

BAB II LANDASAN TEORI

2.1 Rancang Bangun

Rancang Bangun (desain) adalah tahap dari setelah analisis dari siklus pengembangan sistem yang merupakan pendefinisian dari kebutuhan-kebutuhan fungsional, serta menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, termasuk menyangkut mengkonfigurasi dari komponen-komponen perangkat keras dan perangkat lunak dari suatu sistem. (Jogiyanto, 2005:197).

2.2 Media Pembelajaran

2.2.1 Pengertian Media Pembelajaran

Daryanto (2013:5) mendefinisikan, “Media Merupakan salah satu komponen komunikasi yaitu sebagai pembawa pesan dari komunikator menuju komunikan, berdasarkan definisi tersebut dapat dikatakan bahwa proses pembelajaran merupakan proses komunikasi”. Media pembelajaran merupakan suatu alat atau sarana bantu yang digunakan sebagai pembelajaran, sehingga terjadi proses perubahan dalam kemampuan sikap dan perilaku siswa yang relatif permanen sehingga akibat dari pengalaman ataupun pelatihan.

Media pembelajaran adalah segala sesuatu yang digunakan untuk menyalurkan pesan serta dapat merangsang pikiran, perasaan, perhatian dan kemauan si pelajar sehingga dapat mendorong terjadinya proses belajar. Peran seorang tenaga pengajar adalah membuat proses pembelajaran menjadi efektif dengan menggunakan media yang sesuai dengan tujuan pembelajaran yang telah diterapkan.

2.2.2 Fungsi dan Manfaat Media Pembelajaran

Daryanto (2013:8) mendefinisikan, “Media memiliki fungsi sebagai pembawa informasi dari sumber (guru) menuju penerima (siswa) dalam kegiatan interaksi antara siswa dengan lingkungan, fungsi media dapat diketahui berdasarkan adanya kelebihan media dan hambatan yang mungkin timbul dalam proses pembelajaran”. Secara umum manfaat yang dapat diperoleh adalah proses pembelajaran lebih menarik, lebih interaktif, jumlah waktu mengajar dapat dikurangi, kualitas belajar siswa dapat ditingkatkan dan proses belajar mengajar dapat dilakukan dimana dan kapan saja, serta sikap belajar siswa dapat ditingkatkan.

2.2.3 Jenis Media Pembelajaran

Arsyad (2014:31) menjelaskan bahwa ada tiga jenis media pembelajaran yang dapat dikembangkan dan digunakan dalam kegiatan pembelajaran di sekolah yaitu:

1. Media Visual adalah media yang hanya dapat dilihat dengan menggunakan indra penglihatan terdiri atas media yang dapat diproyeksikan (*projected visual*) dan media yang tidak dapat diproyeksikan (*nonprojekted visual*).
2. Media audio adalah media yang mengandung pesan dalam bentuk auditif yang dapat merangsang pikiran, perasaan, perhatian dan kemauan para siswa untuk mempelajari bahan ajar dan sejenisnya.
3. Media audio visual merupakan kombinasi dari media audio dan media audio visual atau media pandang dengar.

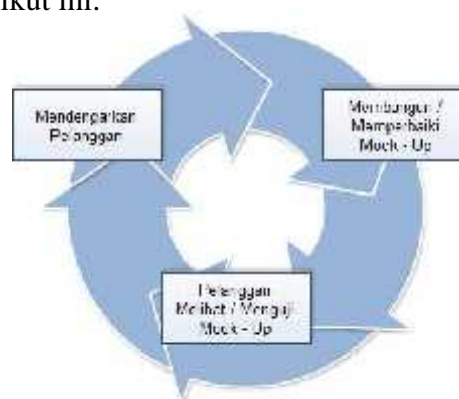
2.3 Definisi *Mobile*

Echols dan Shadily (2013:1) dalam kamus bahasa Inggris-Indonesia menuliskan bahwa “*Mobile*” adalah kata sifat yang berarti dapat bergerak atau dapat digerakkan dengan bebas dan mudah”. Namun *mobile* dapat pula diartikan sebuah benda yang berteknologi tinggi dapat bergerak tanpa menggunakan kabel. Contohnya seperti *smartphone*, *PDA* dan *tablet*.

