

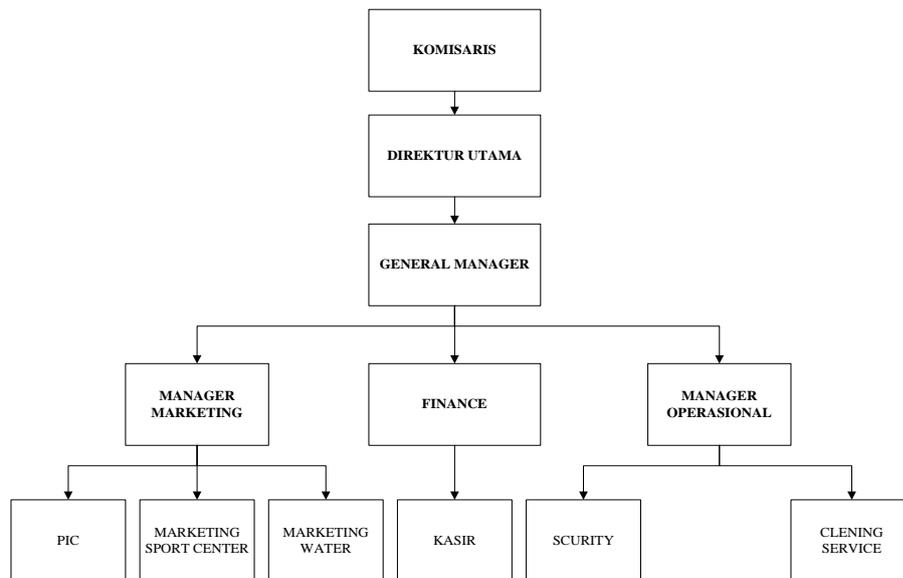
BAB II

LANDASAN TEORI

2.1 Profil The Dome Sport Arena

The Dome Sport Arena Lampung adalah salah satu tempat olahraga yang ada di kota Bandar Lampung. Dimana The Dome Sport Arena memiliki fasilitas berupa : 4 buah Lapangan Futsal, dimana lapangan tersebut menggunakan karpet *synthetic*. The Dome Sport Arena juga memiliki kolam renang.

2.1.1 Struktur Organisasi



Gambar 2.1 Struktur Organisasi The Dome Sport Arena

2.2 Teori Dasar Sistem

2.2.1 Rancang Bangun

Rancang bangun atau desain dalam pengembangan sistem perangkat lunak merupakan upaya untuk mengkonstruksi sebuah sistem yang memberikan kepuasan (mungkin formal) akan kebutuhan fungsional, memenuhi target, memenuhi kebutuhan secara implisit atau eksplisit dari segi performansi maupun penggunaan sumber daya, kepuasan batasan pada proses desain dari segi biaya, waktu, dan perangkat. (Rosa A.S & M. Salahuddin, 2015)

2.2.2 Sistem

Menurut Tata Sutabri (2012) sistem adalah sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu. Model umum sebuah sistem adalah *input*, proses, dan *output*. Hal ini merupakan konsep sebuah sistem yang sangat sederhana sebab sebuah sistem dapat mempunyai beberapa masukan dan keluaran. Selain itu, sebuah sistem mempunyai karakteristik atau sifat-sifat tertentu yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun karakteristik yang dimaksud adalah sebagai berikut:

1. Komponen Sistem (*Component*)

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 “supra sistem”.

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. Dengan demikian, lingkungan luar tersebut harus tetap dijaga dan dipelihara. Lingkungan luar yang merugikan harus dikendalikan. Kalau tidak, maka akan mengganggu kelangsungan hidup sistem tersebut.

4. 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 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 kedalam sistem tersebut masukkan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Contoh, di dalam suatu unit sistem komputer, “Program” adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan “Data” adalah *signal input* untuk diolah menjadi informasi.

6. 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 masukan untuk pengambilan keputusan atau hal-hal lain yang menjadi *input* bagi subsistem lain.

7. Pengolahan Sistem (*Process*)

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

8. 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 dan tujuan yang telah direncanakan.

