

## BAB II

### LANDASAN TEORI

#### 2.1 Konsep Dasar Sistem

Menurut Yuliawan, dkk (2013), bahwa Sistem dapat didefinisikan dengan pendekatan prosedur dan pendekatan komponen. Dengan pendekatan prosedur, sistem dapat didefinisikan sebagai kumpulan dari prosedur-prosedur yang mempunyai tujuan tertentu. Contoh sistem yang didefinisikan dengan pendekatan prosedur ini adalah sistem akuntansi. Sistem ini didefinisikan sebagai kumpulan dari prosedur-prosedur penerimaan kas, pengeluaran kas, penjualan, pembelian dan buku besar. Sedangkan dengan pendekatan komponen, sistem dapat didefinisikan sebagai kumpulan dari komponen yang saling berhubungan satu dengan yang lainnya membentuk satu kesatuan untuk mencapai tujuan tertentu. Contoh sistem yang didefinisikan dengan pendekatan ini misalnya adalah sistem komputer yang didefinisikan sebagai kumpulan dari perangkat keras dan perangkat lunak.

#### 2.2 Konsep Dasar Informasi

Menurut Yuliawan, dkk (2013), bahwa dalam bukunya Sistem Teknologi Informasi, dalam pendefinisian sistem terdapat dua macam pendekatan yaitu menekankan pada prosedur dan menekankan pada komponen atau elemennya.

#### 2.3 Konsep Dasar Sistem Informasi

Menurut Warman, dkk (2012), dimana sebuah sistem informasi merupakan kumpulan dari perangkat keras, lunak serta manusia yang akan mengolah dan menggunakan perangkat keras dan lunak tersebut. Informasi merupakan hal yang sangat penting dengan adanya informasi tersebut dapat diketahui kemajuan dan kegagalan proses pelaksanaan. Sistem yang kurang informasi menunjukkan bahwa sistem tersebut rapuh. Data merupakan informasi yang diolah supaya berguna bagi yang menerimanya. Definisi sistem secara umum yaitu sekumpulan proses

**Hak Cipta Dilindungi Undang-Undang**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

dan seperangkat elemen yang digabung serta dihimpun secara bersama serta saling berintegrasi untuk mencapai suatu tujuan dari organisasi.

Menurut Anisyia (2013), dimana Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*Building Block*), dimana masing-masing blok ini saling berintegrasi satu sama lainnya membentuk satu kesatuan untuk mencapai tujuannya. Adapun blok-blok tersebut adalah sebagai berikut:

1. Blok masukan

Meliputi metode-metode dan media untuk menangkap data yang akan dimasukan, dapat berupa dokumen-dokumen dasar.

2. Blok Model

Terdiri dari kombinasi prosedur, logika dan model matematik yang berfungsi memanipulasi data untuk keluaran tertentu.

3. Blok Keluaran

Berupa keluaran dokumen dan informasi yang berkualitas.

4. Blok Teknologi

Untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran serta membantu pengendalian dari sistem secara keseluruhan.

5. Blok Basis Data

Merupakan kumpulan data yang saling berhubungan satu dengan yang lainnya, tersimpan didalam perangkat keras komputer dan perangkat lunak untuk memanipulasi.

6. Blok Kendali

Meliputi masalah pengendalian yang berfungsi mencegah dan menangani kesalahan atau kegagalan sistem.

## 2.4 Definisi Sistem Informasi Manajemen

Menurut Sumarlinda (2015), bahwa Sistem informasi manajemen adalah suatu sistem manusia/mesin yang terpadu (*intregeted*) untuk menyajikan informasi guna mendukung fungsi operasi, manajemen, dan pengambilan keputusan dalam sebuah organisasi. Sistem ini menggunakan perangkat keras (*hardware*) dan perangkat lunak (*software*) komputer, prosedur pedoman, model manajemen dan keputusan , dan sebuah *database*.

## 2.5 Definisi Administrasi

Menurut White (1996) “Administrasi adalah suatu proses yang pada umumnya terdapat pada semua usaha sekelompok, Negara atau swasta, sipil atau militer, usaha yang besar atau yang kecil”.

Harris Muda Nasution dalam bukunya “Kursus Pengetahuan Administrasi” dalam arti yang sempit bahkan pengertian sehari-hari, maka administrasi artinya adalah Tata Usaha. Tata Usaha ialah suatu pekerjaan yang berhubungan dengan pekerjaan tulis-menulis, surat-menyurat dan mencatat/membukukan setiap perubahan atau kejadian yang terjadi di dalam organisasi.

## 2.6 Star Gym

### 2.6.1 Sejarah Star Gym Pekanbaru

Star Gym merupakan salah satu tempat pusat olahraga yang berada dipekanbaru. Dengan mengedepankan kualitas infrastruktur dan pelayanan kepada member agar Star Gym dapat bersaing dengan pebisnis sejenis lainnya yang berada di kota pekanbaru. Star Gym yang berdiri sejak 09 November 2013 didirikan oleh Roby Zuanda sebagai pemilik sekaligus pimpinan yang mengelola setiap proses bisnis yang berlangsung di Star Gym.

