

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Teori Umum

Adapun teori umum yang dijelaskan pada penelitian ini yaitu sebagai berikut:

2.1.1. Sistem

Sistem berasal dari bahasa latin (*systema*) dan bahasa Yunani (*sustema*) adalah suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan. Menurut (Husda, 2012:111) pengertian sistem dari beberapa ahli sebagai berikut:

Menurut L. James Havery; Sistem adalah prosedur logis dan rasional untuk merancang suatu rangkaian komponen yang berhubungan satu dengan yang lainnya dengan maksud untuk berfungsi sebagai suatu kesatuan dalam usaha mencapai suatu tujuan yang telah ditentukan.

Menurut Raymond Mcleod; Sistem adalah himpunan dari unsur-unsur yang saling berkaitan sehingga membentuk suatu kesatuan yang utuh dan terpadu.

Menurut Gordon B. Davis; Sebuah sistem terdiri dari bagian-bagian yang saling berkaitan yang beroperasi bersama untuk mencapai beberapa sasaran atau maksud.

Menurut McLeod yang dipaparkan oleh Yakub sistem adalah sekelompok elemen elemen yang terintegrasi dengan tujuan yang sama untuk mencapai tujuan (Hendrianto, 2014:58).

Menurut Mulyadi sistem adalah sekelompok unsur yang erat berhubungan satu dengan lainnya, yang berfungsi bersama-sama untuk mencapai tujuan tertentu. Menurut Mathiassen dan Lars sistem adalah kumpulan dari komponen-komponen peralatan model *requirements*, *function* dan *interface* (Syakur, 2014:30).

Berdasarkan pendapat tersebut, maka sistem adalah kumpulan dari elemen – elemen atau komponen-komponen peralatan model *requirements*, *function* dan *interface* yang terintegrasi yang berfungsi bersama-sama dengan tujuan yang sama untuk mencapai tujuan tertentu.

2.1.1.1. Karakteristik Sistem

Menurut (Husda, 2012:112-115) karakteristik atau sifat-sifat sitem adalah sebagai berikut:

1. Komponen sistem (*Component System*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk suatu kesatuan. Suatu sistem dapat mempunyai sistem yang lebih besar yang disebut dengan *supra* sistem.

2. Batasan sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya.

3. Lingkungan luar sistem (*Environment*)

Segala sesuatu diluar dari batas sistem yang mempengaruhi operasi dari suatu sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan atau merugikan.

4. Penghubung sistem (*Interface*)

Media penghubung antara subsistem dengan subsistem yang lainnya disebut dengan penghubung sistem atau *interface*.

5. Masukan Sistem (*Input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa:

- a. Masukan perawatan (*Maintenance Input*) adalah energy yang dimasukkan supaya sistem tersebut dapat beroperasi.
- b. Masukan Sinyal (*Signal Input*) adalah energi yang diproses ntuk mendapatkan keluaran.

6. Keluaran Sistem (*Output*)

Merupakan hasil dari energi yang diolah oleh sistem.

7. Pengolah Sistem (*Process*)

Bagian yang memproses masukan untuk menjadi keluaran yang diinginkan.

8. Tujuan Sistem (*Goal*)

Setiap sistem pasti mempunyai tujuan ataupun sasaran yang mempengaruhi *input* yang dibuuhkan dan *output* yang dihasilkan.

2.1.1.2. Klasifikasi Sistem

Menurut (Sutabri, 2012:15), sistem dapat diklasifikasikan sebagai berikut:

1. Sistem abstrack dan sistem fisik

Sistem abstrak merupakan sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Misalnya sistem *teologia* yaitu sistem yang berupa pemikiran tentang hubungan manusia dengan Tuhan. Sedangkan Sistem fisik merupakan sistem yang ada secara fisik sehingga setiap makhluk dapat melihatnya seperti sistem komputer, sistem penjualan, sistem administrasi personalia, dan lain sebagainya.

2. Sistem alamiah dan sistem buatan manusia

Sistem alamiah merupakan sistem yang terjadi karena melalui proses alam, tidak dibuat oleh manusia. Misalnya sistem perputaran bumi, terjadinya siang malam, dan pergantian musim. Sedangkan Sistem buatan manusia merupakan sistem yang melibatkan hubungan manusia dengan mesin yang disebut dengan *human machine system*. Sistem informasi berbasis komputer merupakan contohnya, karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

3. Sistem deterministik dan sistem probabilistik

Sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi disebut sitem deterministik. Sistem komputer adalah contoh dari sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang dijalankan. Sedangkan sistem Probalistik adalah sistem yang

kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

4. Sistem tertutup dan sistem terbuka

Sistem tertutup adalah sistem yang tidak berhubungan dan tidak terpengaruh terhadap sistem luarnya. Sedangkan Sistem yang terbuka adalah sistem yang berhubungan dan terpengaruhi oleh lingkungan luarnya yang menerima masukan dan menghasilkan keluaran untuk subsist lainnya.

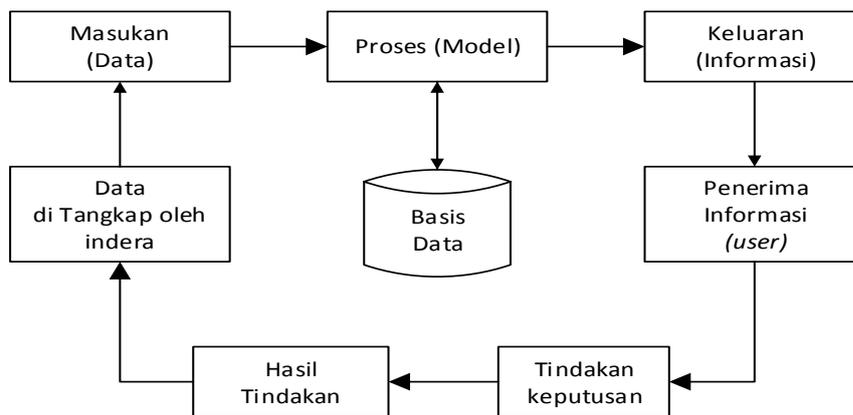
2.1.2. Informasi

Menurut Edhy Sutanta Informasi merupakan hasil pengolahan data sehingga menjadi bentuk yang penting bagi penerimanya dan mempunyai kegunaan sebagai dasar dalam pengambilan keputusan yang dapat dirasakan akibatnya secara langsung saat itu juga atau secara tidak langsung pada saat mendatang (Hendrianto, 2014:58).

Menurut McLeod dalam buku Yakub (2012) yang berjudul “pengantar sistem informasi” menjelaskan bahwa Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya (Hendrianto, 2014:58).

