

BAB II

LANDASAN TEORI

Untuk mendukung pembuatan laporan ini, maka perlu dikemukakan hal-hal atau teori-teori yang berkaitan dengan permasalahan dan ruang lingkup pembahasan sebagai landasan dalam pembuatan laporan ini.

2.1. Tinjauan Pustaka

Dalam pembuatan sistem informasi arsip akta kelahiran ini penulis mengambil kajian pustaka dari dua penelitian sebelumnya. Penelitian pertama berjudul “Rancang Bangun Sistem Informasi Manajemen Kearsipan Pada Badan Perizinan Terpadu dan Penanaman Modal Kota Makassar” yang disusun oleh Rita Lestari (2016).

Penelitian ini memiliki kesamaan dan perbedaan sistem yang akan dibuat. Persamaannya adalah sama-sama Sistem informasi manajemen pengarsipan berbasis *web*. Sistem yang digunakan dalam penelitian tersebut juga menggunakan *PMYSQL*, *PHP* dan *XAMPP*. Sedangkan untuk perbedaan, sistem tersebut hanya mengakomodir penyimpanan hasil *scan* dokumen dengan format gambar atau *jpeg*. Untuk sistem yang dibuat penulis sudah mengakomodir format *filepdf* maupun *jpeg*.

Kajian pustaka dari penelitian yang kedua Rizki Alfiasca P. (2014), yang berjudul “Rancang Bangun Sistem Informasi Manajemen Arsip Berbasis web pada Rumah Sakit Bedah Surabaya”. Penelitian ini bertujuan menghasilkan sistem informasi manajemen arsip yang sesuai dengan proses bisnis pada Rumah Sakit Bedah Surabaya.

Penelitian ini memiliki kesamaan dan perbedaan sistem yang akan dibuat. Persamaannya adalah sama-sama berbasis *web*, Namun yang menjadi perbedaan ialah sistem yang akan dibuat penulis memiliki sistem pencarian data dan dokumen yang tidak memiliki masa kadaluarsa, serta menggunakan sistem Operasi Windows 7. Sedangkan penelitian yang dilakukan oleh Rizki Alfiasca P memiliki sistem yang dapat menentukan data atau dokumen yang memiliki masa kadaluarsa, sistem informasi yang dapat ditangani oleh sistem yang dibuat

adalah dokumen dalam bentuk pdf serta menggunakan Sistem Operasi Windows XP Profesional.

Berikut tampilan dari menu utama Sistem Informasi Manajemen Kearsipan pada Badan Perizinan Terpadu dan Penanaman Modal Kota Makassar sebagaimana pada Gambar 2.1 dan halaman login pimpinan pada Gambar 2.2.