Saat ini Star Gym berlokasi di tengah jantung kota dan ditengah padatnya masyarakat yaitu di JL. Durian. Saat ini Star Gym tidak hanya menyediakan layanan fitness tapi juga menjual *merchandise*, suplemen dan alat Gym. Star Gym juga akan membuat daya saing bagi pesaing Gym lainnya (dapat dilihat pada Lampiran A).

**Hak Cipta Dilindungi Undang-Undang**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

### **2.6.2 Jenis member Star Gym Pekanbaru**

Adapun beberapa jenis member yang ada adalah sebagai berikut:

1. Perhari (non member) : Rp. 15.000
2. Biaya Pendaftaran : Rp. 20.000 (Mahasiswa/Pelajar/Umum)
3. Bayar perbulan Fitnes : Rp. 100.000

### **2.6.3 Biaya Perhari (non member)**

Jenis member yang ini merupakan member yang datang dengan melakukan aktifitas selama 1 hari dengan biaya Rp. 15.000 baik kategori mahasiswa/pelajar atau umum tetap dikenakan biaya perhari.

### **2.6.4 Biaya Pendaftaran**

Bagi yang ingin bergabung menjadi member di Star Gym harus melakukan pendaftaran dengan biaya sebesar Rp. 20.000 untuk mahasiswa, pelajar dan umum dengan ini calon member bisa dikatakan telah bergabung dengan Star Gym.

### **2.6.5 Biaya Perbulan Member Fitnes**

Member akan dikenakan biaya perbulan agar bisa menikmati fasilitas fitnes yang ada di Star Gym dalam kurun waktu satu bulan pasca member membayar untuk 1 bulan.

## 2.6.6 Visi dan Misi Star Gym Pekanbaru



Gambar 2.1 Logo Star Gym

(Sumber: Arsip Star Gym Pekanbaru)

Logo dari Star Gym ini menggambarkan dari sebuah ikatan kekeluargaan dalam sebuah lingkaran yang didalamnya berisi kan orang-orang yang memiliki badan yang atletis dengan dilambangkan dengan logo orang berbadan kekar, dimana arti Star sendiri adalah bintang yang melambang kan bahwa masih banyak yang lebih tinggi sehingga teruslah berusaha untuk mencapai bintang. Dengan kombinasi warna hitam yang berartikan perlindungan, putih berarti kedamaian dan kuning mengartikan kerja sama dan kebahagiaan.

### 1. Visi

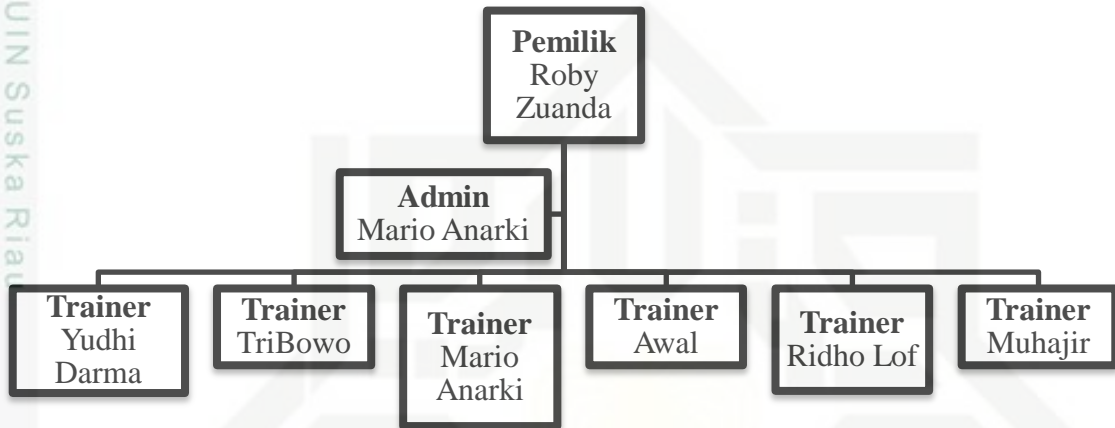
Membuat pusat olahraga yang sangat diminati oleh masyarakat Pekanbaru dengan harga yang terjangkau serta menjadikan pusat olahraga yang memiliki fasilitas yang sangat memuaskan. Selain itu dapat membantu member untuk mendapatkan program yang ingin dicapai.

### 2. Misi

- a. Menyediakan instruktur yang ramah dan mampu memberikan latihan yang baik kepada member.

- b. Menyediakan alat-alat fitness yang lengkap dan aman bagi member.
- c. Menciptakan suasana olahraga yang nyaman dan harmonis yang membuat member akan merasa betah.
- d. Membuat program-program yang diinginkan member akan tercapai.