Menurut Jogiyanto HM., Informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian (*event*) yang nyata (*fact*) yang digunakan untuk pengambilan keputusan (Syakur, 2014:30).

Menurut (Husda, 2012:117-118) secara umum informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi si penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan. Berikut merupakan gambaran dari siklus informasi:



Sumber: (Husda, 2012:118)

Gambar 2.1 Siklus Informasi

Menurut (Husda, 2012:118) Informasi yang berkualitas adalah sebagai berikut:

1. Akurat

Informasi harus bebas dari kesalahan-kesalahan dan tidak bisa atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya.

2. Tepat waktu

Informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak mempunyai nilai lagi karena informasi merupakan landasan dalam pengambilan keputusan.

3. Relevan

Berarti Informasi tersebut mempunyai manfaat untuk pemakainya.

Relevansi informasi untuk tiap-tiap orang berbeda-beda.

Berdasarkan pendapat-pendapat para ahli tersebut maka, informasi adalah data yang diolah menjadi bentuk yang lebih berguna bagi penerimanya yang menggambarkan suatu kejadian-kejadian nyata dan digunakan dalam pengambilan keputusan serta dapat dirasakan akibatnya secara langsung saat itu juga atau tidak langsung pada saat mendatang.

2.1.3. Sistem Informasi

Sistem informasi merupakan kumpulan komponen yang memastikan pengolahan data untuk tujuan memperoleh informasi seluruh kegiatan pengumpulan, manipulasi, proses, penyimpanan, organisasi dan distribusi dalam rangka untuk memastikan pendukung keputusan untuk mencapai tujuan organisasi (Georgescu & Jeflea, 2015:269).

Menurut Hidayat, Taufiq, dan sugiarto, sistem informasi adalah kumpulan dari sub-sub sistem yang saling terintegrasi dan berkolaborasi untuk menyelesaikan masalah tertentu dengan cara mengolah data dengan alat yang namanya komputer sehingga memiliki nilai tambah dan bermanfaat bagi pengguna (Erwanto & Monalisa, 2016:74).

Menurut Abdul Kadir Sistem Informasi adalah sejumlah komponen (manusia, komputer, teknologi, *informasi*, dan prosedur kerja), ada suatu sasaran atau tujuan (Septavia, Gunadhi, & Kurniawati, 2015:2). Setelah menguraikan

definisi dari sistem informasi maka secara umum sistem informasi dapat diartikan sebagai kumpulan dari berbagai sub sistem, baik lainnya yang berjalan secara harmonis untuk mencapai tujuan yang sama yaitu mengumpulkan dan mengelola data menjadi suatu informasi yang bermanfaat dan memiliki sebuah nilai, komponen-komponen sistem informasi menurut Jogiyanto terdiri dari:

1. *Hardware* yaitu kumpulan dari perangkat lunak yang terlihat memungkinkan dapat membentuk sistem seperti komputer, printer dan jaringan.
2. *Software* adalah kumpulan dari perintah-perintah atau fungsi yang ditulis dengan aturan tertentu, memerintahkan agar melaksanakan fungsi tertentu.
3. Data merupakan bahan dasar dari suatu informasi berupa fakta yang mengangkat kejadian- kejadian nyata dan dituangkan kedalam suatu symbol.
4. Prosedur adalah suatu tahapan yang berupa urutan kegiatan yang saling berhubungan untuk mencapai tujuan yang berupa suatu dokumen prosedur seperti : buku petunjuk operasional dan teknis.
5. Manusia merupakan pelaksana dari suatu sistem informasi seperti: Operator, *Programmer, Analyst, Designer* dan sebagainya.

Menurut (Septavia et al., 2015) Sistem informasi memiliki komponen-komponen sebagai pendukungnya, komponen-komponen tersebut diantaranya adalah sebagai berikut:

1. Blok masukan yaitu *input* yang memiliki data yang masuk ke dalam sistem informasi termasuk metode dan media untuk mendapatkan data yang akan dimasukan yang berupa dokumen-dokumen dasar.
2. Blok model yaitu kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di bais data untuk menghasilkan keluaran yang diinginkan.
3. Blok keluaran yaitu informasi yang berkualitas dan dokumentasi yang berguna untuk semua pemakai sistem dan manajemen.
4. Blok teknologi yaitu alat yang digunakan untuk menerima *input*, menyimpan dan mengakses data serta menghasilkan keluaran yang diinginkan.
5. Blok basis data yaitu basis data yang digunakan dan disimpan untuk keperluan penyediaan informasi lebih lanjut dimana berisi data-data yang diorganisasikan sedemikian rupa agar informasi yang dihasilkan berkualitas.
6. Blok kendali yaitu pengendalian pada perusahaan untuk mencegah hal-hal yang dapat merusak sistem atau bila ada kesalahan-kesalahan dapat langsung diatasi.

2.1.4. Rancang Bangun

Menurut Pressman kata “rancang” merupakan kata sifat dari “perancangan” yakni merupakan serangkaian prosedur untuk menerjemahkan hasil analisis dari sebuah sistem kedalam bahasa pemrograman untuk mendeskripsikan dengan

detail bagaimana komponen-komponen sistem diimplementasikan. Kata “bangun” merupakan kata sifat dari “pembangunan” adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun sebagian (Zulfiandri, Hidayatuloh, & Anas, 2014:474).

Dengan demikian pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut ataupun memperbaiki sistem yang sudah ada.

2.1.5. Perancangan Sistem

Perancangan sistem informasi merupakan pengembangan sistem baru dari sistem lama yang ada, dimana masalah-masalah yang terjadi pada sistem lama diharapkan sudah teratasi pada sistem yang baru (Hasugian & Shidiq, 2012:607).

2.1.6. Basis Data

Menurut (Eko Suharyanto, Eka Chandra, & E Gunawan, 2017:226), Basis data atau *database* adalah kumpulan data terstruktur. Agar dapat menambahkan, mengakses, dan memproses data yang tersimpan dalam database komputer, dibutuhkan sistem manajemen basis data (*database management system*).

Basis Data merupakan kumpulan data (arsip) yang saling berhubungan yang merefleksikan fakta-fakta yang terdapat di organisasi dan disimpan dalam bentuk penyimpanan elektronik (Fauzi Bastaman, 2015:3).