2.2.3 Informasi

Menurut Tata Sutabri (2012) Informasi adalah data yang telah diklasifikasi atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi mengolah data menjadi informasi atau tepatnya mengolah data dari tak berguna menjadi berguna bagi penerimanya. Nilai informasi berhubungan dengan keputusan maka informasi menjadi tidak diperlukan keputusan dapat berkisar dari keputusan berulang sederhana sampai keputusan strategis jangka panjang. Nilai informasi dilukiskan paling berarti dalam konteks sebuah keputusan.

2.2.4 Sistem Informasi

Menurut Tata Sutabri (2012) Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk menyediakan kepada pihak luar tertentu dengan laporan- laporan yang diperlukan.

2.2.5 Penyewaan

Menurut Abdul R. Husein (2009), penyewaan adalah, “Peluang bisnis yang sering dimanfaatkan oleh banyak orang sehingga muncul bisnis penyewaan barang-barang, seperti: rental mobil, penyewaan gedung serba guna, penyewaan buku atau dvd penyewaan kontainer, dan sebagainya”. (Abdul R. Husein. 2009) Selain itu penyewaan dapat diartikan sebagai peminjaman jasa atau barang dengan tidak mengabaikan suatu ketentuan atau

kesepakatan dan syarat yang berlaku didalam organisasi tersebut guna mencapai satu tujuan. Barang yang dapat disewa bermacam-macam, tarif dan lama sewa juga bermacam-macam. Rumah umumnya disewa dalam satuan tahun, mobil dan gedung dalam satuan hari, permainan komputer seperti playstation disewa dalam satuan jam, buku yang disewa dalam satuan hari, dan kontainer yang disewa dalam satuan muatan. Tujuan dan Fungsi Penyewaan Pada dasarnya tujuan utama penyewaan adalah untuk memenuhi kebutuhan para konsumen, memperoleh laba dan meningkatkan taraf hidup. Adapun tujuan dan fungsi penyewaan adalah sebagai berikut : a. Untuk memenuhi kebutuhan konsumen b. Untuk menjalankan fungsi perekonomian c. Mendapatkan keuntungan dari masing–masing pihak yaitu penyewa dan pemberi sewa d. Penyewa tidak perlu membeli suatu barang atau hanya sebagai pemakai saat membutuhkan saja.

2.2.6 Penjadwalan

Menurut (Pinedo, 2002) penjadwalan selalu berhubungan dengan pengalokasian sumber daya yang ada pada jangka waktu tertentu, hal tersebut adalah proses pengambilan keputusan yang tujuannya adalah optimalitas.

2.3 HTTP

Menurut Yeni Kustiyahningsih dan Devie Rosa Anamisa(2011) HTTP (*Hypertext Transfer Protocol*) adalah suatu protokol yang menentukan aturan yang perlu diikuti oleh *web browser* dalam meminta atau mengambil suatu dokumen dan

menyediakan dokumen yang diminta oleh *browser*. Protokol ini merupakan protokol standar yang digunakan untuk mengakses halaman HTML. Server HTTP umumnya digunakan untuk melayani dokumen *hypertext*, karena HTTP adalah protokol dengan *overhead* yang sangat rendah, sehingga pada kenyataannya navigasi informasi dapat ditambahkan langsung kedalam dokumen.

2.4 Web Browser

Menurut Yeni Kustiyahningsih dan Devie Rosa Anamisa(2011)*web browser* adalah software yang digunakan untuk menampilkan informasi dari server web. Software ini kini telah dikembangkan dengan menggunakan *user interface grafis*, sehingga pemakai dapat dengan melakukan '*point and click*' untuk pindah antar dokumen. Dapat dikatakan saat ini hanya ada empat *web browser* GUI yang populer yaitu : Internet Explorer, Netscape Navigator, Opera dan Mozilla, Google Chrome.