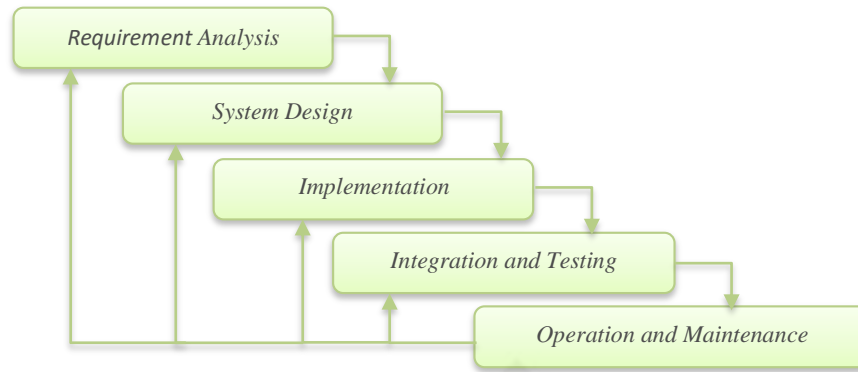
### 2.6.7 Struktur Organisasi Star Gym Pekanbaru



Gambar 2.2 Struktur Organisasi Star Gym  
 (Sumber: Arsip Star Gym Pekanbaru)

### 2.7 Waterfall

Menurut Suprianto, dkk (2014), bahwa Prosedur pengembangan yang digunakan pada sistem informasi manajemen administrasi ini adalah menggunakan prosedur pengembangan model *Waterfall*. Model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap analisis, desain, *coding*, *testing/verification*, dan *maintenance*. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Sebagai contoh tahap desain harus menunggu selesainya tahap sebelumnya yaitu tahap *requirement*. Secara umum tahapan pada model *waterfall* dapat dilihat pada Gambar 2.3.



Gambar 2.3 Siklus Pengembangan Dengan Metode *Waterfall*

Gambar di atas adalah tahapan umum dari model proses ini. Akan tetapi Pressman (2008) memecah model ini menjadi 6 tahapan meskipun secara garis besar sama dengan tahapan-tahapan model *waterfall* pada umumnya. Berikut adalah penjelasan dari tahap-tahap yang dilakukan di dalam model ini menurut Pressman:

- a. *Requirements definition*. Proses pencarian kebutuhan diintensifkan dan difokuskan pada *software*. Untuk mengetahui sifat dari program yang akan dibuat, maka para *software engineer* harus mengerti tentang domain informasi dari *software*, misalnya fungsi yang dibutuhkan, *user interface*. Dari 2 aktivitas tersebut (pencarian kebutuhan sistem dan *software*) harus didokumentasikan dan ditunjukkan kepada pelanggan.
- b. *System and Software Design*. Proses ini digunakan untuk mengubah kebutuhan-kebutuhan diatas menjadi representasi ke dalam bentuk "*blueprint*" *software* sebelum *coding* dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Seperti 2 aktivitas sebelumnya, maka proses ini juga harus didokumentasikan sebagai konfigurasi dari *software*.
- c. *Implementation and Unit Testing*. Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses *coding*. Tahap ini merupakan implementasi dari tahap design yang secara teknis nantinya dikerjakan oleh *programmer*.
- d. *Integration and Sytem Testing*. Sesuatu yang dibuat haruslah diujicobakan. Demikian juga dengan *software*. Semua fungsi-fungsi *software* harus

diujicobakan, agar *software* bebas dari *error*, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

e. *Operation and Maintenance*. Pemeliharaan suatu *software* diperlukan, termasuk di dalamnya adalah pengembangan, karena *software* yang dibuat tidak selamanya hanya seperti itu. Ketika dijalankan mungkin saja masih ada *error* kecil yang tidak ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada *software* tersebut. Pengembangan diperlukan ketika adanya perubahan dari *eksternal* perusahaan seperti ketika ada pergantian sistem operasi, atau perangkat lainnya.

## 2.8 *Android*

Menurut Kusumu, dkk (2015), dimana *Android* adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. *Android* awalnya dikembangkan oleh *Android Inc*, dengan dukungan finansial dari Google, yang kemudian membelinya pada Tahun 2005. Sistem operasi ini dirilis secara resmi pada Tahun 2007, bersamaan dengan didirikannya *Open Handset Alliance*, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. *Android* adalah sistem operasi dengan sumber terbuka, dan *Google* merilis kodenya di bawah lisensi *Apache*. Kode dengan sumber terbuka dan lisensi perizinan pada *Android* memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi, selain itu *Android* memiliki sejumlah besar komunitas pengembang aplikasi yang memperluas fungsionalitas perangkat, umumnya ditulis dalam versi kustomisasi bahasa pemrograman *Java*.

### Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

## 2.9 Arsitektur Android

Menurut Safaat (2012) arsitektur yang ada pada *platform Android* adalah sebagai berikut (Defni, dkk 2014):

1. *Application* dan *Widget Application* dan *widget* ini adalah *layer* dimana *user* berhubungan dengan aplikasi saja. Dimana *user* hanya akan menangani interaksi dengan aplikasi dan juga *widget*.
2. *Application Frameworks Android* adalah *open development platform* yaitu *Android* menawarkan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas mengakses perangkat keras, akses informasi *resources*, menjalankan *services background*, mengatur *alarm*, dan menambahkan status *notification*, dan sebagainya. Pengembang memiliki akses penuh menuju *API Frameworks* seperti yang dilakukan oleh aplikasi kategori inti. Arsitektur aplikasi dirancang supaya pengembang dengan mudah menggunakan kembali komponen yang sudah digunakan (*reuse*). Sehingga bisa disimpulkan *application frameworks* ini adalah *layer* di mana para pembuat aplikasi melakukan pengembangan atau pembuatan aplikasi yang akan dijalankan di sistem operasi *Android*, karena pada *layer* inilah aplikasi dapat dirancang dan dibuat, seperti *content-providers* yang berupa SMS dan panggilan telepon.
3. *Android Runtime Layer* yang membuat aplikasi *Android* dapat dijalankan dimana dalam prosesnya menggunakan implementasi *Linux*. *Dalvik Virtual Machine* merupakan mesin yang membentuk dasar kerangka aplikasi *Android*. Didalam *Android runtime* dibagi menjadi dua bagian yaitu:
  - a. *Core Libraries*: aplikasi *Android* dibangun dalam bahasa *Java*, sementara *dalvik* sebagai *virtual* mesinnya bukan *virtual* mesin *Java*, sehingga diperlukan sebuah *Libraries* yang berfungsi untuk menterjemahkan bahasa *Java/c* yang ditangani oleh *core Libraries*.
  - b. *Dalvik Virtual Machine*: *virtual* mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan pengembangan yang mampu membuat *linux kernel* untuk melakukan *threading* dan manajemen tingkat rendah.

## 2.10 Developer Android

Menurut Defni, dkk (2014) Pada perangkat *Android* bekerja dengan aturan yang simpel yaitunya membiarkan *User* mengelola aplikasi yang akan digunakan dengan bebas, dalam penggunaan sistem *Android* pengelolaan aplikasi pemasangan maupun pencopotan aplikasi dapat menggunakan aplikasi *Playstore* yang ada ataupun Amazon, dengan menggunakan akun google yang sudah terdaftar untuk dapat menggunakan *Google PlayStore* tersebut. Pengguna dapat saja menginstall program bawaan dari ponsel dan kemudian mengantinya dengan program pihak ketiga untuk dijalankan pada perangkat yang *User Android* miliki, dengan begitu penggunaan program yang datang dari luar perangkat dapat saja langsung digunakan untuk menggantikan dari program bawaan dari ponsel atau perangkat *Android*, sehingga dengan inilah kenapa *Android* dapat dikatakan sistem yang bebas, di *Google Playstore* sendiri pengguna dapat menjelajahi aplikasi yang sudah di *upload* pada market untuk dapat di pasang pada perangkat pengguna, *Google Playstore* sendiri akan memfilter aplikasi mana saja yang *Compatible* dengan perangkat yang dimiliki masing-masing pengguna. *Java Platform* sampai dengan saat ini memiliki 3 profil yaitu adalah, *Java ME (Micro Edition)* yaitunya adalah sitem *Java* yang mampu bekerja pada *Embedded System* seperti *Java Card* dan *Headphone*. *Java SE (Standart Edition)* yaitu adalah versi *Java* yang memiliki kinerja yang ditujukan membangun aplikasi yang mampu bekerja pada *PC, server* yang bersifat *stand alone*, kemudian *Java EE Enterprise Edition)* yaitunya adalah *Java* yang ditujukan untuk membuat aplikasi *Enterprise* seperti *Web Aplication (server)* dan *Enterprise Java Bean*.

## 2.11 Android SDK

Menurut Damarullah, dkk (2013), bahwa *Android SDK* merupakan *tools* bagi para *programmer* yang ingin mengembangkan aplikasi berbasis *Google Android*. *Android SDK* mencakup seperangkat alat pengembangan yang komprehensif. *Android SDK* terdiri dari *debugger, libraries, handset emulator, dokumentasi, contoh kode, dan tutorial*.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

## 2.12 Eclipse ADT

Menurut Damarullah, dkk (2013), bahwa ADT adalah sebuah *plugin* untuk Eclipse yang menyediakan alat yang terintegrasi dengan IDE Eclipse. Sedangkan *Android SDK (Software Development Kit)* menyediakan *Tools* dan *API* yang diperlukan untuk mengembangkan aplikasi pada *platform Android* dengan menggunakan bahasa pemrograman *Java*. ADT menyediakan fitur yang membantu untuk mengembangkan aplikasi *Android* dengan cepat. ADT menyediakan GUI untuk mengakses banyak *command line tools* SD dengan desain UI untuk perancangan, desain, dan pembuatan *user interface* aplikasi. Karena ADT adalah sebuah *plugin* untuk Eclipse, maka didapatkan fungsi IDE yang stabil dengan fitur *Android* yang spesifik yang digabung dengan ADT.

