dalam buku Analisis dan Disain Sistem Informasi, jenis-jenis *Flowchart* terdiri dari :

1. Bagan Alir Sistem (*systems flowchart*)

Bagan alir sistem (*systems flowchart*) merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem.

Tabel 2.1 Simbol *Flowchart Systems*

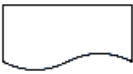


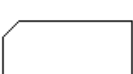
SIMBOL	NAMA	ARTI
	<i>Input / Outout</i>	Merepresentasikan input data atau output data yang di proses atau informasi.
	<i>Penghubung</i>	Keluar atau masuk dari bagian lain flowchart khususnya halaman yang sama.
	<i>Manual Input</i>	Input yang dimasukan secara manual dari keyboard.
	<i>Keputusan</i>	Keputusan dalam program.

	<i>Preparation</i>	Pemberian harga awal.
	<i>Terminal Point</i>	Awal/Akhir flowchart

2. Bagan Alir Dokumen (*document flowcharts*)

Bagan alir dokumen (*document flowcharts*) merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya.





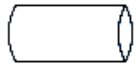

Tabel 2.2 Simbol *Flowchart Document*

SIMBOL	NAMA	ARTI
	<i>Dokumen</i>	I/O dalam format yang dicetak.
	<i>Manual Opration</i>	Operasi manual.
	<i>Predefined Process</i>	Rincian operasi berada di tempat lain
	<i>Punched Card</i>	Input / output yang menggunakan kartu berlubang.

3. Bagan Alir Skematik (*schematic flowcharts*)

Bagan alir skematik (*schematic flowcharts*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem.



Tabel 2.3 Simbol *Flowchart Schematic*

SIMBOL	NAMA	ARTI
	<i>Anak Panah</i>	Merepresentasikan alur kerja.
	<i>Punched Tape</i>	I/O yang menggunakan pita kertas berlubang.
	<i>Communication Link</i>	Transmisi data melalui channel komunikasi, seperti telepon.
	<i>Magnetik Tape</i>	I/O yang menggunakan pita magnetik.
	<i>Magnetik Drum</i>	I/O yang menggunakan drum magnetik.
	<i>Magnetik Disk</i>	I/O yang menggunakan disk magnetik.

4. Bagan Alir Program (*program flowcharts*)

Bagan alir program (*program flowcharts*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program.

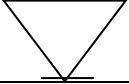

Tabel 2.4 Simbol *Flowchart Program*

SIMBOL	NAMA	ARTI
	<i>Online Storage</i>	I/O yang menggunakan penyimpanan akses langsung.
	<i>Display</i>	Output yang ditampilkan pada terminal.

5. Bagan Alir Proses (*process flowcharts*)

Bagan alir proses (*process flowcharts*) merupakan bagan alir yang banyak digunakan di teknik industri. Bagan alir ini juga berguna bagi analisis sistem untuk menggambarkan proses dalam suatu prosedur.

Tabel 2.5 Simbol *Flowchart Process*

SIMBOL	NAMA	ARTI
	<i>Offline Storage</i>	Penyimpanan yang tidak dapat diakses oleh komputer secara langsung.
	<i>Proses</i>	Mempresentasikan operasi.

2.2.2 Pengertian UML (Unified Modeling Language)

Menurut Shalahuddin (2015:105) “UML (*Unified Modeling Language*) adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

Menurut Mulyani (2016:42) “*Unified Modeling Language* selanjutnya disebut UML adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem”.

Menurut Yasin (2012:267) “*Unified Modeling Language (UML)* adalah notasi yang lengkap untuk membuat visualisasi model suatu

sistem. Sistem berisi informasi dan fungsi, tetapi secara normal digunakan untuk memodelkan sistem komputer”.

Menurut Yasin (2012:268) Tujuan UML diantaranya adalah :

1. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan sistem dan yang dapat saling menukar model dengan mudah dan dimengerti secara umum.
2. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
3. Menyatukan praktek – praktek terbaik yang terdapat dalam pemodelan.

A. *Use Case Diagram*


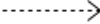





Use case adalah abstraksi dari interaksi antara *sistem* dan *actor*. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara *user* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata *user*. Sedangkan *use case diagram* memfasilitasi komunikasi diantara analis dan pengguna serta antara analisis dan *client*.




Menurut Mulyani (2016:42) “*Use Case Diagram*, yaitu diagram yang digunakan untuk menggambarkan hubungan antara sistem dengan aktor”.

Menurut Yasin (2012:269) “*Use Case diagram* adalah gambar dari beberapa atau seluruh aktor dan *use case* dengan tujuan mengenali interaksi mereka dalam suatu sistem”.

Menurut Hendini (2016:108) Simbol – simbol yang digunakan pada *use case diagram* ditunjukkan pada Tabel 2.6.

Tabel 2.6 Simbol *Use Case Diagram*

GAMBAR	NAMA	KETERANGAN
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<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>).
	<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>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen (sinergi).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

B. Activity Diagram







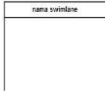
Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

Menurut Mulyani (2016:55) “*Activity diagram* yaitu diagram yang digunakan untuk menggambarkan alur kerja (aktivitas) pada *use case* (proses), logika, proses bisnis dan hubungan antara aktor dengan alur – alur kerja *use case*”.

Menurut Yasin (2012:270) “*Activity diagram* menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya seperti *use case* atau interaksi”.

Menurut Hendini (2016:109) Simbol – simbol yang digunakan pada *activity diagram* ditunjukkan pada Tabel 2.7.

Tabel 2.7 Simbol *Activity Diagram*

GAMBAR	NAMA	KETERANGAN
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
	<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi
	<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
	<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
	<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran
	<i>Decision</i>	Pilihan untuk mengambil keputusan
	<i>Swimlane</i>	Pembagian activity diagram untuk menunjukkan siapa melakukan apa

C. *Sequence Diagram*


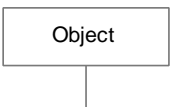

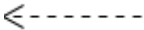

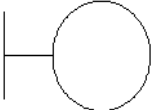
Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

Menurut Yasin (2012:272) “*Sequence diagram* menggambarkan kolaborasi dinamis antara sejumlah objek dan untuk menunjukan rangkaian pesan yang dikirim antar objek juga interaksi antar objek, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem”.

Menurut Hendini (2016:110) “Sequence Diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek”.

Menurut Hendini (2016:110) Simbol – simbol yang digunakan pada *sequence diagram* ditunjukkan pada Tabel 2.8.

Tabel 2.8 Simbol *Sequence Diagram*

GAMBAR	NAMA	KETERANGAN
	<i>Actor</i>	<i>Actor</i> juga dapat berkomunikasi dengan <i>object</i> , maka <i>actor</i> juga dapat diurutkan sebagai kolom.
	<i>Object</i>	<i>Object</i> atau biasa juga disebut partisipan merupakan instance dari sebuah class dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah class (kotak) dengan nama objek didalamnya yang diawali dengan sebuah titik koma.
	<i>Lifeline</i>	<i>Lifeline</i> mengindikasikan keberadaan sebuah object dalam basis waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus vertikal yang ditarik dari sebuah objek.
	<i>Message Return</i>	Menggambarkan pesan/hubungan antar obyek, yang menunjukkan urutan kejadian yang terjadi.
	<i>Activation</i>	<i>Activation</i> dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah <i>lifeline</i> . <i>Activation</i> mengindikasikan sebuah objek yang melakukan suatu aksi.
	<i>Boundary</i>	Boundary terletak di antara sistem dengan dunia sekelilingnya. Semua form, laporan-laporan, antar muka ke perangkat keras seperti printer atau scanner dan antar muka ke sistem lainnya adalah termasuk dalam kategori.