2.1.6.1. Batasan Aturan Basis Data

Menurut James Martin, dalam buku (Sutanta, 2011:53-69) batasan aturan basis data berhubungan dengan lima aspek penting dalam basis data antara lain sebagai berikut:

1. Kerangkapan data (*data redundancy*)

Kerangkapan data (*data redundancy*) adalah munculnya data yang sama secara melimpah (berulang kali) pada *file* basis data yang semestinya tidak diperlukan.

Kejadian kerangkapan data dapat terjadi pada dua kemungkinan yaitu:

- a. Kerangkapan data dalam satu *file*
- b. Kerangkapan data dalam beberapa *file*

2. Inkonsistensi data (*data inconsistency*)

Inkonsistensi data (data inconsistency) atau data tidak konsisten adalah munculnya data yang tidak konsisten pada medan/kolom yang sama dalam satu atau beberapa *file* data yang dihubungkan/direlasikan.

Inkonsistensi data dapat terjadi di akibatkan oleh:

- a. Proses pemasukan data (*data entry*) yang tidak benar
- b. Proses pembaruan data (*update*) yang tidak benar
- c. Pengendalian sistem yang tidak baik/terkontrol

3. Data terisolasi (*data isolation*)

Data terisolasi disebabkan oleh pemakaian beberapa *file* basis data dimana program aplikasi tidak dapat mengakses data-data dari *file* tertentu, kecuali bila

program aplikasi diubah/ditambah sehingga seolah-olah ada *file* yang terpisah/terisolasi terhadap *file* yang lain dalam basis data.

Data terisolasi dapat terjadi akibat:

- a. Tidak adanya kemungkinan untuk menghubungkan antar data dalam *file*
- b. Tidak adanya standarisasi data (berkaitan dengan domain/*format* data, meliputi tipe dan ukuran data)

4. Keamanan data (*data security*)

Keamanan data merupakan aspek kritis dalam basis data. Prinsip dasar dari keamanan data dalam basis data adalah bahwa data-data dalam basis data merupakan sumber informasi yang bersifat sangat penting dan rahasia.

5. Integritas data (*data integrity*)

Integritas data (*data integrity*) berhubungan dengan kinerja sistem agar dapat melakukan kendali/*control* pada semua bagian sistem. Salah satu cara terbaik untuk menjaga integritas data adalah meyakinkan bahwa nilai-nilai data adalah benar sejak pertama kali masuk.

2.2. Tinjauan Teori Khusus

2.2.1. Penyewaan

Pengertian sewa menurut kamus besar bahasa indonesia adalah pemakaian sesuatu dengan membayar uang sewa, uang yang dibayarkan karena memakai atau meminjamkan sesuatu, yang boleh pakai dengan membayar uang dengan uang.

Menurut R. Subekti dan Tjiro Soedibjo penyewaan adalah suatu kesepakatan atau persetujuan dimana pihak yang satu menyanggupkan dirinya untuk menyerahkan suatu kebendaan kepada pihak lain, agar pihak ini dapat menikmatinya dalam jangka waktu tertentu, yang mana pihak yang belakang ini sanggup membayarnya (Septavia et al., 2015:3).

Menurut (Fauzi Bastaman, 2015:2), Penyewaan adalah persetujuan dimana sebuah pembayaran dilakukan atas penggunaan suatu barang atau properti secara sementara oleh orang lain.

Penyewaan berasal dari kata dasar sewa yang mendapat tambahan kata imbuhan pedan akhiran an. Sewa sendiri mempunyai arti yaitu merupakan suatu proses kegiatan pinjam-meminjam, sedangkan penyewaan adalah suatu kegiatan yang melayani jasa peminjaman dengan tidak mengabaikan suatu ketentuan atau kesepakatan dan syarat-syarat yang berlaku didalam organisasi tersebut guna mencapai satu tujuan bersama (Rismayani & SY, 2016:20).

Menurut (Frayoga & Fitriani, 2016:199) sewa dalam kamus besar bahasa Indonesia (departemen pendidikan dan kebudayaan republik Indonesia.2001) adalah pemakaian sesuatu dengan membayar uang sewa, uang yang dibayarkan karena memakai atau meminjamkan sesuatu, yang boleh pakai dengan membayar uang dengan uang. Sedangkan pengertian penyewaan adalah proses, cara, pembuatan menyewa atau menyewakan. Yang dimaksud dengan sewa, yaitu balas jasa atas sewa ruangan dalam keadaan kosong yang dapat ditagih dimuka (pada awal penyewaan) atau dibelakang sesuai dengan kontrak (perjanjian).

Berdasarkan pendapat-pendapat tersebut maka penyewaan adalah suatu kesepakatan atau persetujuan yang dilakukan dengan membayar uang sewa atas penggunaan suatu barang atau properti dalam jangka waktu tertentu.

2.2.2. Futsal

Futsal adalah permainan bola yang dimainkan oleh dua tim, yang masing-masing beranggota lima orang. Tujuannya adalah memasukkan bola ke gawang lawan, dengan memanipulasi bola dengan kaki. Selain lima pemain utama, setiap regu juga diizinkan memiliki pemain cadangan. Tidak seperti permainan sepak bola dalam ruangan lainnya, lapangan futsal dibatasi garis, bukan net atau papan (Rahma, 2015:664).

2.2.3. Internet

Internet adalah singkatan dari *Interconnected Network*. Internet merupakan sebuah sistem komunikasi yang mampu menghubungkan jaringan-jaringan komputer diseluruh dunia (Hendrianto, 2014). Internet dimulai dari jaringan komputer yang dibentuk oleh Departemen Pertahanan Amerika Serikat di tahun 1969, melalui proyek ARPA yang disebut ARPANET (*Advanced Research Project Agency Network*) mendemonstrasikan bagaimana dengan *hardware* dan *software* komputer yang berbasis UNIX, dapat melakukan komunikasi dengan jarak yang tidak terhingga melalui saluran telepon.

Proyek ARPANET merancang bentuk jaringan, kehandalan, seberapa besar informasi dapat dipindahkan, dan akhirnya semua standar ditentukan menjadi cikal bakal pembangunan protocol baru yang sekarang dikenal sebagai TCP/IP

(*Transmission Control Protocol/Internet Protocol*). ARPANET hanya menghubungkan 4 situs saja yaitu *Stanford Research Institute, University of California, Santa Barbara, University of Utah* yang membentuk satu jaringan terpadu di tahun 1969, dan secara umum ARPANET diperkenalkan pada bulan Oktober 1972 (Husda, 2012:162).