## 2.13 Object Oriented Analysis (OOA)

Menurut Anisya (2013), bahwa Analisis berorientasi objek adalah sebuah langkah investigasi untuk menemukan (mengidentifikasi) dan mendefinisikan objek-objek atau konsep-konsep yang ada di ruang masalah. Adapun tujuannya adalah untuk memahami permasalahan secara menyeluruh, mengungkapkan apa yang harus dikerjakan oleh sistem untuk memenuhi kebutuhan pemakai serta mengetahui ruang lingkup produk dan pemakai yang akan menggunakan produk tersebut.

OOA mempelajari permasalahan dengan menspesifikasikannya atau mengobservasi permasalahan tersebut dengan menggunakan metode berorientasi objek. Biasanya analisa sistem dimulai dengan adanya dokumen permintaan (*requirement*) yang diperoleh dari semua pihak yang berkepentingan (Misal: klien, developer, pakar). Dokumen permintaan memiliki 2 fungsi yaitu: memformulasikan kebutuhan klien dan membuat suatu daftar tugas (Sholih, 2006).

*Object Oriented Analysis (OOA)* melihat pada *domain* masalah, dengan tujuan untuk memproduksi sebuah model konseptual informasi yang ada di daerah yang sedang dianalisis. Model analisis tidak mempertimbangkan kendala-kendala pelaksanaan apapun yang mungkin ada, seperti konkurensi, distribusi, ketekunan,

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Ha  
Sci  
ta  
m  
ik  
/I  
S  
s  
a  
R  
i  
a  
u

S  
t  
a  
t  
e  
I  
s  
l  
a  
m  
i  
U  
n  
i  
v  
e  
r  
s  
i  
t  
a  
s  
S  
u  
l  
t  
a  
n  
S  
y  
r  
i  
K  
a  
s  
i  
m  
R  
i  
a  
u

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

atau bagaimana sistem harus dibangun. Kendala pelaksanaan ditangani selama desain berorientasi objek (OOD). Analisis dilakukan sebelum Design.

Sumber-sumber untuk analisis dapat persyaratan tertulis pernyataan, dokumen visi yang formal, wawancara dengan stakeholder atau pihak yang berkepentingan lainnya. Sebuah sistem dapat dibagi menjadi beberapa domain, yang mewakili bisnis yang berbeda, teknologi, atau bidang yang diminati, masing-masing dianalisis secara terpisah (Sholiq, 2006).

Hasil analisis berorientasi objek adalah deskripsi dari sistem secara fungsional yang diperlukan untuk melakukan pengembangan dalam bentuk sebuah model konseptual. Itu biasanya akan disajikan sebagai seperangkat menggunakan kasus, satu atau lebih UML diagram kelas, dan sejumlah diagram interaksi. Tujuan dari analisis berorientasi objek adalah untuk mengembangkan model yang menggambarkan perangkat lunak komputer karena bekerja untuk memenuhi seperangkat persyaratan yang ditentukan pelanggan (Sholiq, 2006).

#### 2.14 Metodologi Berorientasi Objek (*Object Oriented Design*)

Menurut Rachman, dkk (2012) Metodologi berorientasi objek dapat didefinisikan sebagai suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya.

Sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus (dinkapsulasi) menjadi kelompok data dan fungsi. Setiap komponen dalam sistem tersebut dapat mewarisi atribut, sifat dan komponen lainnya serta dapat berinteraksi satu sama lainnya.

#### 2.15 Teknik pemodelan dalam OOAD

1. Model Objek:
  - a. Model objek Menggambarkan struktur statis dari suatu objek dalam sistem dan relasinya.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

- b. Model objek berisi diagram objek. Diagram objek adalah *graph* dimana *node*-nya adalah kelas yang mempunyai relasi antar kelas.
2. Model Dinamik
  - a. Model dinamik menggambarkan aspek dari sistem yang berubah setiap saat.
  - b. Model dinamik dipergunakan untuk menyatakan aspek kontrol dari sistem.
  - c. Model dinamik berisi state diagram. *State diagram* adalah *graph* dimana *node*-nya adalah state dan *arc* adalah transisi antara *state* yang disebabkan oleh *event*.
3. Model Fungsional
  - b. Model fungsional menggambarkan transformasi nilai data di dalam sistem.

Model fungsional berisi data *flow diagram*. DFD adalah suatu *graph* dimana *node*-nya menyatakan proses dan *arc*-nya adalah aliran data.

### 2.16 Unified Modeling Language (UML)

Menurut Alim, dkk (2013) *Unified Modeling Language* (UML) adalah bahasa standar yang digunakan untuk menulis *blueprint* perangkat lunak. UML dapat digunakan untuk memvisualisasi, menspesifikasikan, membangun, dan mendokumentasikan artifak dari sistem perangkat lunak. UML terdiri atas tiga *building block*, yaitu:

1. *Things*

*Things* adalah *building block* berbasis objek yang utama dari UML. *Things* terdiri atas 4 macam, yaitu:

- a. *Structural Things*

*Structural things* adalah bagian model statis yang merepresentasikan elemen konseptual atau fisik. Jenis-jenis *structural things* adalah *class*, *interface*, *collaboration*, dan *Use Case*.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

b. *Behaviorial Things*

*Behaviorial things* adalah bagian dinamis dari model UML yang merepresentasikan *behavior*. Jenis *behaviorial things* adalah *interaction, state machine, dan activity*.

c. *Grouping Things*

*Grouping things* adalah bagian dari model UML yang berfungsi *untuk* mengelompokkan elemen. Satu-satunya jenis *grouping things* adalah *package*.

d. *Annotational Things*

*Annotational things* adalah bagian penjelas dari model UML. Jenis *annotational things* adalah *note* yang digunakan untuk memberikan komentar.