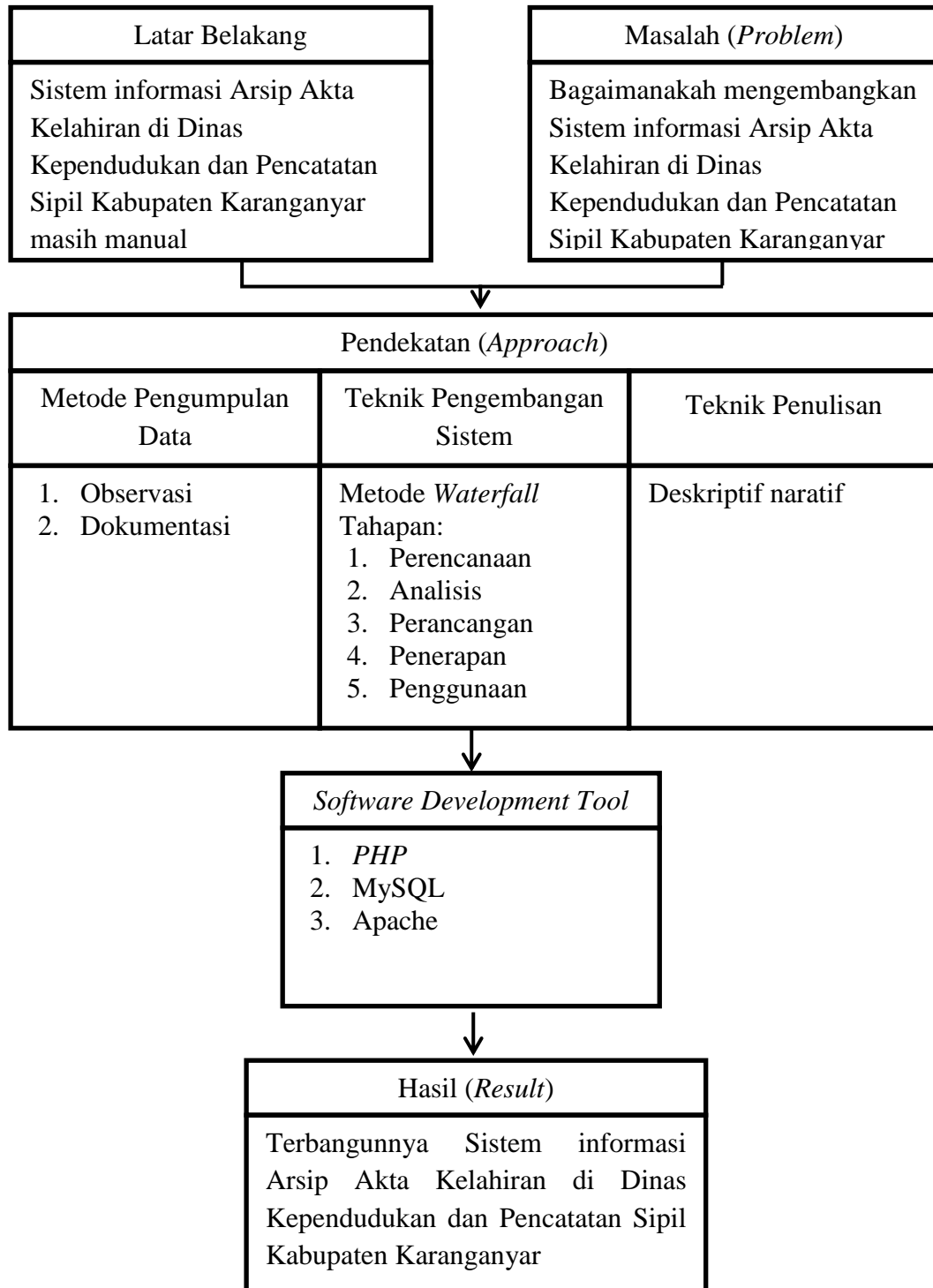
Gambar 2.1. Halaman Utama



Gambar 2.2. Halaman Login Pimpinan

2.2. Kerangka Berpikir

Kerangka berpikir disajikan secara skematis sebagaimana dalam Gambar 2.4. berikut ini.



Gambar 2.4. Kerangka Berfikir

Keterangan Gambar :

1. Latar belakang

Bahwa untuk pengelolaan berkas arsip yang selama ini berjalan di Dinas Kependudukan dan Pencatatan Sipil Kabupaten Karanganyar, belum memanfaatkan teknologi informasi secara menyeluruh.

2. Masalah

Dikarenakan pengelolaan berkas masih manual, akan banyak ditemui masalah, baik untuk pemeliharaan, pencarian berkas maupun pengolahan untuk data statistiknya.

3. Hasil

Untuk itu sebagai solusi untuk mengatasi permasalahan yang selama ini terjadi berkaitan dengan penanganan dan pencarian arsip, maka sistem informasi arsip akta kelahiran ini harus terbangun.

4. Pendekatan

Pendekatan yang dilakukan untuk penelitian ini yaitu :

- a. Metode pengumpulan data dengan melakukan observasi di lokasi penelitian dan melakukan dokumentasi
- b. Teknik pengembangan sistem menggunakan metode *waterfall*
- c. Teknik penulisan deskriptif naratif

5. Software

Software yang digunakan untuk pembuatan sistem ini diantaranya, PHP, MySQL dan Apache.