Menurut Lani Sidharta, Internet adalah interkoneksi antar jaringan komputer namun secara umum internet harus dipandang sebagai sumber daya informasi. Isi Internet adalah informasi, dapat dibayangkan sebagai *database* atau perpustakaan multimedia yang sangat besar dan lengkap. Bahkan internet dipandang sebagai dunia dalam bentuk lain (maya) karena hampir seluruh aspek kehidupan di dunia nyata ada di internet seperti bisnis, hiburan, olah raga, politik dan lain sebagainya (Hendrianto, 2014:59).

2.2.4. Web

WWW atau dikenal dengan *Web* atau situs adalah sistem dimana informasi dalam bentuk teks, suara, gambar dan lain-lain yang disimpan di *server-server* yang terdapat di seluruh dunia. Dokumen *web* dibuat dengan menggunakan format HTML atau *Hypertext Mark-up Language* (Husda, 2012:163).

Menurut W. B. Anita *Web* dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video, dan atau gabungan dari semuanya baik yang bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan halaman (Jihadi, 2017:8). Menurut

Raharjo *Web* merupakan salah satu layanan internet yang digunakan pada suatu jaringan komputer (Septavia et al., 2015:1).

2.2.5. Website

Menurut Jhonsen, *Website* (Situs *Web*) merupakan kumpulan dari halaman-halaman *web* yang berhubungan dengan *file-file* lain yang terkait. Dalam sebuah *website* terdapat suatu halaman yang dikenal dengan sebutan *home page*. *Home page* adalah sebuah halaman yang pertama kali dilihat ketika seseorang mengunjungi *website*. Dari *home page*, pengunjung dapat mengklik *hyperlink* untuk pindah kehalaman lain yang terdapat dalam *website* tersebut (Hendrianto, 2014:59).

2.2.6. PHP

Bahasa pemrograman populer untuk pengembangan situs *web* adalah PHP, *Ruby*, *Python* dan *Java*. PHP dipilih untuk analisis kerangka dan pengembangan situs *web*, karena merupakan sumber bahasa *scripting* terbuka dengan banyak keuntungan dukungan untuk jenis *database* yang berbeda, banyak struktur bahasa diambil dari C dan *Perl* bahasa. Sejak PHP menjadi bahasa populer dan banyak digunakan, PHP memiliki banyak dokumentasi di *Web* (Prokofyeva & Boltunova, 2017:51-56).

Menurut Bimo sunarfrihantono, ST.,PHP adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis. Maksud dari *server-side scripting* adalah sintaks dan perintah-perintah yang diberikan akan sepenuhnya akan dijalankan diserver tetapi disertakan pada

dokumen HTML. Pembuatan *web* ini merupakan kombinasi antara php sendiri sebagai bahasa pemrograman dan HTML sebagai pembangun halaman *web* (Kristiawan & Uly Wardati, 2015:8).

PHP memiliki 2 macam method yang bisa digunakan untuk mengirim data dari *client (browser)* ke server, yaitu *POST* dan *GET*. Method *POST* digunakan untuk mengirim data dari *client* ke server dimana data tersebut akan dikirim melalui http header.

Sebagai contoh kita membuat suatu *form input* data dengan nama *formInput.php* yang isinya sebagai berikut:

```
<html>
<head>
<title> Form Input </title>
</head>
<body>
<form method="POST" action="tInput.php">
Judul : <input type="text" name="judul"> <br>
Penulis : <input type="text" name="penulis"> <br>
Penerbit : <input type="text" name="penerbit"> <br>
<input type="submit" value="submit">
</form>
</body>
</html>
```

Method *GET*, yang bekerja dengan perintah fungsi *\$GET* akan mengirim data atau berkomunikasi melalui parameter yang akan terlihat pada *URL* (Saputra, 2012:93).

2.2.7. HTML

Menurut Bimo Sunarfrihantono, ST., HTML (*Hypertext Markup Language*) merupakan salah satu *format* yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan di halaman *web* (Kristiawan & Uly Wardati, 2015:8).

HTML bisa disebut bahasa paling dasar dan penting yang digunakan untuk menampilkan dan mengelola tampilan pada halaman *website*. HTML menggunakan ekstensi *file* yaitu **.htm** dan **.html**. *Format* Ekstensi berformat **.htm** awalnya hanya untuk mengakomodasi pengguna html dalam proses DOS (Saputra, 2012:1).

Beberapa hal yang bisa dilakukan dengan HTML yaitu (Firstiara Maudi, Laila Nugraha, & Sasmito, 2014:101):

1. Mengontrol tampilan dari *Web page* dan contentnya.
2. Mempublikasikan dokumen secara *online* sehingga bisa di akses dari seluruh dunia.
3. Membuat *online form* yang bisa di gunakan untuk menangani pendaftaran,transaksi secara *online*.
4. Menambahkan objek-objek seperti image, audio, video dan juga java applet dalam dokumen HTML.

Ciri-ciri HTML adalah sebagai berikut (Firstiara Maudi et al., 2014:102):

1. Tersusun oleh *tag-tag* seperti `<html>.....</html>`
2. Pada umumnya tag selalu memiliki tag pembuka dan kemudian ada tag penutupnya.
3. Tidak *case sensitive*, artinya huruf kapital maupun bukan huruf kapital akan dianggap sama.
4. Nama *file* berupa *.html atau *.htm.

Dalam buku (Saputra, 2012:5) Dokumen html memiliki sebuah struktur yang harus diikuti aturan pembuatannya. Untuk membangun suatu pondasi kerangka *website* ada beberapa elemen-elemen wajib pada *file* html, salah satunya adalah Elemen *Body*. Elemen *Body* merupakan bagian utama dalam dokumen *web*. Untuk menampilkan suatu teks atau informasi atau ang dikenal dengan sebtan konten, maka harus meletakkan teks tersebut pada elemen *body*. Struktur elemennya sebagai berikut:

```
<html>
<head>
    <title> Tuliskan Judul Disini </title>
</head>
<body>
    Tuliskan Konten Disini
</body>
</html>
```

2.2.8. CSS

Menurut Kadir, CSS merupakan singkatan dari *Cascading Style Sheet*. CSS biasa digunakan dalam dokumen HTML untuk menciptakan suatu *style* yang dipakai untuk mengatur penampilan elemen HTML. Dengan menggunakan *style*, suatu elemen dapat diformat dengan fitur yang jauh lebih kaya daripada yang disediakan oleh elemen HTML itu sendiri. Sebagai contoh, pengaturan seperti warna tulisan bisa ditangani melalui *style* tanpa melibatkan *tag* HTML yang berfungsi untuk mengatur warna (Firstiara Maudi et al., 2014:103).