2. *Relationship*

*Relationship* merupakan *building block* UML yang berfungsi sebagai penghubung antar *things*. Jenis-jenis *relationship* antara lain *dependency, Association, Generalization, Realization*.

3. *Diagram*

Diagram adalah presentasi grafis yang merupakan *kombinasi* antara *things* dan *relationship*. Diagram dibuat untuk memvisualisasikan sistem dari sejumlah perspektif yang berbeda, sehingga diagram merupakan proyeksi terhadap sistem. UML mempunyai tiga belas diagram.

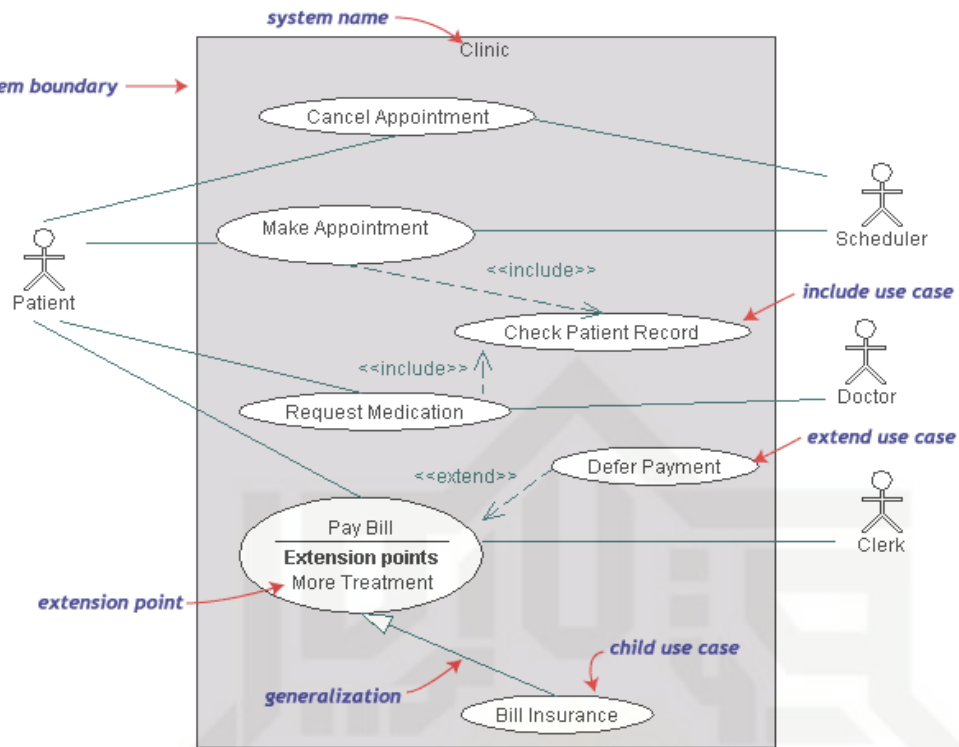
**2.16.1 Use Case Diagram**

Menurut Winata, dkk (2013) *Use Case* model adalah teknik pemodelan untuk mendapatkan *functional requirement* dari sebuah sistem, menggambarkan interaksi antara pengguna dan sistem, menjelaskan secara naratif bagaimana sistem akan digunakan, menggunakan skenario untuk menjelaskan setiap aktivitas yang mungkin terjadi. Ada beberapa bagian didalam *Use Case* model.

Hak Cipta Dilindungi Undang-Undang


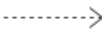


1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

© Hak cipta milik UIN Suska Riau









Gambar 2.4 Use Case Diagram  
(Sumber: Winata, dkk, 2013)

Tabel 2.1 Simbol Use Case Diagram

No	Gambar	Nama	Keterangan
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan Use Case.
2		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
3		Generalization	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
4		Include	Menspesifikasikan bahwa Use Case sumber secara eksplisit.

Tabel 2.1 Simbol *Use Case Diagram* (Lanjutan)

5		<i>Extend</i>	Menspesifikasikan bahwa <i>Use Case</i> target memperluas perilaku dari <i>Use Case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi






(Sumber: Winata, et.al, 2013)

### 2.16.2 Activity Diagram

Menurut Winata, dkk, (2013) *Activity Diagram* merupakan teknik untuk menjelaskan *business process*, menjelaskan teks *Use Case* dalam notasi grafis dengan menggunakan notasi yang mirip *flow chart*, meskipun terdapat sedikit perbedaan notasi.