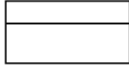

	<i>Control</i>	<i>Control</i> berhubungan dengan fungsionalitas seperti pemanfaatan sumber daya, pemrosesan terdistribusi, atau penanganan kesalahan.
	<i>Entity</i>	<i>Entity</i> digunakan menangani informasi yang mungkin akan disimpan secara permanen. <i>Entity</i> bisa juga merupakan sebuah tabel pada struktur basis data.
	<i>Message</i>	<i>Message</i> , digambarkan dengan anak panah horizontal antara <i>Activation</i> . <i>Message</i> mengindikasikan komunikasi antara <i>object-object</i> .
	<i>Self-Message</i>	<i>Self-message</i> atau panggilan mandiri mengindikasikan komunikasi kembali kedalam sebuah objek itu sendiri.

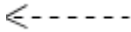
D. Class Diagram

Class Diagram menggambarkan struktur dan deskripsi *class*, *package*, dan objek beserta hubungan satu, antara lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class Diagram* berfungsi untuk menjelaskan tipe dari objek sistem dan hubungannya dengan objek yang lain. Objek adalah nilai tertentu dari setiap *attribute* kelas *entity*.

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan orientasi objek. *Class* menggambarkan keadaan (*attribute/property*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi).

Tabel 2.9 Simbol *Class Diagram*

GAMBAR	NAMA	KETERANGAN
	<i>Composition</i>	Jika sebuah <i>class</i> tidak bisa berdirisendiri dan harus merupakan bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi <i>Composition</i> terhadap <i>class</i> tempat dia bergantung tersebut. Sebuah <i>relationship composition</i> digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/ <i>solid</i> .
	<i>Aggregation</i>	<i>Aggregation</i> mengindikasikan keseluruhan bagian <i>relationship</i> dan biasanya disebut sebagai relasi “mempunyai sebuah” atau “bagian dari”. Sebuah <i>aggregation</i> digambarkan sebagai sebuah garis dengan sebuah jajaran genjang yang tidak berisi/tidak solid.
	<i>Class</i>	<i>Class</i> adalah blok - blok pembangun pada pemrograman berorientasi objek. Sebuah <i>class</i> digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari <i>class</i> . Bagian tengah mendefinisikan <i>property/atribut class</i> . Bagian akhir mendefinisikan <i>method-method</i> dari sebuah <i>class</i> .
<u>1..n</u> Owned by 1	<i>Association</i>	Sebuah asosiasi merupakan sebuah <i>relationship</i> paling umum antara 2 <i>class</i> , dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>class</i> . Garis ini bisa melambangkan tipe-tipe <i>relationship</i> dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah <i>relationship</i> (Contoh: <i>One-to-one</i> , <i>one-to-many</i> , <i>many-to-many</i>).
	<i>Generalization</i>	Sebuah relasi <i>generalization</i> sepadan dengan sebuah relasi <i>inheritance</i> pada konsep berorientasi objek. Sebuah <i>generalization</i> dilambangkan dengan sebuah panah dengan kepala.

	<i>Dependency</i>	Kadangkala sebuah class menggunakan <i>class</i> yang lain. Hal ini disebut <i>dependency</i> . Umumnya penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan class yang lain. Sebuah <i>dependency</i> dilambangkan sebagai sebuah panah bertitik-titik.
---	-------------------	---




E. Statechart Diagram




Statechart diagram adalah suatu *diagram* yang menggambarkan daur hidup (*behaviour pattern*) dari sebuah *object* dari awal *object* tersebut di inisialisasi hingga *object* tersebut di *destroy*. Sebuah *statechart diagram* terdiri dari satu *initial state*, nol atau lebih *final state*, *state* dan *event*.

Menurut Sopiah (2012:192) “*Statechart Diagram* merupakan model perilaku yang dinamis dari kelas (*class*) secara individual maupun beberapa bentuk dari objek”.

Menurut Sopiah (2012:192) Simbol – simbol yang digunakan pada *statechart diagram* ditunjukkan pada Tabel 2.10.

Tabel 2.10 Simbol *Statechart Diagram*

GAMBAR	NAMA	KETERANGAN
	<i>Initial state</i>	<i>Initial state</i> ialah kondisi awal ketika <i>object</i> dibentuk.
	<i>Final state</i>	<i>Final state</i> adalah kondisi ketika <i>object</i> tersebut di <i>destroy</i> .
	<i>State</i>	<i>State</i> adalah keadaan atau kondisi yang sedang terjadi di <i>object</i> .

/ event transition 	<i>Event transition</i>	<i>Event transition</i> adalah <i>event</i> atau kejadian yang mengakibatkan <i>state</i> atau keadaan dalam <i>object</i> tersebut berubah.
/ event iteration 	<i>Event iteration</i>	<i>Event iteration</i> adalah <i>event</i> yang tidak mengakibatkan perubahan <i>state</i> dari <i>object</i> .
	<i>Guard condition.</i>	<i>Guard condition</i> digunakan untuk memberi keterangan atau kondisi pada <i>event</i> .

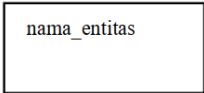
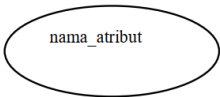
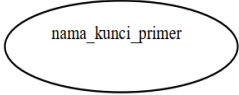
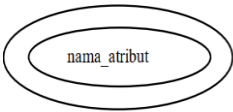
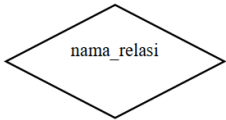
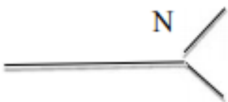
2.2.3 Pengertian ERD (*Entity Relationship Diagram*)

Menurut Sukamto dan Shalahuddin (2014:15) “*ERD* adalah pemodelan awal basis data yang dikembangkan berdasarkan teori himpunan dalam bidang matematika. *ERD* digunakan untuk pemodelan basis data relasional”.

Menurut Sukamto dan Shalahuddin (2014:16), *ERD* memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen). Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow’s Foot, dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi dari Chen.

Berikut adalah simbol-simbol yang digunakan pada *ERD* dengan notasi Chen :

Tabel 2.11 Simbol *Entity Relationship Diagram* (ERD)

GAMBAR	NAMA	KETERANGAN
	Entitas/ <i>entity</i>	Entitas merupakan data inti yang akan disimpan, bakal tabel 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 termasuk 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 <i>Multivalue</i>	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/as sociation	Penghubung antara relasi dan entitas dimana kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut kardinalitas.