2.4 Motode Model Prototipe

Rosa A.S dan M. Shalahuddin (2015:32) menjelaskan bahwa pelanggan (*customer*) membayangkan kumpulan kebutuhan yang diinginkan tapi tidak terspesifikasikan secara detail dari segi masukan (*input*), proses, maupun keluaran (*output*). Di sisi lain seorang pengembang perangkat lunak harus menspesifikasikan sebuah kebutuhan secara detail dari segi teknis dimana pelanggan sering kurang mengerti hal teknis ini.

Model prototipe (*prototyping model*) dimulai dari mengumpulkan ebutuhan pelanggan terhadap perangkat lunak yang dibuat. Lalu dibuatlah program prototipe agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program prototipe biasanya merupakan program yang belum jadi. Program ini biasanya menyediakan tampilan dengan simulasi alur perangkat lunak sehingga tampak seperti perangkat lunak yang sudah jadi. Program prototipe ini dievaluasi oleh pelanggan atau *user* sampai ditemukan spesifikasi yang sesuai dengan keinginan pelanggan atau *user*. Adapun gambar dari model prototipe ini dapat dilihat pada gambar 2.1 berikut ini:



Gambar 2.1 Ilustrasi model Prototipe

Mock-up adalah sesuatu yang digunakan sebagai model desain yang digunakan untuk mengajar, demokrasi, evaluasi desain, promosi, atau keperluan lain. Sebuah *mock-up* disebut sebagai prototipe perangkat lunak jika menyediakan atau mampu mendemonstrasikan pengujian desain sistem perangkat lunak. Iterasi terjadi pada pembuatan prototipe sampai sesuai dengan keinginan pelanggan (*customer*) atau user.

Model prototipe juga memiliki kelemahan sebagai berikut:

1. Pelanggan dapat sering mengubah-ubah atau menambah-tambah spesifikasi kebutuhan karena menganggap aplikasi sudah dengan cepat dikembangkan, karena adanya iterasi ini dapat menyebabkan pengembang banyak mengalah dengan pelanggan karena perubahan atau penambahan spesifikasi kebutuhan perangkat lunak.
2. Pengembang lebih sering mengambil kompromi dengan pelanggan untuk mendapatkan prototipe dengan waktu yang cepat sehingga pengembang lebih sering melakukan segala cara (tanpa idealis) guna menghasilkan prototipe untuk didemokrasikan. Hal ini dapat menyebabkan kualitas perangkat lunak yang kurang baik atau bahkan menyebabkan iterative tanpa akhir.

Permasalahan dapat terjadi pada model prototipe, hal ini dapat diatasi dengan melakukan perjanjian antara pengembang perangkat lunak dengan pelanggan (*customer*) atau user agar model prototipe hanya digunakan untuk mendefinisikan spesifikasi kebutuhan perangkat lunak, tapi tidak untuk seluruh proses pengembangan seluruh sistem perangkat lunak.

2.5 *Unified Modeling Language*

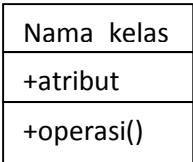


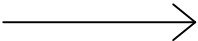

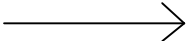

Rosa A.S dan M. Shalahuddin (2015:137) menjelaskan bahwa “Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa permodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek”.

Alat-alat yang digunakan dalam suatu metodologi umumnya berupa suatu gambar diagram atau grafik. Adapun alat pengembangan sistem yang digunakan yaitu sebagai berikut:

1. *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Banyak berbagai kasus, perancangan kelas yang dibuat tidak sesuai dengan kelas-kelas yang dibuat pada perangkat lunak, sehingga tidaklah ada gunanya lagi sebuah perancangan karena apa yang dirancang dan hasil jadinya tidak sesuai. *Class diagram* pada media pembelajaran ini dapat dilihat pada tabel 2.1 berikut:

Tabel 2.1 Bagan *Class Diagram*

Simbol	Deskripsi
kelas 	Kelas pada struktur sistem
Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
Asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan / <i>dependency</i> 	Kebergantungan antarkelas
Aggregation 	Menggambarkan relasi agregasi

2. Use Case Diagram

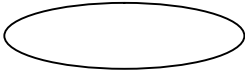
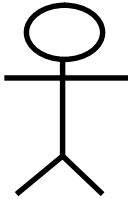
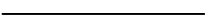
Use Case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang berhak menggunakan fungsi-fungsi itu.