Tabel 2.2 Simbol *Activity Diagram*

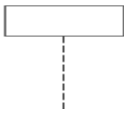

No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran


(Sumber: Winata, et.al, 2013)

### 2.16.3 *Sequence Diagram*

Menurut Hofmeister, Nord dan Soni (1999) *Sequence Diagram* menjelaskan interaksi obyek-obyek yang saling berkolaborasi (berhubungan), mirip dengan *Activity Diagram* yaitu menggambarkan alur kejadian sebuah aktivitas tetapi lebih detil dalam menggambarkan aliran data termasuk data atau *behaviour* yang dikirimkan atau diterima namun kurang mampu menjelaskan detil dari sebuah algoritma (Winata, et al., 2013).

Tabel 2.3 Simbol *Sequence Diagram*



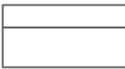



NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
---	---	----------------	--


### 2.16.4 Class Diagram

Menurut Rational Software Corporation (1997) *Class diagram* merupakan diagram paling umum yang dijumpai dalam pemodelan berbasis UML. Di dalam *Class diagram* terdapat *class* dan *interface* beserta atribut-atribut dan operasinya, relasi yang terjadi antar objek, *constraint* terhadap objek-objek yang saling berhubungan dan *inheritance* untuk organisasi class yang lebih baik. *Class diagram* juga terdapat *static view* dari elemen pembangun sistem. Pada intinya *Class diagram* mampu membantu proses pembuatan sistem dengan memanfaatkan konsep *forward* ataupun *reverse engineering*. (Winata, et al., 2013)

Tabel 2.4 Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<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> ).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri

Tabel 2.4 Simbol *Class Diagram* (Lanjutan)

7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya
---	---	--------------------	---

(Sumber: Winata, et.al, 2013)

### 2.17 PHP

Menurut Ramadhani, dkk (2009) PHP adalah bahasa *scripting* yang menyatu dengan HTML dan dijalankan pada *server side*. Artinya semua sintaks yang diberikan akan sepenuhnya dijalankan pada *server* sedangkan yang dikirimkan pada browser hanya hasilnya saja.

Pada awalnya PHP merupakan kependekan dari Personal *Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari *web*. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilsan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP (Anisya, 2013).

### 2.18 MySql

Menurut Ramadhani, dkk (2013) MySQL adalah sebuah perangkat lunak sistem manajemen basisdata SQL atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. *Relational Database Management System* (RDBMS).

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis di bawah lisensi *General Public License* (GPL). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh

dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL.

### 2.19 Database

Menurut Warman, dkk (2012), bahwa *Database* merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan diperangkat keras komputer dan digunakan diperangkat lunak untuk memanipulasinya. *Database* merupakan salah satu komponen yang sangat penting dalam sistem informasi, karena merupakan basis sistem dalam menyediakan informasi bagi para pemakai.

Sistem manajemen *database* merupakan perangkat lunak yang di disain untuk membantu dalam hal pemeliharaan dan *utility* data dalam jumlah yang besar. Penyusunan *database* bisa dilakukan secara sederhana dengan menuliskan data-datanya pada sebuah buku khusus atau juga dengan cara apapun yang digunakan. Untuk memudahkan penyusunan dan penampilan data, maka digunakan bentuk tabel.

### 2.20 Java

Menurut Defni, dkk (2014), dimana *Java* memiliki cara kerja yang unik dibandingkan dengan bahasa perograman lainnya yaitu bahasa perograman *Java* bekerja menggunakan *interpreter* dan juga *compiler* dalam proses pembuatan program, *Interpreter Java* dikenal sebagai perograman *bytecode* yaitu dengan cara kerja mengubah paket class pada *Java* dengan ekstensi *.Java* menjadi *class*, hal ini dikenal sebagai *class bytecode*, yaitunya *class* yang dihasilkan agar program dapat dijalankan pada semua jenis perangkat dan juga *platform*, sehingga program *Java* cukup ditulis sekali namun mampu bekerja pada jenis lingkungan yang berbeda.

### 2.21 Black Box Testing

Menurut M. Sidi Mustaqbal, dkk (2015), bahwa *Black Box Testing* berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat

mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program.

*Black Box Testing* bukanlah solusi alternatif dari *White Box Testing* tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *White Box Testing*.

*Black Box Testing* cenderung untuk menemukan hal-hal berikut:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.

Pengujian didesain untuk menjawab pertanyaan-pertanyaan berikut:

1. Bagaimana fungsi-fungsi diuji agar dapat dinyatakan valid?
2. Input seperti apa yang dapat menjadi bahan kasus uji yang baik?
3. Apakah sistem sensitif pada input-input tertentu?
4. Bagaimana sekumpulan data dapat diisolasi?
5. Berapa banyak rata-rata data dan jumlah data yang dapat ditangani sistem?
6. Efek apa yang dapat membuat kombinasi data ditangani spesifik pada operasi sistem?