2.2.4 *Hypertext Preprocessor (PHP)*

A. Pengertian PHP

Menurut Madcoms dalam Ayu dan Permata (2016 : 12) “*Hypertext Preprocessor (PHP)* adalah bahasa script yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk membuat program situs web dinamis”. PHP dapat digunakan dengan gratis (*free*) dan bersifat *Open Source*. PHP dirilis dalam lisensi *PHP license*. Untuk membuat program PHP kita diharuskan untuk menginstal web server terlebih dahulu.

Menurut sutarman dalam Rusliantoro (2016:15) “PHP (*Hypertext Preprocesscor*) adalah salah satu *server side scripting* yang di desain khusus aplikasi web. PHP dapat disisipkan diantara bahasa-bahasa HTML, dan karna *server side scripting*, maka *script* PHP akan dieksekusi di *server* sehingga yang dikirimkan ke *browser* adalah hasil jadi dalam bentuk HTML, dan kode PHP tidak akan terlihat”.

B. Sejarah PHP

PHP pertama kali dibuat pada musim gugur tahun 1994 oleh Rasmus Lerdoff, yang di gunakan oleh websitenya untuk mencatat siapa saja yang berkunjung dan melihat biodatanya. Versi pertama yang release tersedia pada awal tahun 1995. Dikenal sebagai tool Personal Home Page, yang terdiri atas engine parser yang sangat sederhana yang hanya mengerti beberapa makro khusus dan

sejumlah utilitas yang sering digunakan pada halaman-halaman web, seperti buku tamu, counter pengunjung dan lainnya.

Parser diprogram ulang pada pertengahan 1995 dan diberi nama PHP/FI versi 2.0. FI berasal dari paket Rasmus lainnya yang ditulis untuk menginterpretasikan data dari form, yang kemudian di kombinasikan dengan tool Personal Home Page dan ditambahkan dukungan untuk database mySQL (mini SQL). Pada tahun 1996 pengguna PHP/FI diperkirakan mencapai 15.000 website di seluruh dunia. Dan pada pertengahan 1997 mencapai 50.000 situs. Pada saat itu juga terdapat perubahan dalam pengembangan PHP. PHP berubah dari proyek pribadi Rasmus menjadi sebuah tim yang lebih terorganisasi. Parsenya dikembangkan oleh Zeev Suraski dan Andi Gutsman yang kemudian menjadi dasar untuk versi 3, dan banyak utilitas tambahan yang diprogram untuk menambah kemampuan dari versi sebelumnya.

C. Konsep dasar PHP

1. Sintak dasar PHP

Ada beberapa aturan sintaks yang harus dipenuhi ketika membuat file program PHP.

a PHP opening dan closing tag

b PHP mendukung komentar seperti pada bahasa 'C', 'C++', dan Unix *shell-style*. (*Perl style*).

2. Konstanta dalam PHP

Suatu konstanta dapat kita definisikan dengan menggunakan fungsi `define()` yang merupakan salah satu fitur function dari PHP.

3. Aritmatika dalam PHP

Untuk mempermudah menggunakan operand dan operator pada PHP, diberikan tabel mengenai operator seperti terlihat pada tabel ini :

Tabel 2.12 Operator Dasar Aritmatika

Tingkatan	Operator	Nama Operasi	Contoh
1	+	Positif	+3
2	-	Negative	-3
3	+	Tambah	3+3
4	-	Kurang	3-3
5	*	Kali	3*3
6	/	Bagi	3/3
7	%	Modulo	5%5

2.2.5 *Framework*

Framework adalah sebuah *software* untuk memudahkan para programmer untuk membuat sebuah aplikasi web yang di dalam nya ada berbagai fungsi diantaranya plugin, dan konsep untuk membentuk suatu sistem tertentu agar tersusun dan terstruktur dengan rapih.

Dengan menggunakan *framework* bukan berarti kita akan terbebas dengan pengkodean. Karena kita sebagai seorang pengguna *framework* haruslah menggunakan fungsi–fungsi dan variable yang ada di dalam sebuah *framework* yang kita gunakan. Untuk saat ini *framework* terbagi dua yaitu *framework* PHP dan *framework* CSS. *Framework* PHP terdiri dari : *Laravel*, *Yii*, *Code Igniter*, *Symphony*, *Zend Framework*, *Cake PHP*, *Fuel PHP*, dan lain sebagainya. *Framework* CSS terdiri dari : *Bootstrap*, *Gumby*, *Foundation*, *Less Framework*, *JQuery UI*, *Unsemantic*, *Blue Print CSS*.

2.2.6 *Laravel*

Laravel adalah sebuah *framework* PHP yang dirilis dibawah lisensi MIT, dibangun dengan konsep *Model View Controller* (MVC). *Laravel* adalah pengembangan website berbasis MVP yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan

aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu.

MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. MVC memisahkan aplikasi berdasarkan komponen-komponen aplikasi, seperti : manipulasi data, *controller*, dan *user interface*.

- a) Model, mewakili struktur data. Biasanya model berisi fungsi - fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain - lain.
- b) View, bagian yang mengatur tampilan ke pengguna. Bisa dikatakan berupa halaman web.
- c) *Controller*, merupakan bagian yang menjembatani model dan view.

Beberapa fitur yang terdapat di laravel :

- a) *Bundles*, yaitu sebuah fitur dengan sistem pengemasan modular dan tersedia beragam di aplikasi.
- b) *Eloquent ORM*, merupakan penerapan PHP lanjutan menyediakan metode internal dari pola “*active record*” yang menagatasi masalah pada hubungan objek database.
- c) *Application Logic*, merupakan bagian dari aplikasi, menggunakan controller atau bagian route.

- d) *Reverse Routing*, mendefinisikan relasi atau hubungan antara link dan route.
- e) *Restful controllers*, memisahkan logika dalam melayani HTTP GET and POST.
- f) *Class Auto Loading*, menyediakan loading otomatis untuk class PHP.
- g) *View Composer*, adalah kode unit logikal yang dapat dieksekusi ketika *view* sedang *loading*.
- h) *IoC Container*, memungkinkan objek baru dihasilkan dengan pembalikan *controller*.
- i) *Migration*, menyediakan sistem kontrol untuk skema database.
- j) *Unit Testing*, banyak tes untuk mendeteksi dan mencegah regresi.
- k) *Automatic Pagination*, menyederhanakan tugas dari penerapan halaman.

Kesimpulan Eloquent ORM, jadi dengan Eloquent ORM kita bisa mengelola data yang ada pada database dari hanya satu buah model. misalnya kita punya table siswa, maka kita juga akan mempunyai sebuah model dengan nama siswa, dengan model siswa ini kita bisa mengelola data-data yang ada pada tabel siswa dengan mudah dan cepat. Kita bisa menginput data, mengedit, menampilkan, mengupdate, bahkan kita juga bisa menggunakan relasi tabel dengan sangat mudah, dan struktur penulisan coding nya pun sangat singkat.

Jika secara manual pada PHP native, jika kita ingin mengakses atau menampilkan data dari table siswa, biasanya kita menggunakan query “select * from siswa”, sedangkan, jika kita menggunakan eloquent laravel, kita cukup mendefinisikan nama modelnya, kemudian kita bisa langsung menggunakan fungsi “all()” untuk mengambil semua data pada table siswa.