Syarat penamaan pada *use case* adalah nama didefinisikan semudah mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

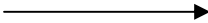
1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor.

Use case diagram pada media pembelajaran ini dapat dilihat pada tabel 2.2 berikut:

Tabel 2.2 Bagan *Use Case* Diagram

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

Tabel 2.2 Bagan *Use Case* Diagram (lanjutan)

<p>Ekstensi / <i>extend</i></p> 	<p>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.</p>
---	--

3. *Activity* Diagram


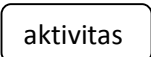



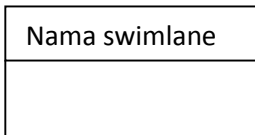
Diagram aktivitas atau *activity* diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisni dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

Activity diagram pada media pembelajaran ini dapat dilihat pada tabel 2.3 berikut:

Tabel 2.3 Bagan *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>decision</i> 	Asosiasi percabangan diaman 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

4. *Sequence Diagram*

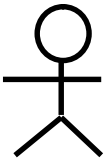
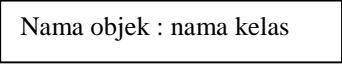
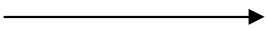

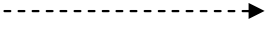
Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah

dicukup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Sequence Diagram pada media pembelajaran ini dapat dilihat pada tabel 2.4 berikut:

Tabel 2.4 Bagan *Sequence Diagram*

Simbol	Deskripsi
Aktor 	Pelaku dalam <i>sequence diagram</i>
Objek 	Menambahkan objek baru pada diagram
<i>Stimulus</i> 	Menggambar pesan (<i>message</i>) antar dua objek
<i>Selfstimulus</i> 	Menggambar pesan (<i>message</i>) yang menuju dirinya sendiri
<i>Return message</i> 	Menggambarkan pengembalian diri pemanggilan prosedur

2.6 Aplikasi

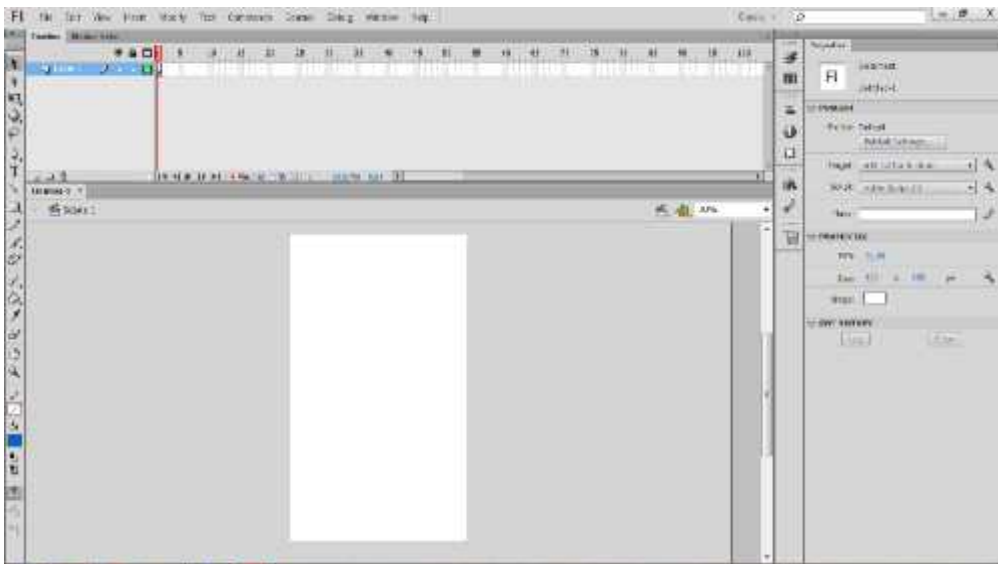
Setiawati (2009:9) mendefinisikan bawah “Aplikasi merupakan penerapan pengimplementasian suatu hal, data, permasalahan, pekerjaan kedalam suatu sarana atau media yang dapat digunakan untuk menerapkan atau mengimplementasikan hal atau permasalahan tersebut sehingga berubah menjadi suatu bentuk yang baru, tanpa menghilangkan nilai-nilai dasar dari hal, data, permasalahan atau pekerjaan”.