CSS saat ini sudah mencapai versi 3 dimana pada setiap versi pasti ada peningkatan yang dilakukan. Menurut (Saputra, 2012:28) terdapat peningkatan tiap versi CSS dari versi 1 sampai versi 3 yaitu:

1. *CSS1*, masih kuno, CSS hanya dikembangkan dan digunakan untuk formatting dokumen html.
2. *CSS2*, disini sudah mulai menggunakan *font*, *table-layot*, dan berbagai media printer.
3. *CSS3*, merupakan pengembangan dari versi CSS sebelumnya. Peningkatan yang mencolok pada versi ini adalah peningkatan fitur yang mengarah pada efek animasi.

2.2.9. Dreamweaver

Adobe Dreamweaver merupakan program penyunting halaman *web* keluaran *Adobe Systems* yang dulu dikenal sebagai *Macromedia Dreamweaver* keluaran *Macromedia*. Program ini banyak digunakan oleh pengembang *web*

karena fitur-fiturnya yang menarik dan kemudahan penggunaannya. Versi terakhir *Macromedia Dreamweaver* sebelum *Macromedia* dibeli oleh *Adobe Systems* yaitu versi 8. Versi terakhir Dreamweaver keluaran Adobe Systems adalah versi 10 yang ada dalam Adobe Creative Suite 4 sering disingkat Adobe CS4 (Firstiara Maudi et al., 2014:102).

Menurut P, Adhi *Dreamweaver* adalah sebuah tools untuk membantu kita menuliskan kode HTML secara visual. Menurut Milician *Dreamweaver is a powerful Hyper Text Markup Language (HTML) editor used by professionals, as well as beginners. Dreamweaver* adalah *Hyper Text Markup Language (HTML)* editor yang digunakan oleh professional, serta pemula (Jihadi, 2017:7-8).



Sumber:(Andi, 2013:2).

Gambar 2.2 Tampilan Awal *Dw Adobe Dreamweaver CS6*

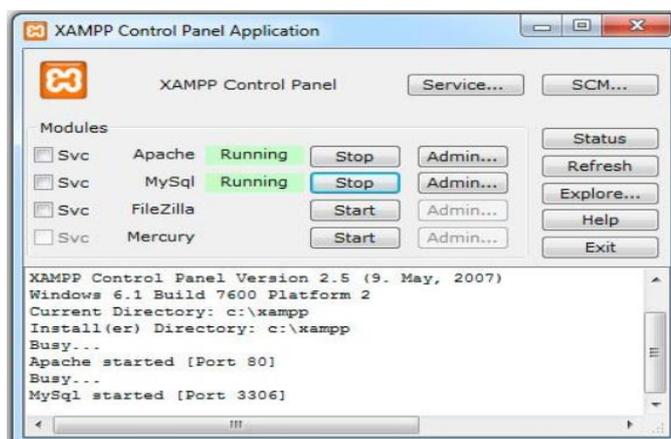
2.2.10. Javascript

JavaScript merupakan bahasa pemrograman *web client side*. Kalau HTML digunakan untuk membuat halaman *web* statis, maka *JavaScript* digunakan untuk

membuat halaman *web* yang interaktif dan dinamis. Karena sebagai bahasa pemrograman, *JavaScript* dapat digunakan untuk membuat aplikasi matematis, efek animasi sederhana, bahkan juga untuk membuat *game*. Hampir *browser* yang ada saat ini sudah *support JavaScript*. Dokumen *JavaScript* dapat dibuat dengan *text editor* biasa, seperti: Notepad, Wordpad, Notepad++, dll, yaitu dengan menyimpannya kedalam format *.js (Firstiara Maudi et al., 2014:103).

2.2.11. XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsi dari XAMPP adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama XAMPP merupakan singkatan dari X (sistem operasi apapun), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia dalam *GNU General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis.



Sumber: (Triyanto, Ruslianto, & Fadly, 2015)

Gambar 2.3 Tampilan XAMPP Control Panel

2.2.12. MySQL

MySQL adalah *database open source* terpopuler di dunia. Dengan kinerja, kehandalan dan kemudahan penggunaan yang terbukti, *MySQL* telah menjadi pilihan *database* terdepan untuk aplikasi berbasis *web*, yang digunakan oleh properti *web* profil tinggi termasuk *Facebook*, *Twitter*, *YouTube*, *Yahoo!* dan banyak lagi.

Kata “*SQL*” dari “*MySQL*” adalah singkatan dari “*Structured Query Language*”. *SQL* adalah bahasa standar yang paling umum digunakan untuk mengakses *database*. Bergantung pada lingkungan pemrograman yang kita pakai, kita bisa memasukkan *SQL* secara langsung (misalnya, untuk men-*generate* laporan), memasukkan pernyataan *SQL* ke dalam kode yang ditulis dalam bahasa lain, atau menggunakan API khusus yang dapat menyembunyikan sintaks *SQL* (Eko Suharyanto et al., 2017:226).

2.2.13. SDLC

SDLC dimulai dari tahun 1960-an, untuk mengembangkan sistem skala usaha besar secara fungsional untuk konglomerat pada jaman itu. SDLC singkatan dari *Software Development Life Cycle* atau kadang disebut juga *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya yang berdasarkan *best practice* atau cara-cara yang sudah teruji baik (Rossa & Shalahuddin, 2013:24-26).

Tahapan-tahapan yang ada pada SDLC secara global adalah sebagai berikut (Rossa & Shalahuddin, 2013:24-26):

1. Inisiasi (*Initiation*)

Tahapan yang ditandai dengan pembuatan proposal proyek perangkat lunak.

2. Pengembangan konsep sistem (*System concept development*)

Mendefinisikan lingkup konsep termasuk dokumen lingkup sistem, analisis manfaat biaya, manajemen rencana, dan pembelajaran kemudahan sistem.

3. Perencanaan (*planning*)

Mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya.

4. Analisis Kebutuhan (*requirements analysis*)

Menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *user*. Membuat dokumen kebutuhan fungsional.

5. Desain (*design*)

Mentransformasikan kebutuhan detail menjadi kebutuhan yang sudah lengkap, dokumen desain sistem focus pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan.

6. Pengembangan (*development*)

Mengonversi desain ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang

dibutuhkan; membuat basis data dan mempersiapkan prosedur kasus pengujian; mempersiapkan berkas atau *file* pengujian, pengodean, pengompilasian, memperbaiki dan membersihkan program; peninjauan pengujian.

7. Integrasi dan pengujian (*integration and test*)

Mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional. Dengan diarahkan oleh staf penjamin kualitas (*quality assurance*) dan *user*. Menghasilkan laporan analisis pengujian.