6. Hasil

Hasil dari penelitian ini adalah terbangunnya sistem informasi arsip akta kelahiran di Dinas Kependudukan dan Pencatatan Sipil Kabupaten Karanganyar

2.3. Teori Pendukung

Penyusunan Tugas Akhir memerlukan suatu referensi pendukung yang digunakan sebagai landasan teori agar penelitian dapat berjalan dengan benar dan tidak menyimpang dari kaedah ilmu pengetahuan yang ada. Landasan teori diperoleh dari berbagai sumber dan literatur yang mempublikasikan pendapat beberapa ilmuwan yang digunakan sebagai pendukung pembahasan masalah dalam penelitian Tugas Akhir. Berikut ini beberapa diantaranya:

2.3.1. Sistem Informasi

Menurut Iswandy (2015), Sistem Informasi adalah sistem di dalam suatu organisasi yang berfungsi mengolah transaksi, mendukung operasi, serta menyediakan informasi yang diperlukan bagi pihak yang berkepentingan.

Jadi dapat diartikan Sistem Informasi merupakan sarana bagi sebuah organisasi untuk mendukung operasi atau bisnis, agar bisa menyediakan informasi untuk yang membutuhkan. Sistem informasi terdiri dari Seperangkat komponen yang saling berhubungan dan berintegrasi yang berfungsi memproses, mendistribusikan, serta menyimpan informasi guna mendukung keputusan dan pengawasan di dalam organisasi.

2.3.2. Arsip

Secara etimologi kata arsip berasal dari bahasa Yunani (*Greek*), yaitu *archium* yang artinya peti untuk menyimpan sesuatu. Semula pengertian arsip itu memang menunjukkan tempat atau gedung tempat penyimpanan arsipnya, tetapi perkembangan terakhir orang lebih cenderung menyebut arsip sebagai warkat itu sendiri.

Menurut Undang-Undang No. 7 tahun 1971 tentang “Ketentuan Pokok Kearsipan” pada Bab I pasal 1 berbunyi sebagai berikut:

- (1) Naskah-naskah yang dibuat dan diterima oleh lembaga Negara dan Badan-badan Pemerintahan dalam bentuk corak apapun, baik dalam keadaan tunggal maupun berkelompok dalam rangka pelaksanaan kegiatan Pemerintah.

(2) Naskah-naskah yang dibuat dan diterima oleh Badan-badan swasta dan/atau perorangan dalam bentuk corak apapun, baik dalam keadaan tunggal maupun berkelompok, dalam rangka pelaksanaan kebangsaan.

Sedang menurut Undang-Undang Republik Indonesia Nomor 43 Tahun 2009 tentang kearsipan menyebutkan bahwa:

“Arsip adalah rekaman kegiatan atau peristiwa dalam berbagai bentuk dan media sesuai dengan perkembangan teknologi informasi dan komunikasi yang dibuat dan diterima oleh lembaga negara, pemerintahan daerah, lembaga pendidikan, perusahaan, organisasi politik, organisasi kemasyarakatan, dan perseorangan dalam pelaksanaan kehidupan bermasyarakat, berbangsa dan bernegara”

Jadi dengan terbitnya Undang-Undang nomor 43 tahun 2009 tersebut, pemerintah secara eksplisit menyebutkan bahwa pergeseran dari pengertian arsip itu sendiri. Dari dulunya yang terkesan bahwa arsip adalah naskah-naskah yang berbasis kertas, mulai bergeser menjadi arsip digital.

2.3.3.Web

Web atau *Website* dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau bergerak, suara dan 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-jaringan halaman (Riyadi dkk., 2013).

Halaman web tersebut ditempatkan di *server web* yang dapat diakses melalui jaringan seperti internet, ataupun jaringan lokal (LAN) melalui alamat internet yang dikenali sebagai URL (*Uniform Resource Locator*), yaitu karakter tertentu yang menunjukkan sebuah alamat yang bisa diakses melalui *web browser*.

2.3.4.PHP

Menurut Ramadhani dkk.(2013), PHP atau *Hypertext Preprocessor* adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam TML. PHP banyak dipakai untuk memprogram situs *web* dinamis, PHP juga dapat digunakan untuk membangun sebuah CMS.