2.2.7 WEB

World Wide Web (WWW), lebih dikenal dengan web, merupakan salah satu layanan yang didapat oleh pemakai komputer yang terhubung ke internet. Web pada awalnya adalah ruang informasi dalam internet, dengan menggunakan *hyperteks* pemakai di tuntun untuk menemukan informasi dengan mengikuti link yang disediakan, karna dinilai lebih efisien dan mudah diakses dimana saja. Sehingga hal ini lebih memudahkan dalam proses pencarian informasi, pengolahan data atau *transfer* data.

Menurut Abdullah (2015:1) “Web dapat diartikan sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital baik berupa text, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet”.

Menurut Nilasari (2014:2) “Web atau sering disebut situs merupakan kumpulan halaman web yang dijalankan dari suatu alamat web domain”.

Menurut Beki (2015:35) “Web merupakan kumpulan halaman – halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan – jaringan halaman”.

2.2.8 *Hypertext Markup Language (HTML)*

Proses tampilnya sebuah halaman website di browser melibatkan HTML. *Hypertext Markup Language (HTML)* tergolong dalam salah satu format yang digunakan dalam pembuatan dokumen yang terbaca oleh web.

Menurut Solichin (2016:10) “HTML merupakan bahasa pemrograman web yang memberitahukan peramban web (*web browser*) bagaimana menyusun dan menyajikan konten di halaman web”.

Menurut Simarmata (2010:52) “HTML adalah bahasa markup untuk menyebarkan informasi pada web”.

Berdasarkan teori dari para ahli di atas, maka *Hypertext Markup Language (HTML)* merupakan bahasa pemrograman yang dikenal oleh browser untuk menampilkan informasi lebih menarik di halaman web melalui browser.

2.2.9 *Cascading Style Sheet (CSS)*

Cascading style sheet (CSS) digunakan untuk menampilkan sebuah web dengan tampilan yang menarik, memperindah tampilan web dan mudah digunakan.

Menurut Prasetio (2014:252) “CSS adalah suatu teknologi yang digunakan untuk memperindah tampilan halaman web (situs) Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML”.

Menurut Solichin (2016:10) “CSS bahasa pemrograman yang diusulkan oleh Hakon Wilum Lie pada tahun 1994 dan distandarisasi oleh W3C yang berfungsi untuk mempercantik tampilan web”.

Maka dari itu, CSS merupakan bahasa pemrograman web yang digunakan untuk mengatur konten dalam sebuah halaman web yang ditulis dalam bahasa markup agar halaman web tersebut lebih menarik dan terstruktur.

2.2.10 *Bootstrap*

Menurut Alatas (2013:40) “*bootstrap* adalah *front-end framework* yang bagus dan luar biasa yang menyediakan tampilan untuk mobile device (Handphone, smartphone dll.) guna mempercepat dan mempermudah pengembangan website, Bootstrap menyediakan HTML, CSS dan Javascript siap pakai dan mudah untuk dikembangkan”.

Bootstrap merupakan framework untuk membangun desain web secara responsif. Artinya, tampilan web yang dibuat oleh bootstrap akan menyesuaikan ukuran layar dari browser yang kita gunakan baik di desktop, tablet ataupun *mobile device*. Fitur ini bisa diaktifkan ataupun dinonaktifkan sesuai dengan keinginan kita sendiri. Sehingga, kita bisa membuat web untuk tampilan *desktop* saja dan apabila dirender oleh mobile browser maka tampilan dari web yang kita buat tidak bisa beradaptasi sesuai layar. Dengan bootstrap kita juga bisa membangun web dinamis ataupun statis.

Kelebihan dari menggunakan Bootstrap adalah kerangka ini dibangun menggunakan Less, sebuah teknologi CSS yang sederhana dan mudah untuk digunakan. Less juga menawarkan lebih banyak kekuatan dan fleksibilitas dari CSS pada umumnya. Dengan Less, pengembang dapat mengakses dengan mudah informasi dan fungsi warna, variabel, dan operasi penggunaan.

2.2.11 Normalisasi

Menurut Connolly (2010:32) “normalisasi merupakan suatu teknik untuk menghasilkan sekumpulan hubungan dengan property yang diinginkan, yang memberikan kebutuhan data terhadap suatu perusahaan”. Tujuan dari normalisasi adalah sebagai berikut :

- a. Meminimalkan jumlah atribut yang diperlukan untuk mendukung kebutuhan data dari suatu perusahaan.

- b. Untuk memperoleh atribut yang bersifat *functional dependencies*.
- c. Untuk menghilangkan data yang bersifat *redundancy* pada tiap atribut.

Menurut Connolly (2010:32) Terhadap beberapa bentuk normalisasi pada database yaitu :

1. *Unnormalized Normal form* (UNF)

Unnormalized Normal form (UNF) merupakan sebuah tabel yang mengandung satu atau lebih *repeating group*. Berikut adalah contoh dari bentuk normalisasi UNF :

No Nota	Tanggal	Jaminan	No Anggota	Nama	Telepon	Kode Buku	Judul	Jenis	Biaya Sewa	ID Petugas	Nama Petugas
210	8/10/2001	KTP	1001	Doni	82333444000	S1	Soekarno	Sejarah	20000	K01	Muslim
						O1	David Beckham	Olahraga	15000		
						K3	Kungfu Panda	Komik	10000		

Gambar 2.1 *Unnormalized Normal Form* (Connolly 2010)

2. *First Normal Form* (1NF)

First Normal Form (1NF) merupakan sebuah relasi dimana setiap potongan baris dan kolom mengandung satu dan mungkin hanya satu nilai, dan proses untuk mengubah tabel UNF ke dalam *First Normal Form* (1NF) adalah dengan cara harus diidentifikasi dan menghilangkan bagian yang mengandung *repeating group* pada tabel.

Ada dua pendekatan untuk menghilangkan perulangan kelompok (*repeating group*) dari tabel yang belum dinormalisasikan, yaitu:

- a) Dengan memasukkan data kedalam kolom kosong dari baris yang berisi perulangan data. Dengan kata lain, kita mengisi yang kosong dengan 12 duplikat data yang tidak diulang, yang diinginkan. Pendekatan ini umumnya ditunjuk sebagai '*flattening*' pada tabel.
- b) Dengan menempatkan perulangan data, sepanjang dengan sebuah salinan atribut kunci yang asli ke dalam sebuah relasi yang terpisah. Terkadang tabel yang belum dinormalisasi mungkin berisi lebih dari satu perulangan kelompok. Pada kasus seperti ini, pendekatan dapat diulang sampai tidak ada lagi perulangan yang terjadi. Sebuah set relasi dapat berada pada 1NF jika tidak terdapat perulangan kelompok (*repeating group*).