2.5 Alat dan Teknik Pengembangan Sistem

2.5.1 Metode *Prototype Model*

Pressman (2012, p50) metode prototyping sebagai suatu paradigma dimulai dengan pengumpulan kebutuhan, pengembangan bertemu dengan pengguna dan mengidentifikasi objektif keseluruhan dari perangkat lunak, selanjutnya mengidentifikasi segala kebutuhan yang diketahui secara garis besar dimana definisi-definisi lebih jauh merupakan keharusan, kemudian dilakukan perancangan kilat, lalu diakhiri dengan evaluasi. Metode prototyping dibagi dalam 5 sistem antara lain :

1) Analisis dan definisi persyaratan

Analisis dan definisi persyaratan adalah proses pengumpulan kebutuhan dilakukan secara intensif untuk memesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2) Perancangan sistem

Perancangan sistem adalah proses multistep yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

3) Implementasi dan pengujian unit

Perancangan sistem dan perangkat lunak ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4) Integrasi dan pengujian sistem

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

5) Operasi dan pemeliharaan

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karna adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru. Berikut contoh gambar 2.2.



Gambar 2.2 Sistem Model *prototyping*

Kelebihan dari metode *prototyping* ini sebagai berikut :

- 1) Adanya komunikasi yang baik antara pengembang dan pelanggan
- 2) Pengembangan dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan.
- 3) Lebih menghemat waktu dalam pengembangan sistem.
- 4) Penerapan menjadi lebih mudah karna pemakaian mengetahui apa yang diharapkannya.

Kekurangan dari metode prototype ini sebagai berikut :

- 1) Resiko tinggi yaitu untuk masalah-masalah yang tidak terstruktur dengan baik, ada perubahan yang besar dari waktu ke waktu, dan adanya persyaratan data yang tidak menentu.
- 2) Interaksi pemakai penting. Sistem harus menyediakan dialog online antara pelanggan dan komputer.

Hubungan pelanggan dengan komputer yang disediakan mungkin tidak menceritakan teknik perancangan yang baik.

2.6 Database

Menurut Verdi Yasin S.Kom.,M.Kom (2012) mengatakan Basis data (*database*) adalah kumpulan informasi yang akan disimpan didalam komputer secara sistematis, sehingga dapat digunakan oleh suatu program komputer untuk memperoleh informasi dari basis data tersebut. Basis data adalah sekumpulan data yang terhubung satu sama lain secara logika dan suatu deskripsi data yang dirancang untuk memenuhi kebutuhan informasi suatu organisasi. Pendefinisian basis data meliputi spesifikasi dari tipe data, struktur dan batasan dari data atau informasi yang akan disimpan. *Database* merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi pada para pengguna atau *user*. Adapun tujuan dari database antara lain :

1. Kecepatan dan Kemudahan (*Speed*)

Pemanfaatan basis data memungkinkan kita untuk dapat menyimpan data atau melakukan perubahan/manipulasi terhadap data atau menampilkan kembali data tersebut dengan lebih cepat dan mudah.

2. Efisiensi Ruang Penyimpanan(*Space*)

Karena keterkaitan erat antara kelompok dalam basis data, maka redundansi data pasti selalu ada. Dengan basis data,efisiensi/optimalisasi penggunaan ruang penyimpanan dapat dilakukan karena kita dapat melakukan penekanan jumlah redundansi data, baik menerapkan sejumlah pengkodean atau membuat relasi-relasi antar kelompok data yang saling berhubungan.

3. Keakuratan (*Accuracy*)

Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan/batasan tipe data, domain data, keunikan data dan sebagainya yang secara ketat dapat diterapkan dalam sebuah basis data, sangat berguna untuk menekan ketidakakuratan/penyimpanan data.

4. Ketersediaan (*Availability*)

Pertumbuhan data sejalan waktu akan semakin membutuhkan ruang penyimpanan yang besar. Padahal tidak semua data selalu kita gunakan/butuhkan. Karena itu kita dapat melakukan pemilihandata, sehingga data yang sudah jarang kita gunakan dapat kita pindahkan kedalam media penyimpanan *offline*.

5. Kelengkapan (*Completeness*)

Untuk mengkomodasi kebutuhan kelengkapan data yang semakin berkembang, maka kita tidak hanya dapat menambah *record-record* data, tetapi juga dapat melakukan perubahan struktur dalam basis data, baik dalam penambahan objek baru(tabel) atau dengan penambahan *field-field* baru pada suatu tabel.