8. Implementasi (*implementation*)

Persiapan implementasi, implementasi perangkat lunak pada lingkungan produksi (lingkungan pada *user*) dan menjalankan resolusi dari permasalahan yang teridentifikasi dari fase intergrasi dan pengujian.

9. Operasi dan pemeliharaan (*operations and maintenance*)

10. Mendeskripsikan pekerjaan

Mendeskripsikan pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan produksi (lingkungan pada *user*), termasuk implementasi akhir dan masuk pada proses peninjauan.

11. Disposisi (*disposition*)

Mendeskripsikan aktifitas akhir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktifitas *user*.

SDLC memiliki beberapa model dalam penerapan prosesnya antara lain: model *Waterfall*, Model Prototipe, Model Rapid Application Development (RAD),

Model Interatif, Model Spiral. Dalam penelitian ini akan membahas salah satu model tersebut yaitu Model *Waterfall* (Rossa & Shalahuddin, 2013:26-37).

2.2.13.1. Model Waterfall

Menurut Pressman *waterfall* Model adalah model klasik yang bersifat sistematis, berurutan dalam membangun software. Nama model ini sebenarnya adalah “*Linear Sequential Model*”. Model ini sering disebut dengan “*Classic Life Cycle*” atau *waterfall model*. Model ini termasuk ke dalam model *generic* pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai didalam *Software Engineering* (SE). Model ini melakukan pendekatan secara sistematis dan berurutan. Model ini disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. *Waterfall Model* memiliki enam tahapan, yaitu *system engineering*, *requirements analysis*, *design*, *coding*, *testing* dan *maintenance* (Agustia, Suci, Inayati, & Darujati, 2017).

2.2.14. UML

UML adalah bahasa pemodelan tujuan standar umum di bidang ilmu komputer dan rekayasa perangkat lunak. UML bertujuan untuk menjadi bahasa pemodelan standar yang model sistem konkuren dan sistem distribusi. UML adalah *de facto* standar industri yang berkembang di bawah naungan *Object Management Group* (OMG). UML menggabungkan teknik terbaik dari

pemodelan data (diagram hubungan entitas), pemodelan bisnis (aliran kerja), pemodelan objek, dan pemodelan komponen. Hal ini dapat digunakan dengan semua proses, sepanjang hidup pengembangan perangkat lunak (Siddique, Iaeng, & Main, 2010:264).

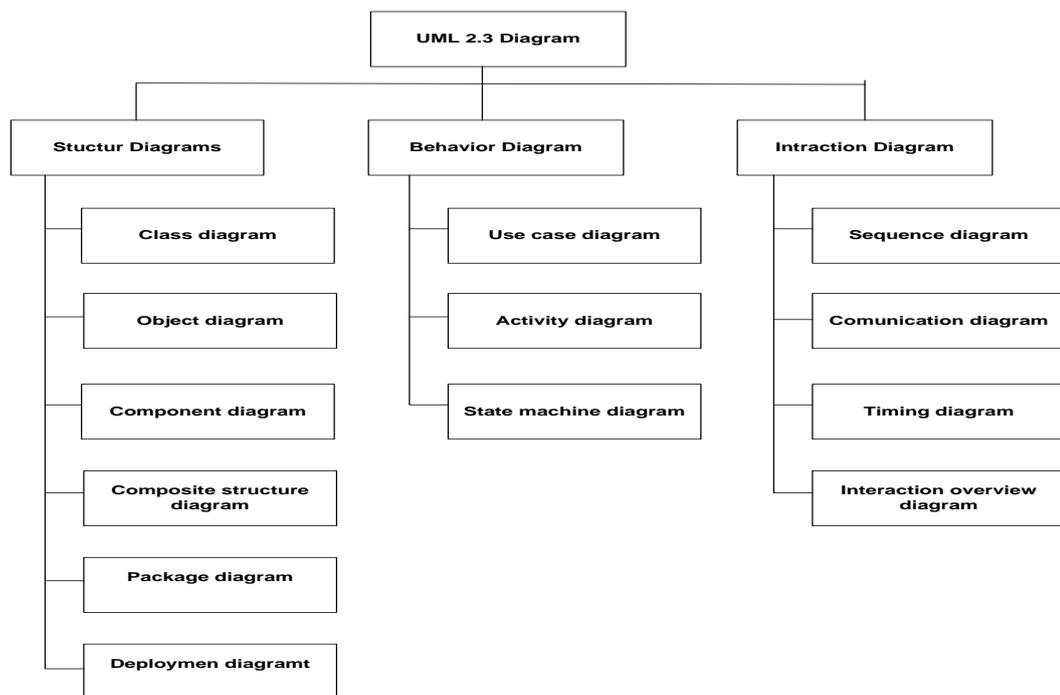
Menurut Amanda UML adalah bahasa yang digunakan untuk memvisualisasikan, mendefinisikan, membangun dan membuat dokumen dari arsitektur perangkat lunak. UML dapat digunakan pada semua proses melalui metodologi pengembangan perangkat lunak dan melakukan implementasinya pada teknologi yang berbeda (Agustia et al., 2017).

Menurut Ginting Unified *Modelling Language (UML)* bukanlah suatu proses melainkan bahasa pemodelan secara grafis untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan seluruh artifak sistem perangkat lunak. Penggunaan model ini bertujuan untuk mengidentifikasi bagian-bagian yang termasuk dalam lingkup sistem yang dibahas dan bagaimana hubungan antara sistem dengan sub sistem maupun sistem lain diluarnya (Agustia et al., 2017).

Secara fisik, UML adalah sekumpulan spesifikasi yang dikeluarkan oleh OMG. UML terbaru adalah UML 2.3 yang terdiri dari 4 macam spesifikasi, yaitu *Diagram Interchange Specification*, UML Infrastructure, UML Superstructure, dan Object Constaint Language (Rossa & Shalahuddin, 2013:140).

2.2.15. Diagram UML

Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan menjadi 3 kategori. Pembagian kategori dapat dilihat pada gambar dibawah ini:



Sumber: (Rosa & Shalahuddin, 2013)

Gambar 2.4 Diagram UML

Berikut ini penjelasan singkat dari pembagian kategori tersebut:

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

Dari 13 diagram tersebut dalam melakukan rancang bangun sistem informasi penyewaan lapangan futsal berbasis *web* penulis menggunakan *Class diagram*, *Use Case diagram*, *Activity diagram*, dan *Sequence Diagram* berikut penjelasan dari masing-masing diagram:

2.2.15.1. Class Diagram

Diagram kelas (*class diagram*) adalah suatu diagram yang melukiskan kelas yang sesuai dengan komponen- komponen perangkat lunak yang digunakan untuk membangun aplikasi perangkat lunak (Hasugian & Shidiq, 2012:609). Diagram class dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron (Rossa & Shalahuddin, 2013:141-147).