Berikut adalah contoh bentuk normalisasi 1NF :

No Nota	Tanggal	Jaminan	No Anggota	Nama	Telepon	Kode Buku	Judul	Jenis	Biaya Sewa	ID Petugas	Nama Petugas
210	8/10/2001	KTP	1001	Doni	82333444000	S1	Soekarno	Sejarah	20000	K01	Muslim
210	8/10/2001	KTP	1001	Doni	82333444000	O1	David Beckham	Olahraga	15000	K01	Muslim
210	8/10/2001	KTP	1001	Doni	82333444000	K3	Kungfu Panda	Komik	10000	K01	Muslim

Gambar 2.2 *First Normal Form (Connolly 2010)*

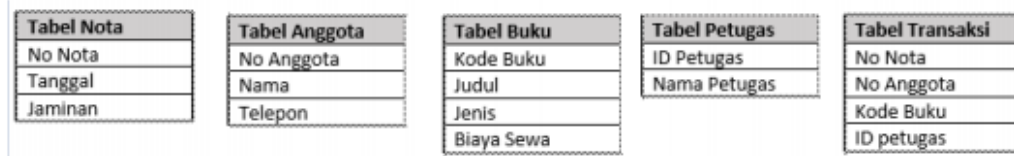
3. *Second Normal Form (2NF)*

Second Normal Form (2NF) dapat dihasilkan dengan cara melihat apakah ada atribut yang bukan merupakan *primary key* dapat merupakan fungsi dari sebagian *primary key (partial dependence)*.

Dalam bentuk normal kedua setiap atribut yang bergantung secara parsial harus dipisahkan. Bentuk normal akan diperoleh bila setiap

atribut yang bukan merupakan *primary key* dari suatu tabel secara penuh yang merupakan functional dependence dari *primary key* itu.

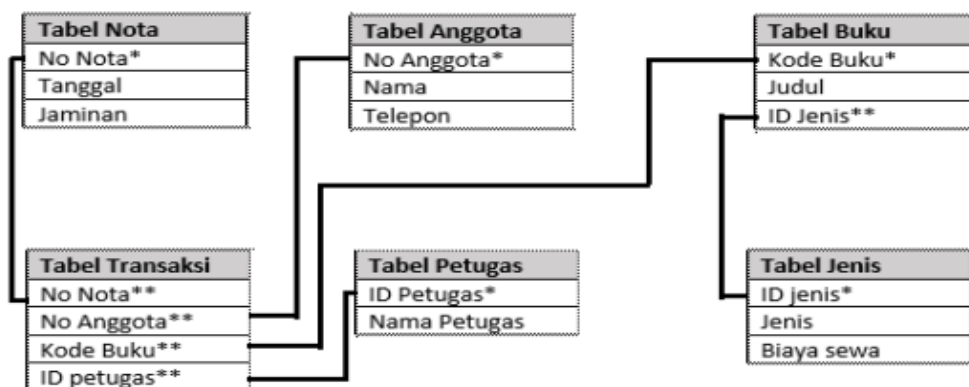
Berikut adalah contoh bentuk normalisasi 2NF :



Gambar 2.3 Second Normal Form (Connolly 2010)

4. Third Normal Form (3NF)

Third Normal Form (3NF) akan secara langsung dilakukan pengujian dengan cara melihat apakah terdapat atribut bukan *key* yang bergantung *funksional* terhadap atribut yang bukan *key* yang lain atau disebut (*transitive dependence*). Dengan cara yang sama, maka setiap *transitive dependence* harus dipisahkan. *Third Normal Form* (3NF) dapat dikatakan sudah normal apabila anomali yang ada di dalamnya sudah tidak ada, pada kasus tertentu normalisasi dilakukan sampai BCNF. Berikut adalah contoh normalisasi bentuk 3NF :



Gambar 2.4 Third Normal Form (Connolly 2010)

2.2.12 Basis Data (*Database*)

Sistem yang terkomputerisasi memiliki basis data untuk memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan.

Menurut Sukanto 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”.

Menurut Lubis (2016 : 2) “Basis Data merupakan gabungan file data yang dibentuk dengan hubungan atau relasi yang logis dan dapat diungkapkan dengan catatan serta bersifat independent”.

Maka dari itu, basis data merupakan sekumpulan data atau informasi yang telah diolah dan tersimpan serta dapat digunakan kembali apabila dibutuhkan.

2.2.13 MySQL (*My Structure Query Language*)

MySQL dikategorikan sebagai perangkat lunak dan sistem pembuat database yang bersifat terbuka (*Open Source*) dan berjalan di berbagai sistem operasi baik di *Windows* maupun *Linux*.

Menurut Wahana Komputer (2010:111) “MySQL adalah salah satu aplikasi server yang digunakan untuk manajemen suatu data dan banyak digunakan khalayak di seluruh dunia. Fungsi terpenting dari

MySQL adalah sebagai content management suatu *website*, yaitu mengatur isi informasi yang ditampilkan suatu *website*".

Menurut Ahmar (2013:11) "MySQL adalah sistem yang berguna untuk melakukan proses pengaturan koleksi-koleksi struktur data (*database*) baik meliputi proses pembuatan atau proses pengelolaan *database*".

Jadi MySQL adalah sebuah perangkat lunak yang dapat membuat *database* yang bersifat *open source* dan sebagai sumber dan pengolahan data untuk membangun aplikasi *web*. MySQL sering digunakan oleh programmers dikarenakan kelebihan-kelebihan yang dimiliki oleh MySQL.

Berikut tipe-tipe data yang didukung oleh MySQL, antara lain:

1. Tipe data untuk bilangan

Tabel 2.13 Tipe Data MySQL Bilangan

NO	Tipe Data	Bentuk Dasar Penulisan	Keterangan	Ukuran
1.	TINYINT	TINYINT [(M)] [UNSIGNED] [ZEROFILL]	Digunakan untuk menyimpan data bilangan bulat positif dan negatif. Bilangan bulat terkecil, dengan jangkauan untuk bilangan bertanda -128 sampai dengan 127 dan untuk yang tidak bertanda 0 sampai dengan 255. Bilangan tak bertanda ditandai dengan kata	1 Byte

			UNSIGNED	
2.	SMALLINT	SMALLINT [(M)] [UNSIGNED] [ZEROFILL]	<p>Digunakan untuk menyimpan data bilangan bulat positif dan negatif.</p> <p>Jangkauan untuk bilangan bertanda -32768 sampai dengan 32767 dan untuk yang tidak bertanda (jangkauan unsigned) 0 sampai dengan 65535</p>	2 Byte
3.	MEDIUMINT	MEDIUMINT [(M)] [UNSIGNED] [ZEROFILL]	<p>Digunakan untuk menyimpan data bilangan bulat positif dan negatif.</p> <p>Jangkauan untuk bilangan bertanda -8388608 sampai dengan 8388607 dan untuk yang tidak bertanda (jangkauan unsigned) 0 sampai dengan 16777215</p>	3 Byte
4.	INT	INT [(M)] [UNSIGNED] [ZEROFILL]	<p>Digunakan untuk menyimpan data bilangan bulat positif dan negatif .</p> <p>Jangkauan untuk bilangan bertanda -2147483648 sampai dengan 2147483647 dan untuk yang tidak bertanda (jangkauan unsigned) 0 sampai dengan 4294967295</p>	4 Byte
5.		INTEGER [(M)] [UNSIGNED] [ZEROFILL]	Sama dengan INT.	