6. Keamanan (*Security*)

Ada sejumlah sistem pengolahan basis data yang tidak menerapkan aspek keamanan dalam sebuah basis data.

7. Kebersamaan Pemakaian (*Sharebility*)

Pemakai basis data seringkali tidak terbatas pada satu pemakai saja atau di satu lokasi saja oleh satu sistem aplikasi.

2.7 Perangkat Lunak Pendukung

2.7.1 XAMPP

Menurut Riyanto (2010) paket PHP dan MySQL berbasis *Open Source* yang dapat digunakan sebagai *tool* pembantu pengembangan aplikasi berbasis PHP.

2.7.2 PHP

PHP adalah singkatan dari "PHP: *Hypertext Preprocessor*", yang merupakan sebuah bahasa scripting yang terpasang pada HTML. Sebagian besar sintaks mirip dengan bahasa C, Java dan Perl, ditambah beberapa fungsi PHP yang spesifik. Tujuan utama penggunaan bahasa ini adalah untuk memungkinkan perancang web menulis halaman web dinamik dengan cepat (Agus Saputra 2011).

2.7.3 MySQL

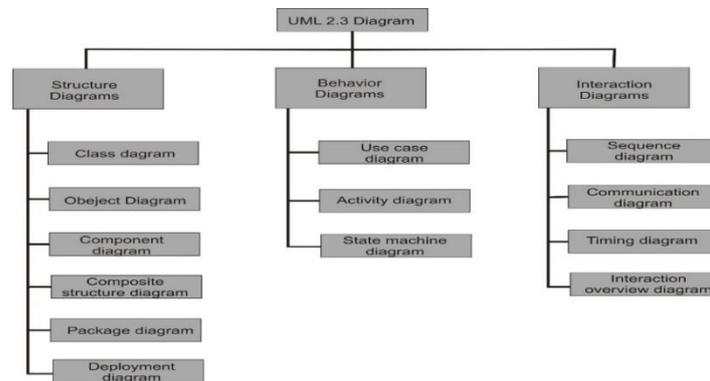
MySQL adalah database yang menghubungkan PHP menggunakan perintah query dan escape character yang sama dengan PHP (Firdaus 2007). Sebagai sebuah program penghasil database, MySQL tidak mungkin berjalan sendiri tanpa adanya aplikasi pengguna (interface) yang berguna sebagai program aplikasi database yang dihasilkan. MySQL dapat didukung oleh hampir semua program aplikasi baik *Open Source* seperti PHP maupun yang tidak *Open Source* yang ada pada platform windows seperti Visual Basic, Delphi dan lainnya.

2.7.4 Diagram UML

Menurut Rosa dan Shalahuddin (2015) pada UML 2.3 terdiri dari 13 macam diagram yang di kelompokkan dalam 3 kategori.

- 1) *Structure diagram* yaitu kumpulan diagram yang di gunakan untuk menggambarkan suatu struktur statis dari sistem yang di modelkan.
- 2) *Behavior diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
- 3) *Interaction diagram* yaitu kumpulan diagram yang di gunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

Lebih jelasnya dapat di lihat pada gambar 2.3 berikut.



Gambar 2.3 Diagram UML

Berikut penjelasan dari jenis-jenis diagram tersebut antara lain:

1. *Class Diagram*

Menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan di buat untuk membangun sistem

2. *Object Diagram*

Menggambarkan struktur sistem dari segi penanaman objek dan jalannya objek dalam sistem.

3. *Component Diagram*

Menunjukkan organisasi dan ketergantungan di antara kumpulan komponen dalam sebuah sistem.

4. *Composite Structure Diagram*

Menggambarkan struktur dari bagian – bagian yang saling terhubung maupun mendeskripsikan struktur pada saat berjalan (*runtime*) dari *instance* yang saling terhubung.

5. *Package Diagram*

Menyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam diagram uml.

6. *Deployment Diagram*

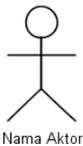
Menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi.