Saat ini terdapat banyak metoda atau teknik untuk melaksanakan *Black Box Testing*, antara lain:

1. *Equivalence Partitioning*
2. *Boundary Value Analysis/Limit Testing*
3. *Comparison Testing*
4. *Sample Testing*
5. *Robustness Testing*
6. *Behavior Testing*
7. *Requirement Testing*
8. *Performance Testing*
9. Uji Ketahanan (*Endurance Testing*)
10. Uji Sebab-Akibat (*Cause-Effect Relationship Testing*)

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

## 2.22 HTML 5

Menurut M. Rosidi, dkk (2013), dimana HTML 5 Sama hal dengan sebuah *software*, HTML juga mempunyai berbagai versi. Versi terakhir dari HTML adalah HTML 5. Meskipun belum seluruhnya *browser* yang dapat menginterpretasikan *tag-tag* dari HTML 5 namun diyakin bahwa HTML 5 kan menjadi standar baru desain aplikasi. HTML sendiri merupakan suatu bahasa pemrograman yang umumnya digunakan untuk membuat sebuah halaman *aplikasi*, versi sebelumnya dari HTML adalah HTML 4.01 yang dirilis pada tahun 1999, pada versi HTML 5 saat ini telah mempunyai kapabilitas baru yang tidak dimiliki oleh HTML 4.01, diantaranya *tag* baru yang bernama *canvas* yang digunakan untuk menampilkan gambar atau animasi dan mendukung elemen pengembangan untuk membuat gambar garis atau gambar dengan menggambar di atas *canvas* tersebut.

HTML 5 merupakan kerjasama antara *World Wide Aplikasi Consortium* (W3C) dan *Aplikasi Hypertext Application Technology Working Group* (WHATWG). WHATWG bekerja bentuk aplikasi dan aplikasi, dan W3C bekerja dengan *XHTML 2.0*. Pada tahun 2006, mereka memutuskan untuk bekerja sama dan menciptakan sebuah versi baru dari HTML.

## 2.23 Android Studio

Menurut Juansyah (2015), bahwa *Android* studio adalah *Integrated Development Environment* (IDE) resmi untuk pengembangan aplikasi *Android* dan bersifat *open source* atau gratis. Peluncuran *Android* Studio ini diumumkan oleh *Google* pada 16 mei 2013 pada *event Google I/O Conference* untuk tahun 2013. Sejak saat itu, *Android* Studio menggantikan *Eclipse* sebagai IDE resmi untuk mengembangkan aplikasi *Android*.

*Android* studio sendiri dikembangkan berdasarkan *IntelliJ IDEA* yang mirip dengan *Eclipse* disertai dengan *ADT plugin (Android Development Tools)*. *Android* studio memiliki fitur:

- a. Projek berbasis pada *Gradle Build*.
- b. *Refactory* dan pembenahan bug yang cepat.
- c. *Tools* baru yang bernama “Lint” dikalim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
- d. Mendukung *Proguard and App-signing* untuk keamanan.
- e. Memiliki GUI aplikasi *Android* lebih mudah.
- f. Didukung oleh *Google Cloud Platfrom* untuk setiap aplikasi yang dikembangkan.

## 2.24 *Internet*

Menurut Sulihati dan Andriyani (2016), bahwa *Internet* yang merupakan singkatan dari *international network*, jaringan internasional dimana komputer-komputer di seluruh dunia saling berhubungan untuk berhubungan Antara suatu komputer dengan komputer yang lain, dengan berbagai model, berbagai sistem operasi, berbagai tipe jaringan digunakanlah suatu protokol (suatu sistem yang mengatur hubungan antara satu komputer dengan komputer yang lain) yaitu TCP/IP, singkatan dari *Transmission Control Protocol/Internet Protocol*.

## 2.25 *Website*

Menurut Marlana dan Sasongko (2010), bahwa *Website* merupakan sebuah halaman berisi informasi yang dapat dilihat jika komputer terkoneksi dengan internet. Dengan adanya *website*, semua orang di dunia bisa mendapatkan dan mengelola informasi dengan berbagai sumber yang tersedia di *internet*. *Website* sendiri saat ini bisa memuat berbagai macam media, mulai dari teks, gambar, suara bahkan video.

## 2.26 *Web view Android*

Menurut Meimaharani dan Laily (2014), bahwa *Web responsive design* pertama kali diperkenalkan oleh Ethan Marcotte pada artikelnya yang sangat inspiratif *Web Responsive Design*. Semakin banyaknya perangkat yang muncul dengan berbagai ukuran, maka sebuah situs perlu untuk mengenali ukuran perangkat pengguna. Ketika masih berpikir saat ada perangkat baru yang dirilis

dan akan memperbarui situs agar sesuai, maka harus mencari solusi yang lebih efektif dan responsif bagaimana cara agar situs hanya mengakui lebar browser saja tanpa melakukan pembaruan yang lebih spesifik. Yang membedakan *web responsive* dengan *website* yang lain yaitu penggunaan W3C CSS3 media dengan cairan proporsi berbasis *grid*, sehingga dapat fleksibel diterapkan diberbagai *gadget* yang ada.

#### Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