6.	BIGINT	BIGINT [(M)] [UNSIGNED] [ZEROFILL]	<p>Digunakan untuk menyimpan data bilangan bulat positif dan negatif .</p> <p>Jangkauan untuk bilangan bertanda - 9223372036854775808 sampai dengan 9223372036854775807 dan untuk yang tidak bertanda 0 sampai dengan 18446744073709551615</p>	8 Byte
7.	FLOAT	FLOAT [(M,D)] [ZEROFILL]	<p>Digunakan untuk menyimpan data bilangan pecahan positif dan negatif presisi tunggal.</p> <p>Jangkauan: - 3.402823466E+38 s/d - 1.175494351E-38, 0, dan 1.175494351E-38 s/d 3.402823466E+38</p>	4 Byte
8.	DOUBLE	DOUBLE [(M,D)] [ZEROFILL]	<p>Digunakan untuk menyimpan data bilangan pecahan positif dan negatif presisi ganda.</p> <p>Tidak dapat bersifat unsigned.</p> <p>Nilai yang diijinkan adalah - 1.7976931348623157E+308 hingga - 2.2250738585072014E-308 untuk nilai negatif, 0, dan 2.2250738585072014E-308 hingga 1.7976931348623157E+308 untuk nilai positif.</p>	8 Byte

9.		DOUBLE PRECISION [(M,D)] [ZEROFILL]	Bilangan pecahan berpresisi ganda.	8 Byte
10.		REAL [(M,D)] [ZEROFILL]	Sinonim dari DOUBLE.	8 Byte
11.	DECIMAL	DECIMAL [(M,D)] [ZEROFILL]	Digunakan untuk menyimpan data bilangan pecahan positif dan negatif. Misalnya DECIMAL(5,2) dapat digunakan untuk menyimpan bilangan -99,99 sampai dengan 99,99	M Byte
12.		NUMERIC [(M,D)] [ZEROFILL]	Sama dengan Decimal.	M Byte

2. Tipe Data untuk Tanggal dan Jam

Tabel 2.14 Tipe Data MySQL Tanggal dan Jam

NO	Tipe Data	Bentuk Penulisan Dasar	Keterangan	Ukuran
1.	DATETIME		Kombinasi tanggal dan waktu dengan jangkauan dari '1000-01-01 00:00:00' sampai dengan '9999-12-31 23:59:59'	8 Byte
2.	DATE		Kombinasi tanggal dan waktu dengan jangkauan dari '1000-01-01' sampai dengan '9999-12-31'	8 Byte
3.	TIMESTAMP	TIMESTAMP[(M)]	Kombinasi tanggal dan waktu dengan jangkauan dari	4 Byte

			'1970-01-01' sampai dengan '2037'	
4.	TIME		Digunakan untuk menyimpan waktu dengan jangkauan dari -838:59:59 sampai dengan 838:59:59	3 Byte
5.	YEAR		Digunakan untuk menyimpan data tahun dari tanggal antara 1901 sampai dengan 2155	1 Byte

3. Tipe data untuk karakter dan Lain-lain

Tabel 2.15 Tipe Data MySQL Karakter dan lain-lain

NO	Tipe Data	Bentuk Dasar Penulisan	Keterangan	Ukuran
1.	CHAR	CHAR(M) [BINARY]	Data string dengan panjang yang tetap. CHAR(1) cukup ditulis dengan CHAR. $1 \leq M \leq 255$, Jika ada sisa, maka sisa tersebut diisi dengan spasi (misalnya nilai M adalah 10, tapi data yang disimpan hanya memiliki 7 karakter, maka 3 karakter sisanya diisi dengan spasi). Spasi ini akan dihilangkan apabila data dipanggil. Nilai dari CHAR akan disortir dan diperbandingkan secara case-insensitive menurut default character set yang tersedia, kecuali bila atribut BINARY disertakan.	M Byte
2.	VARCHAR	VARCHAR(M) [BINARY]	Ukuran L+1 byte dengan $L \leq M$ dan $1 \leq M \leq 255$. Data string dengan panjang bervariasi tergantung datanya. Jika nilai M adalah 10 sedangkan data yang disimpan	L+1 Byte

			hanya terdiri dari 5 karakter, maka lebar data tersebut hanya 5 karakter saja, tidak ada tambahan spasi.	
3.	TINYBLOB, TINYTEXT		L+1 byte, dengan $L < 2^8$. Tipe TEXT atau BLOB dengan panjang maksimum 255 karakter.	L+1 Byte
4.	BLOB, TEXT		L+2 byte, dengan $L < 2^{16}$. Tipe TEXT atau BLOB dengan panjang maksimum 65535 karakter.	L+2 Byte
5.	MEDIUMBLOB, MEDIUMTEXT		L+3 byte, dengan $L < 2^{24}$. Tipe TEXT atau BLOB dengan panjang maksimum 1677215 karakter.	L+3 Byte
6.	LONGBLOB, LONGTEXT		L+4 byte, dengan $L < 2^{32}$. Tipe TEXT atau BLOB dengan panjang maksimum 4294967295 karakter.	L+4 Byte
7.	ENUM	ENUM('nilai1', 'nilai2',...)	Ukuran 1 atau 2 byte tergantung nilai enumerasinya maks 65535 nilai.	1 atau 2 Byte
8.	SET	SET('nilai1', 'nilai2',...)	Ukuran 1,2,3,4 atau 8 byte tergantung jumlah anggota himpunan maks 64 anggota.	1, 2, 3, 4, atau 8 Byte

2.2.14 Perangkat Lunak Pendukung

Perangkat lunak pendukung yang digunakan ada tiga macam aplikasi karena komputer memerlukan suatu aplikasi dalam pembuatan aplikasi lainnya, aplikasi yang digunakan adalah *Microsoft Office Visio 2007*, *Enterprise Architect* dan *Adobe XD*.

1. *Microsoft Office Visio*

Menurut Frediryna (2015:16) “*Microsoft Office Visio* merupakan suatu program aplikasi komputer untuk membantu membuat diagram alir (*flowchart*) dan pembuatan gambar teknik seperti desain *office*, *floor plan*, desain elektrik, dan lain-lain”.

2. *Enterprise Architect*

Enterprise Architect adalah sebuah *platform* visualisasi untuk merancang dan membangun sebuah sistem perangkat lunak, untuk pemodelan proses bisnis, dan untuk lebih umumnya digunakan tujuan pemodelan. *Enterprise Architect* adalah sebuah alat yang sangat maju yang mencakup seluruh aspek dari mulai siklus pengembangan, memberikan penelusuran secara mendetail atau penuh dari mulai awal tahap mendesain sampai mengontrol perkembangan, pemeliharaan, pengujian dan perubahan kontrol. Dimana *tools Enterprise Architect* dikembangkan oleh *Sparx Systems* sebuah perusahaan Australia dengan inovasi dan pengembangan dalam pemodelan UML. *Sparx System* adalah anggota dari *Object Management Group (OMG)*.

Unified Modeling Language (UML) adalah sebuah bahasa grafis untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan bentuk sebuah sistem perangkat lunak. UML juga menawarkan cara standar untuk menulis di sistem, termasuk konseptualal-hal seperti proses bisnis dan fungsi sistem serta hal-hal konkret seperti pernyataan sebuah bahasa pemrograman, skema *database*, dan perangkat lunak dapat digunakan kembali komponen perangkat lunak.