Ketika teknologi internet semakin berkembang, saat ini bahasa PERL dan CGI sudah mulai ditinggalkan karena jauh ketinggalan jaman. Sebagian besar designer web banyak beralih ke bahasa *server-side scripting* yang lebih dinamis seperti PHP. *Software* ini juga bersifat *open source*. Seluruh aplikasi berbasis *web* dapat dibuat dengan PHP. Namun kekuatan yang paling utama PHP adalah pada konektivitasnya dengan *system database* di dalam *web*.

2.3.5. MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris : *database management system*) atau DBMS yang *multi thread*, *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 dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. *Relational Database Management System* (RDBMS), (Ramadhani dkk, 2013).

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). 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 (*Structured Query Language*)

2.3.6. Web Browser

Menurut Dimuksa dan Sukadi (2013), *web browser* adalah aplikasi yang digunakan untuk menjelajahi internet atau untuk mencari sebuah informasi dari suatu halaman *web* atau *blog*, awalnya *web browser* hanya berorientasi pada teks dan belum dapat menampilkan gambar. Namun, *web browser* sekarang tidak hanya menampilkan text dan gambar tetapi juga *filemultimedia* seperti video dan suara. *Web browser* juga dapat mengirim dan menerima *e-mail*, mengelola bahasa HTML (*Hyper Text Markup Language*) sebagai *input*, dan menjadikan halaman *web* sebagai hasil *output* yang informatif.

2.3.7. UML

2.3.7.1. Use Case Diagram


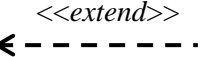
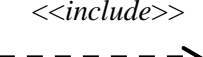


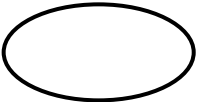
Use case diagram adalah diagram yang digunakan untuk menggambarkan secara ringkas siapa yang menggunakan sistem dan apa saja yang bisa dilakukannya. Diagram *use case* tidak menjelaskan secara detail tentang penggunaan *use case*, namun hanya memberi gambaran singkat hubungan antara *use case*, *aktor*, dan *system*. Melalui diagram *use case* dapat diketahui fungsi-fungsi apa saja yang ada pada *system*. (Rosa dan Shalahudin, 2014)

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara *aktor* dengan *sistem*. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, *meng-create* sebuah daftar dan sebagainya.

Seorang/sebuah *aktor* adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal.

Simbol-simbol yang digunakan dalam *use case* diagram disajikan pada Tabel 2.1.

Tabel 2.1. *Use Case Diagram*

No.	Gambar	Nama	Keterangan
1.		<i>Association</i>	Jalur komunikasi antara aktor dan use case berpartisipasi.
2.		<i>Extend</i>	Insertion tambahan ke use case yang tidak diketahui
3.		<i>Include</i>	Insertion tambahan ke use case yang secara eksplisit menggambarkan insertion
4.		<i>Use Case Generalization</i>	Hubungan antara Use Case satu dengan Use Case lainnya.
5.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan berinteraksi dengan Use Case.
6.		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.

2.3.7.2. Class Diagram

Class Diagram mendeskripsikan jenis-jenis obyek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. *Class Diagram* juga menunjukkan properti dan operasi sebuah *class* dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut. Diagram kelas mempunyai 3 macam relationships (hubungan) sebagai berikut:

a. *Association*

Suatu hubungan antara bagian dari dua kelas, terjadi *association* antara dua kelas jika salah satu bagian dari kelas mengetahui yang lainnya dalam melakukan suatu kegiatan. Di dalam diagram, sebuah *association* adalah penghubung yang menghubungkan dua kelas.