2.7 Multimedia

Rosch dalam buku Darmawan (2011: 32) mengatakan bahwa multimedia dipandang sebagai suatu kombinasi antara komputer dan video. Houghton dalam Sumarno (2011:19) mendefinisikan multimedia sebagai bentuk komunikasi multi bentuk dengan menggunakan perangkat komputer atau sejenisnya. Pendapat yang hampir sama dikemukakan oleh Mc. Cormik dalam Darmawan (2011:32) yang mengatakan bahwa multimedia merupakan sebuah kombinasi tiga elemen yaitu suara, gambar dan teks. Hal ini juga meliputi pengertian yang dikemukakan oleh Robin dan Linda dalam Darmawan (2011: 32) menyebutkan bahwa multimedia merupakan alat yang dapat menciptakan presentasi yang dinamis dan interaktif yang mengkombinasikan teks, grafik, animasi, audio, dan video.

2.8 Adobe Flash Professional CS6

Maulana (2014:31), pada dasarnya baik *Flash* versi *Macromedia* maupun versi *Adobe*, area kerjanya terbagi menjadi beberapa bagian, di antaranya menu, *toolbox*, *timeline*, *stage*, dan *panel*.



Gambar 2.2 Tampilan Area Kerja Adobe Flash CS6

a. *Toolbox*

Di dalam *toolbox* terdapat macam-macam *tool* yang bisa kita gunakan untuk menggambar objek. *Toolbox* ini terdiri dari beberapa bagian yaitu *selection tool*, *drawing tool*, *painting tool*, dan *navigation tool*.

b. *Timeline*

Timeline pada *Adobe Flash CS6* secara *default* terletak diatas *stage*. *Timeline* berfungsi untuk mengontrol keseluruhan objek dan animasi yang terdapat pada *stage*.

c. *Layer*

Layer dapat diilustrasikan seperti tumpukan-tumpukan yang berisi objek di dalamnya. Dengan *layer*, kita bisa mengatur *movie* pada *stage* dan menentukan kedalaman atau lapisan suatu objek. Untuk menambahkan *layer* baru, kita dapat memilih menu *Insert > Timeline > Layer* atau dengan mengklik *icon* yang terdapat pada bagian bawah *timeline*.

Terdapat beberapa mode yang dapat dipilih pada *layer* di antaranya :

- *Guide layer*, berfungsi untuk membuat animasi dengan menggunakan jalur atau *track* berupa garis yang telah dibuat.
- *Motion layer*, digunakan untuk membuat pergerakan animasi *tween*.
- *Masking layer*, digunakan untuk membuat animasi efek *masking*.

d. *Frame*

Frame merupakan bagian dari *layer* yang digunakan untuk mengatur pergerakan animasi. Di dalam *frame* bisa terdiri dari teks, gambar, audio, video, dan kode program *ActionScript*.

Frame terdiri dari beberapa bentuk, di antaranya :

- *Keyframe*, *frame* yang memiliki bentuk bulatan hitam, menandakan bahwa di dalamnya terdapat objek.
- *Blank keyframe*, *frame* berbentuk bulat putih, menandakan bahwa *frame* masih kosong.
- *Action frame*, *frame* dengan bulatan putih dan terdapat huruf 'a', menandakan di dalamnya terdapat kode program *ActionScript*.