3. Adobe XD

Adobe XD adalah alat desain pengalaman pengguna berbasis vektor untuk aplikasi seluler, yang dikembangkan dan diterbitkan oleh *Adobe Inc.* aplikasi ini digunakan untuk membuat atau merancang desain *interface* sebuah aplikasi, baik *desktop*, *website* ataupun *mobile*. *Adobe XD* juga merupakan sebuah *tools* untuk mendesain sebuah *User Interface* yang tersedia untuk *macOS* dan *windows* meskipun ada versi untuk *IOS* dan *Android* juga dapat membantu mempratinjau hasil kerja langsung pada perangkat seluler.

4. Visual Studio Code (VSCode)

Menurut yang dilansir dalam situs website visual studio code (<https://code.visualstudio.com/>) Visual Studio Code atau bisa disebut VSCode adalah text editor kode ringan serta ekstensi yang

memiliki dukungan hampir setiap bahasa pemrograman dan ,
seperti :

1. Python.
2. C/C++.
3. C#.
4. PHP.
5. Ruby.
6. HTML;
7. CSS.
8. JavaScript

2.2.15 Pengertian XAMPP

Menurut Madcoms dalam Ayu dan Permata (2018 : 20) “Xampp adalah sebuah paket kumpulan *software* yang terdiri dari Apache, MySQL, PhpMyAdmin, PHP, Perl, Filezilla, dan lain.”

Xampp berfungsi untuk memudahkan instalasi lingkungan PHP, dimana biasanya lingkungan pengembangan web memerlukan PHP, Apache, MySQL dan PhpMyAdmin.

Melalui program ini, programmer web dapat menguji aplikasi web yang dikembangkan dan mempresentasikannya ke pihak lain secara langsung dari komputer, tanpa perlu terkoneksi ke internet.

XAMPP juga dilengkapi fitur manajemen basis data PHPMyAdmin seperti pada server hosting sungguhan, sehingga

pengembang web dapat mengembangkan aplikasi web berbasis database secara mudah.

2.2.16 Black Box Testing

Pengujian black box merupakan pengujian dimana kasus uji didesain berdasarkan spesifikasi dan berfokus pada output yang dihasilkan sebagai respon dari input yang dipilih dan kondisi-kondisi eksekusi (Srinivas dan Jagruthi. 2012 : 12). Kelebihan dari pengujian black box adalah :

- a. Penguji tidak memerlukan pengetahuan akan bahasa pemrograman yang digunakan.
- b. Membantu menemukan ambiguitas dan ketidakkonsistenan yang terjadi antara penguji dan pengembang.

2.2.17 Pengertian User Requirements Spesification

Menurut yang dilansir dalam situs website ofni system (<http://www.ofnisystems.com/services/validation/user-requirement-specifications/>) *User Requirements Spesification* adalah menggambarkan kebutuhan bisnis untuk apa yang dibutuhkan pengguna dari sistem. *user requirements specification* ditulis pada awal proses validasi, biasanya sebelum sistem dibuat. Mereka ditulis oleh pemilik sistem dan pengguna terakhir, dengan masukan dari jaminan kualitas. Persyaratan yang diuraikan dalam URS biasanya

diuji dalam kualifikasi kinerja atau pengujian penerimaan pengguna. *user requirements specification* tidak dimaksudkan sebagai dokumen teknis.

URS pada umumnya adalah dokumen perencanaan, dibuat ketika sebuah bisnis berencana untuk memperoleh suatu sistem dan berusaha menentukan kebutuhan spesifik. Ketika suatu sistem telah dibuat atau diperoleh, atau untuk sistem yang kurang kompleks, spesifikasi dapat dikombinasikan dengan dokumen persyaratan fungsional.

2.2.18 SDLC (*System Development Life Cycle*)

SDLC (*Systems Development Life Cycle*) dalam rekayasa sistem dan rekayasa perangkat lunak, adalah proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem tersebut. Konsep ini umumnya merujuk pada sistem komputer atau informasi. SDLC juga merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak, yang terdiri dari tahap-tahap:

- a Rencana (*Planning*),
- b Analisis (*Analysis*),
- c Desain (*Design*),
- d Implementasi (*Implementation*),
- e Uji Coba (*Testing*),
- f Dan Pengelolaan (*Maintenance*).

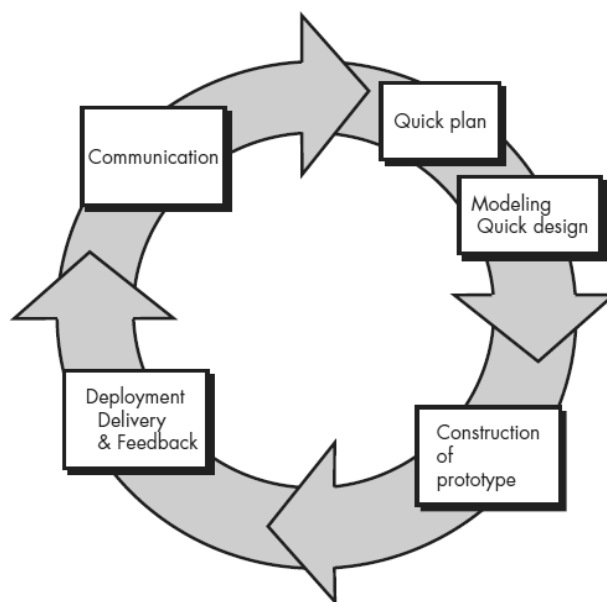
Dalam rekayasa perangkat lunak, konsep SDLC mendasari berbagai jenis metodologi pengembangan perangkat lunak. Metodologi-metodologi ini membentuk suatu kerangka kerja untuk perencanaan dan pengendalian pembuatan sistem informasi, yaitu proses pengembangan perangkat lunak. Terdapat tiga jenis metode siklus hidup sistem yang paling banyak digunakan, yakni siklus hidup sistem tradisional (*traditional system life cycle*), siklus hidup menggunakan prototyping (*life cycle using prototyping*), dan siklus hidup sistem orientasi objek (*object-oriented system life cycle*).

SDLC juga merupakan metodologi umum dalam pengembangan sistem yang menandai kemajuan usaha analisis dan desain. Konsep dalam SDLC mendasari model pengembangan perangkat lunak tersebut antara lain *waterfall*, *prototype*, *iterative*, *spiral*, *rapid application development* (RAD) dan lainnya.

2.2.19 Pengertian *Prototype*

Menurut Roger S Pressman (2012) , Metode *Protortype* merupakan metodologi pengembangan software yang menitik beratkan pada pendekatan aspek desain, fungsi dan *user-interface*. *Developer* dan *user* fokus pada *user-interface* dan bersama-sama mendefinisikan spesifikasi, fungsi, desain dan bagaimana *software* bekerja. *Developer* dan *user* bertemu dan melakukan komunikasi dan menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang

akan dibutuhkan. *Developer* mengumpulkan detail dari kebutuhan dan memberikan suatu gambaran dengan cetak biru (*prototype*). Dari proses tersebut akan diketahui detail-detail yang harus dikembangkan atau ditambahkan oleh *developer* terhadap cetak biru, atau menghapus detail-detail yang tidak diperlukan oleh *user*. Proses akan terjadi terus menerus sehingga produk sesuai dengan keinginan dari *user*.