7. *Use Case Diagram*

Pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan di buat dapat di lihat pada tabel

Tabel 2.1. Simbol – simbol *use case diagram*

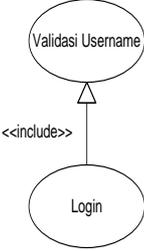
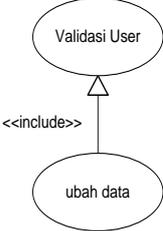
(sumber : buku rekayasa perangkat lunak terstruktur dan berorientasi objek,2015)

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

Tabel 2.1. Lanjutan simbol-simbol *use case* diagram

<p>Ekstensi / extend</p> <p>«extends»</p> 	<p>Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya use case tambahan memiliki nama depan yang sama dengan use case yang sama dengan use case yang ditambahkan, misalnya :</p> <p>arah panah mengarah pada use case</p>
<p>Generalisasi/ <i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi user yang lebih umum dari lainnya, misalnya:</p> <p>Arah panah mengarah pada use case yang menjadi generalisasinya (umum).</p>
<p>Menggunakan <i>/include/uses</i></p> <p>--<<include>>--></p> <p>--<<uses>>--></p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan pada <i>use case</i> ini.</p> <p>Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>:</p> <ul style="list-style-type: none"> • <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selaludipanggil saat <i>use case</i> tambahandijalankan, missal pada kasus berikut:

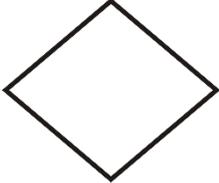
Tabel 2.1. Lanjutan simbol-simbol *use case* diagram

	<ul style="list-style-type: none">  <pre> graph BT Login((Login)) -- <<include>> --> ValidasiUsername((Validasi Username)) </pre> <p><i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, missal pada kasus berikut:</p>  <pre> graph BT ubahData((ubah data)) -- <<include>> --> ValidasiUser((Validasi User)) </pre> <p>Kedua interpretasi diatas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>
--	--

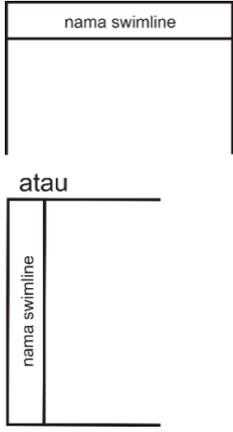
8. Activity Diagram

Menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak dapat di lihat pada tabel 2.2 berikut

Tabel 2.2 Simbol-Simbol *Activity Diagram*
 (sumber : buku rekayasa perangkat lunak terstruktur dan berorientasi objek,2015)

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

Tabel 2.2. Lanjutan Simbol-simbol *activity diagram*

Swimlane	
	<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi</p>

9. *State Machine*Diagram

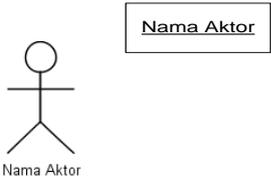
Di gunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek.

10. *Sequence* Diagram

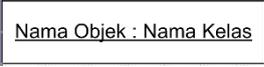
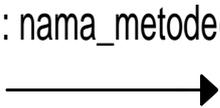
Menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek dapat di lihat pada tabel 2.3

Tabel 2.3 Simbol – simbol *sequence diagram*

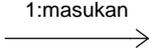
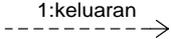
(sumber : buku rekayasa perangkat lunak terstruktur dan berorientasi objek,2015)

Simbol	Deskripsi
<p>Aktor</p> <p>atau</p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.</p>

Tabel 2.3. Lanjutan Simbol-simbol *sequence* diagram

<p>Garishidup / lifeline</p> 	<p>Menyatakan kehidupan suatu objek.</p>
<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktuaktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.</p>
<p>Pesantipe create</p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
<p>Pesantipe call</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>  <p>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.</p>

Tabel 2.3. Lanjutan Simbol-simbol *sequence* diagram

<p>Pesantipe send</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
<p>Pesantipe return</p> 	<p>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.</p>
<p>Pesantipe destroy</p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada <i>destroy</i>.</p>

11. *Communication* Diagram

Menggambarkan interaksi antar objek /bagian dalam bentuk urutan pengiriman pesan

12. *Timing* Diagram

Fokus dalam penggambaran terkait batasan waktu

13. *Interaction Overview* Diagram
Menggambarkan sekumpulan urutan aktivitas