- *Frame label*, yaitu *frame* yang diberi nama *label* dan ditandai dengan bendera warna merah.

e. *Stage*

Stage adalah area putih berbentuk kotak yang terletak ditengah area kerja *Flash*. Seperti istilah *stage* pada panggung teater, *stage* di dalam *Flash* berfungsi untuk menampilkan semua objek maupun *movie* yang berjalan di atasnya. Objek yang diletakan pada *stage* bisa berupa teks, gambar, dan video.

f. Panel *Properties*

Panel ini menampung semua properti yang terdapat pada *tool-tool* dalam *Flash*. Untuk mengaktifkannya, pilih menu *Windows > Properties*.

g. Panel *Library*

Seperti layaknya perpustakaan, *library* pada *Flash* berfungsi menampung semua simbol yang telah dibuat maupun hasil impor dari file luar seperti audio, video, gambar, dll. Untuk menampilkan *panel library*, bisa melalui menu *Window > Library* atau dengan menekan tombol CTRL+L.

h. Panel *Action*

Panel ini berfungsi sebagai tempat menuliskan *ActionScript*. Untuk mengaktifkannya, pilih menu *Window > Action* atau menekan tombol F9.

2.9 Action Script 3.0

Maulana (2014:18) mendefinisikan, *ActionScript 3.0* atau disingkat AS3 merupakan bahasa pemrograman yang bekerja pada *Adobe Flash*, *Flex*, dan *FlashDevelop*. *ActionScript 3.0* pertama kali dirilis pada tahun 2006 bersamaan dengan diluncurkannya *Flash* versi 9 sekaligus *Flash* pertama yang kini telah diakuisisi oleh *Adobe System Inc.* yaitu *Adobe Flash CS3*.

ActionScript 3.0 adalah bahasa pemrograman yang didasarkan pada *ECMAScript*, yaitu standar bahasa pemrograman yang dikembangkan oleh *ECMA (European Computer Manufacturers Association)*. Dengan standar

ini, AS3 mampu melakukan integrasi data yang cepat dengan berbagai bahasa pemrograman lainnya seperti *JavaScript* dan *XML*.

2.10 Cool Edit Pro

Agus (2010:10) *Cool Edit Pro* digunakan untuk perekaman suara yang berfungsi sebagai dubbing sebuah adegan dengan suara sendiri. Peralatan yang digunakan dalam perekaman suara menggunakan *Cool Edit Pro* adalah *microphone* dan *speaker*.

2.11 Adobe Premiere CC 2015

Merupakan program aplikasi berbasis video editing yang diproduksi oleh perusahaan perangkat lunak *Adobe System Incorporated*. *Adobe Premiere Pro CC 2015* memiliki banyak fasilitas dan fitur yang menunjang dalam penggunaan video editing professional. Dalam pembuatan media promosi 2D motion graphics *Adobe Premiere CC 2015* berfungsi sebagai pengeditan video serta pengolahan suara sehingga media promosi akan lebih menarik.

2.12 Populasi

Sugiyono (2011:80) pengertian populasi adalah wilayah generalisasi yang terdiri atas: obyek/subyek yang mempunyai kualitas dan karakteristik tertentu yang ditetapkan oleh peneliti untuk dipelajari dan kemudian ditarik kesimpulannya.

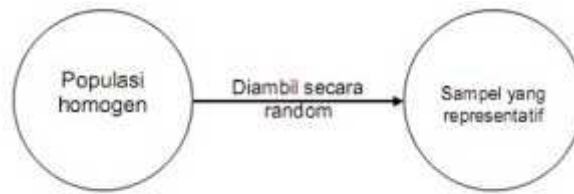
2.13 Sample

Pengertian sample menurut Sugiyono (2001:118) adalah bagian dari jumlah dan karakteristik yang dimiliki oleh populasi tersebut.

2.14.1 Simple Random Sampling

Menurut Sugiyono (2011:57) dinyatakan simple (sederhana) karena pengambilan sample anggota populasi dilakukan secara acak tanpa memperhatikan strata yang ada dalam populasi tersebut. Margono (2004:126) menyatakan bahwa simple random sampling adalah teknik untuk mendapatkan sample yang langsung dilakukan pada unit sampling. Cara demikian dilakukan bila anggota populasi dianggap tidak homogen. Teknik ini dapat digunakan bilamana jumlah unit sampling di dalam populasi tidak terlalu besar. Misal, populasi terdiri dari 500 orang mahasiswa program S1 (unit sampling). Untuk memperoleh sample sebanyak 150 orang dari populasi tersebut, digunakan teknik ini, baik dengan cara undian, ordinal, maupun tabel bilangan random.

Teknik ini dapat digambarkan dibawah ini :



Gambar 2.3 Teknik Simple Random Sampling