Berikut simbol-simbol yang ada pada diagram kelas:

Tabel 2.1 Simbol Class Diagram

Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur system
<p>Antarmuka / <i>Interface</i></p> 	Sama dengan konsep <i>interfaced</i> dalam pemograman berorientasi objek
<p>Asosiasi / <i>association</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>

Lanjutan Tabel 2.1

Simbol	Deskripsi
Asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan / <i>dependency</i> 	Relasi antara kelas dengan makna kebergantungan antar kelas
Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whol-part</i>)

Sumber: (Rossa & Shalahuddin, 2013:146).

2.2.15.2. Usecase Diagram

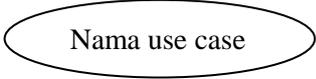
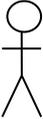
Menurut Suhendar dan Hariman Gunadi *Usecase diagram* adalah rangkaian/uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. *Usecase* digunakan untuk membentuk tingkah-laku benda/ *things* dalam sebuah model serta direalisasikan oleh sebuah *collaboration* (Rismayani & SY, 2016:21). Umumnya *usecase* digambarkan dengan sebuah *elips* dengan garis yang *solid*, biasanya mengandung

nama. *Usecase* menggambarkan proses sistem (kebutuhan sistem dari sudut pandang *user*).

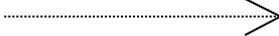
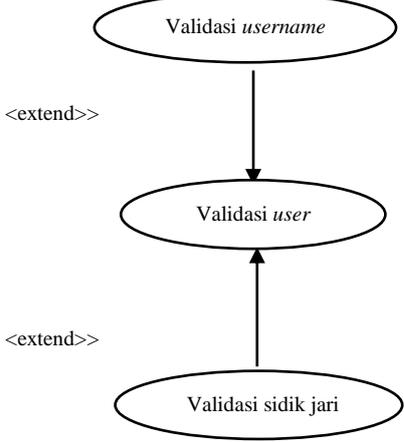
Syarat penamaan pada *usecase* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *usecase* yaitu aktor dan *usecase* (Rossa & Shalahuddin, 2013:165-166).

- a. Aktor merupakan orang, proses, atau sistem lain berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau *actor*. Berikut simbol-simbol yang ada pada diagram *Usecase Diagram*.

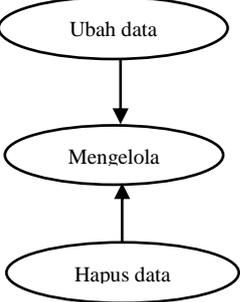
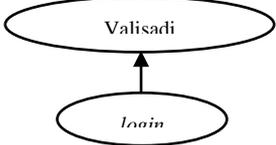
Tabel 2.2 Simbol-Simbol Usecase Diagram

No	Simbol	Deskripsi
1.	<p><i>Use case</i></p>  <p>Nama use case</p>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama <i>use case</i>
2.	<p>Aktor/ <i>actor</i></p>  <p>Nama aktor</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; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor

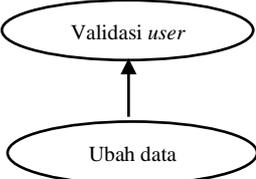
Lanjutan Tabel 2.2

No.	<i>Simbol</i>	<i>Deskripsi</i>
3.	Asosiasi / <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
4.	Ekstensi / <i>extend</i> << <i>extend</i> >> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan  arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i> -nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.

Lanjutan Tabel 2.2

No	Simbol	Deskripsi
5.	<i>Generalisasi / generalization</i> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p>  <p>arah panah mengarah pada <i>use case</i> yang menjadi generalisasi (umum).</p>
6.	Menggunakan / include/ uses	<p>Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>usecase</i> ini</p>
7.	<p><<extend>></p>  <p><<uses>></p> 	<p>Ada dua sudut pandang yang cukup besar mengenai include di use case:</p> <p>1) Include berarti use case yang ditambahkan akan selalu dipanggil saat use case tambahan dijalankan, misal pada kasus berikut:</p>  <p>2) Include berarti use case tambahan akan selalu melakukan pengecekan apakah use case yang ditambahkan telah</p>

Lanjutan Tabel 2.2

No	Simbol	Deskripsi
		<p>dijalankan sebelum use case tambahan dijalankan, misal pada kasus berikut:</p>  <pre> graph BT A([Ubah data]) --> B([Validasi user]) </pre> <p>Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

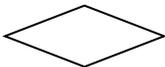
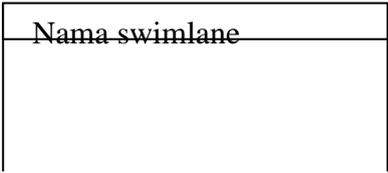
Sumber: (Rosa & Shalahuddin, 2013:157-158)

2.2.15.3. Activity Diagram

Activity Diagram adalah representasi grafis dari alur kerja kegiatan bertahap dan tindakan dengan dukungan untuk pilihan, *iterasi* dan *concurrency*. Dalam *Unified Modeling Language*, aktivitas diagram dapat digunakan untuk menggambarkan bisnis dan alur kerja langkah-demi-langkah operasional komponen dalam suatu sistem. Diagram aktivitas menunjukkan aliran keseluruhan kontrol (Siddique et al., 2010:264).

Pada *Activity diagram* hal yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas diagram bisa dilakukan oleh sistem (Rossa & Shalahuddin, 2013:161-163).

Tabel 2.2 Simbol-Simbol *Activity Diagram*

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

Sumber: (Rosa & Shalahuddin, 2013)

2.2.15.4. *Sequence Diagram*

Sequence diagram (diagram urutan) adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek

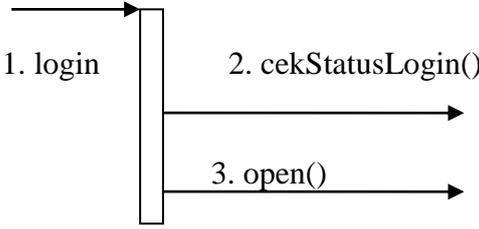
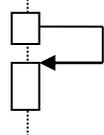
tersebut termasuk pengguna, *display*, dan sebagainya berupa pesan/*message*. *Sequence Diagram* digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk menghasilkan output tertentu (Sy & Rismayani, 2016:21).