Gambar 2.5 Model Prototype Menurut (Roger S Pressman 2012)

Adapun tahapan-tahapan Prototyping sebagai berikut :

- a. *Communication.*
- b. *Quick Plan and Modelling Quick Design.*
- c. *Construction of Prototype.*
- d. *Deployment Delivery and Feedback*

2.2.20 Elisitasi

Menurut Yousuf dan Asgar dalam International Journal Of Computer Applications (2019:11), *“Requirements Elicitation (RE) is defined as the process of obtaining a comprehensive understanding of stakeholder’s requirements. It is the initial and main process of requirements engineering phase. Elicitation process usually involves interaction with stakeholders to obtain their real needs”*. Yang artinya Persyaratan *Elicitation* (RE) didefinisikan sebagai proses untuk mendapatkan pemahaman yang komprehensif tentang kebutuhan pemangku kepentingan. Ini adalah proses awal dan utama dari fase rekayasa kebutuhan. Proses elisitasi biasanya melibatkan interaksi dengan para pemangku kepentingan untuk mendapatkan kebutuhan nyata mereka. Elisitasi merupakan pengumpulan kebutuhan aktivitas awal dalam rekayasa kebutuhan (*Requirements Engineering*). Menurut Siahaan dalam Dzulhaq (2017:13). Elisitasi berisi usulan rancangan sistem baru yang diinginkan oleh pihak manajemen terkait dan disanggupi oleh penulis untuk dieksekusi.

Tahap-Tahap Elisitasi Menurut Bachtiar dan Atikah (2015:51) Elisitasi dilakukan melalui tiga tahap yaitu sebagai berikut:

1. Berisi seluruh rancangan sistem baru yang diusulkan oleh pihak manajemen terkait melalui proses wawancara.
2. Merupakan hasil pengklasifikasian dari elisitasi tahap I berdasarkan metode MDI. Metode MDI ini bertujuan untuk

memisahkan antara rancangan sistem yang penting dan harus ada pada sistem baru dengan rancangan yang disanggupi oleh penulis untuk dieksekusi.

- a. “M” pada MDI itu artinya Mandatory (penting).

Maksudnya requirement tersebut harus ada dan tidak boleh dihilangkan pada saat membuat sistem baru.

- b. “D” pada MDI itu artinya Desirable.

Maksudnya requirement tersebut tidak terlalu penting dan boleh dihilangkan. Tetapi jika requirement tersebut digunakan dalam pembentukan sistem, akan membuat sistem tersebut lebih sempurna

- c. “I” pada MDI itu artinya Inessential.

Maksudnya requirement tersebut tidak terlalu penting dan boleh dihilangkan. Tetapi jika requirement tersebut digunakan dalam pembentukan sistem, akan membuat sistem tersebut lebih sempurna.

3. Merupakan hasil penyusutan dari elisitasi tahap II dengan cara mengeliminasi semua requirement yang optionnya I pada MDI. Selanjutnya semua requirement yang tersisa diklasifikasikan kembali melalui metode TOE, yaitu sebagai berikut:

- a. “T” artinya Technical.

Maksudnya bagaimana tata cara/teknik pembuatan requirement tersebut dalam sistem yang diusulkan.

- b. “O” artinya Operational

Maksudnya bagaimana tata cara penggunaan requirement tersebut dalam sistem yang akan dikembangkan.

- c. “E” artinya Economy

Maksudnya berapakah biaya yang diperlukan guna membangun requirement tersebut di dalam sistem.

Metode TOE tersebut dibagi kembali menjadi beberapa option, yaitu :

- a. High (H): yaitu sulit untuk dikerjakan, karena teknik pembuatan dan pemakaiannya sulit serta biayanya mahal. Sehingga requirement tersebut harus dieliminasi.
- b. Middle (M): yaitu mampu untuk dikerjakan.
- c. Low (L): yaitu mudah untuk dikerjakan.

- 4. Merupakan hasil akhir yang dicapai dari suatu proses elisitasi yang dapat digunakan sebagai dasar pembuatan suatu sistem yang akan dikembangkan.

2.3 Konsep Dasar *Literature Review*

2.3.1 Definisi *Literature Review*

Literature review adalah sekumpulan jurnal, atau penelitian lain yang akan menjadi sebuah acuan penelitian peneliti, dalam penelitian yang dilakukan ilmuwan sebelumnya dapat memberikan masukan atau ide yang didalamnya hampir memiliki topik penelitian yang sama.

Menurut Hasibuan dalam Rasdiana (2013:2) “*Literature review* berisi uraian tentang teori, temuan dan bahan penelitian lain yang diperoleh dari bahan acuan untuk dijadikan landasan kegiatan penelitian.”

“*Literature review* adalah uraian tentang teori, temuan dan bahan penelitian lainnya yang diperoleh dari bahan acuan untuk dijadikan landasan kegiatan penelitian untuk menyusun kerangka pemikiran yang jelas dari perumusan masalah yang ingin diteliti” (Deviachrista, 2013).

Berdasarkan beberapa pendapat di atas, dapat disimpulkan *literature review* adalah berupa bahan uraian teori yang diperoleh dan digunakan untuk melihat dan menganalisa nilai tambah penelitian serta untuk dijadikan landasan kegiatan penelitian.

2.3.2 Manfaat *Literature Review*

Manfaat dari Studi Pustaka (*Literature Review*) yaitu:

1. Mengidentifikasi kesenjangan (*identify gaps*).
2. Menghindari pembuatan ulang (*reinventing the wheel*).
3. Mengidentifikasi metode yang pernah dilakukan.
4. Meneruskan penelitian sebelumnya, serta mengetahui orang lain yang spesialisasi dan area penelitiannya sama di bidang ini

2.3.3 Sumber *Literature Review*

Adapun *literature review* yang peneliti ambil sebagai landasan awal ataupun pendukung dari kegiatan penelitian, yaitu :

1. Penelitian yang dilakukan Farida Nursiyam dan Dr. Anne Ratnasari, Dra., M.Si (2018) yang berjudul “Strategi *Event Organizer* dalam Menyelenggarakan Pameran” Tujuan dari penelitian mereka adalah untuk mengetahui manajemen event yang dilakukan oleh PT. Mediatama Binakreasi Media komunikasi apa yang banyak digunakan dalam menginformasikan pameran dagang INACRAFT 2018, serta mengapa media online menjadi media andalan untuk menginformasikan media online pada khalayak.
2. Penelitian yang dilakukan Taufiq Iqbal Fadillah, Guntur Prabawa Kusuma, dan Hadi Prasetyo Utomo (2015) yang berjudul “Aplikasi Periklanan dan Pemesanan Tiket Event Organizer Berbasis Web” Tujuan dari penelitian mereka adalah agar memudahkan pemilik

acara ataupun pemasangan iklan untuk mempromosikan acara dan iklan mereka dan dengan mudah melakukan pemesanan tiket langsung melalui website yang tersedia sehingga pembeli tidak perlu mengantre panjang untuk mendapatkan tiket.

3. Penelitian yang dilakukan Rhesa Elia Nugroho (2018) yang berjudul “Pembuatan Sistem Informasi ”ETICK” (Event Registration and Ticketing) Menggunakan Framework Laravel” Tujuan dari penelitian adalah untuk dapat melakukan unggah informasi dan promosi event yang akan diselenggarakan serta melakukan penjualan tiket elektronik, sehingga lebih mudah tanpa harus datang ke loket tiket.