Berikut adalah symbol-simbol yang ada pada *sequen diagram* (A.S & Shalahuddin, 2013:165):

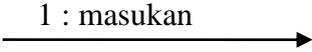
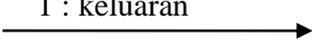
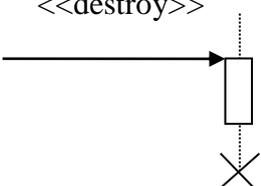
Tabel 2.3 Simbol-Simbol *Sequence Diagram*

Simbol	Deskripsi
<p data-bbox="325 875 405 902">Aktor</p> <div data-bbox="491 1099 539 1223" style="text-align: center;">  </div> <p data-bbox="437 1391 592 1417" style="text-align: center;">nama_aktor</p> <p data-bbox="325 1462 384 1489">atau</p> <div data-bbox="418 1547 619 1653" style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block;"> <u>Nama aktor</u> </div> </div> <p data-bbox="325 1756 552 1783">tanpa waktu aktif</p>	<p data-bbox="845 875 1342 1346">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor</p>

Lanjutan Tabel 2.3

Simbol	Deskripsi
Garis hidup / <i>lifeline</i>	Menyatakan kehidupan suatu objek
Objek <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <u>Nama objek : nama kelas</u> </div>	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya  <p>Maka cekStatusLogin() dan open() dilakukan didalam metode login() Aktor tidak memiliki waktu aktif</p>
Pesan tipe create <<create>> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarahkan pada objek yang dibuat
Pesan tipe call 1 : nama_metode() 	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.  <p>1 : nama_metode()</p>

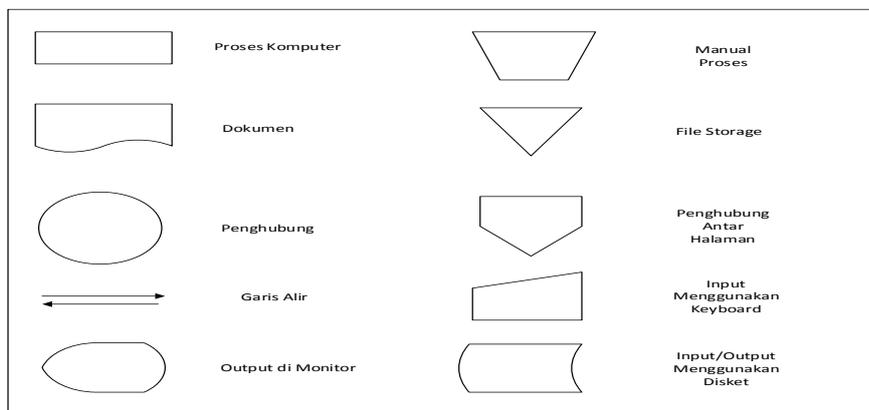
Lanjutan Tabel 2.3

Simbol	Deskripsi
	Arah panah mengarah pada objek yang memiliki operasi / metode, karena ini memanggil operasi / metode maka operasi/metode yang dipanggil harus ada pada diagram kelas objek yang berinteraksi
Pesan tipe send 	Menyatakan bahwa suatu objek mengirimkan data / masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
Pesan tipe return 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
Pesan tipe destroy 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

sumber: (Rossa & Shalahuddin, 2013)

2.2.16. Aliran Sistem Informasi

Menurut (Ismael, 2017:149) Aliran Sistem Informasi (ASI) merupakan bagan yang menunjukkan arus pekerjaan serta keseluruhan dari sistem. Bagan ini menjelaskan urutan dari prosedur-prosedur yang ada dalam system. Adapun simbol-simbol yang dapat digunakan pada Aliran Sistem Informasi (ASI) ini adalah sebagai berikut:



Sumber: (Ismael, 2017:149)

Gambar 2.5 Simbol-simbol Aliran Sistem Informasi

2.3. Penelitian Terdahulu

Penelitian terdahulu yang digunakan sebagai bahan penimbang dalam penelitian ini. Sesuai dengan judul maka penelitian terdahulu yang berkaitan adalah sebagai berikut:

Tabel 2.4 Penelitian Terdahulu

No	Penelitian dan tahun	Judul	Hasil
1.	Alfian Nur Rahma, Volume 1, No.1 April 2015 e-Proceeding of Applied Science	Aplikasi Penyewaan Lapangan Futsal Berbasis Web dan SMS Gateway	Membahas mengenai penyewa dapat mengetahui informasi jadwal lapangan yang <i>available</i> tanpa datang langsung ke tempat futsal dengan fitur cari jadwal dan melakukan booking jadwal melalui fitur SMS Gateway jika tidak mempunyai jaringan internet.

Lanjutan Tabel 2.4

No	Penelitian dan tahun	Judul	Hasil
2.	Humisar Hasugian dan Ahmad Nur Shidiq, Semantik 2012. ISSN: 979-26-0255-0	Rancang Bangun Sistem Informasi Industri Kreatif Bidang Penyewaan Sarana Olahraga	Memaparkan bahwa penerapan sistem komputerisasi akan meminimalkan terjadinya kesalahan-kesalahan input data yang sering dilakukan petugas dan dapat membantu pengelola Raihan <i>Sports Center</i> dalam menangani proses pelayanan penyewaan sarana olahraga dengan cepat serta menghasilkan informasi-informasi yang bermanfaat dalam pengambilan keputusan oleh pihak pengelola.
3.	Reza Fauzi Bastaman Vol. 12 No.1 2015 ISSN: 2302-7339	Perancangan Aplikasi Penyewaan Sarana dan Prasarana Di Sekolah Menengah Kejuruan NEGERI 3 GARUT	Memaparkan bahwa sistem yang dirancang dapat mengakomodasi kebutuhan dalam mempermudah dan mempercepat kinerja staf yang bertugas dalam mengelola data penyewaan Sarana dan Prasarana di SMKN 3 GARUT.
4.	Intan Septavia, Rd.Erwin Gunadhi dan Rina Kurniawati, 2015 <i>Jurnal - STT-Garut</i> ISSN : 2302-7339	Sistem Informasi Penyewaan Mobil Berbasis Web Di Jasa Karunia Tour And Travel	Sistem Informasi Penyewaan Mobil Berbasis Web Di Jasa Karunia Tour And Travel” memaparkan bahwa dengan adanya aplikasi yang berbasis web produktivitas kerja perusahaan dapat berkembang lebih baik lagi serta dapat memberikan informasi secara detail dan akurat mengenai mobil-mobil yang di sewakan oleh jasa karunia tour dan travel.