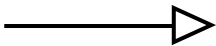

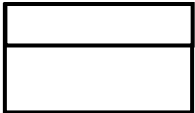
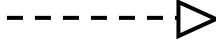
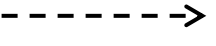
b. *Aggregation*

Suatu *association* dimana salah satu kelasnya merupakan bagian dari suatu kumpulan, *aggregation* memiliki titik pusat yang mencakup keseluruhan bagian sebagai contoh: *Order detail* merupakan kumpulan dari *order*.

c. *Generalization*

Suatu hubungan turunan dengan mengasumsikan satu kelas merupakan satu *super class* (kelas super) dari kelas yang lain. *Generalization* memiliki tingkatan yang berpusat pada *super class*, contoh: *Payment* adalah *super class* dari *Cash*, *Check*, *Kredit*. *Simbol Class Diagram* dapat dilihat pada Tabel 2.2. dibawah ini.




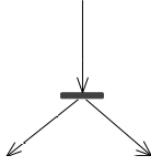
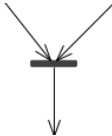
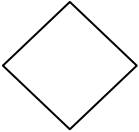

Tabel 2.2. *Class Diagram*

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>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
3.		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4.		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
5.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.

2.3.7.3. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* yaitu:




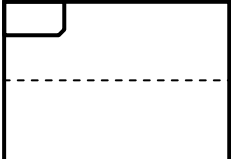
Tabel 2.3. *Activity Diagram*

No.	Gambar	Nama	Keterangan
1.		<i>Start Point</i>	Diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
2.		<i>End Point</i>	Akhir aktifitas
3.		<i>Activities</i>	Menggambarkan suatu proses/ kegiatan bisnis
4.		<i>Fork/</i> percabangan	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
5.		<i>Join</i> (penggabungan) <i>atau rake</i>	Digunakan untuk menunjukkan adanya dekomposisi
6.		<i>Decision Points</i>	Menggambarkan pilihan untuk pengambilan keputusan, true atau false
7.		<i>Swimlane</i>	Pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa

2.3.7.4. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *Sequence Diagram* yaitu:

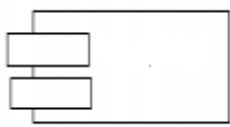

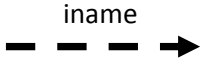
Tabel 2.4. *Sequence Diagram*

No.	Gambar	Nama	Keterangan
1.		<i>Life Line</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>Actor</i>	Pengguna di luar sistem.
4.		<i>Fragment</i>	Menggambarkan batas grafis suatu diagram.

2.3.7.5. Component Diagram

Component Diagram merupakan bagian fisik dari sebuah sistem, karena menetap di komputer tidak berada dianalisis. *Component* terhubung melalui antar muka yang digunakan dan dibutuhkan. Simbol *Component Diagram* dapat dilihat pada Tabel 2.5.

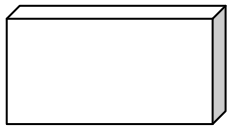
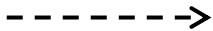

Tabel 2.5. *Component Diagram*

No.	Gambar	Nama	Keterangan
1.		<i>Component</i>	<i>Physical</i> dari sebuah sistem.
2.		<i>Interface</i>	A <i>name state</i> informasi yang menjadi ciri dari perilaku
3.		<i>Usage</i>	Situasi dimana satu elemen membutuhkan lain fungsi yang benar

2.3.7.6. Deployment Diagram

Deployment Diagram menunjukkan susunan fisik sebuah sistem, menunjukkan bagian perangkat lunak mana yang berjalan pada perangkat keras mana. *Deployment Diagram* juga menggambarkan tata letak sebuah sistem secara fisik, menampakan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware*, menunjukkan hubungan komputer dengan perangkat (nodes) satu sama lain dan jenis hubungannya. Simbol-simbol *Deployment Diagram* ditunjukkan oleh Tabel 2.6 berikut.

Tabel 2.6 Simbol-simbol Deployment Diagram

No	Gambar	Nama	Keterangan
1		Node	Biasanya mengacu pada perangkat keras (<i>hardware</i>) dan perangkat lunak (<i>software</i>), jika di dalam node disertakan komponen untuk menkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang sudah didefinisikan sebelumnya pada diagram komponen.
2		Depedency	Sebuah dependency digunakan untuk menotasikan relasi antara dua component.
3		Link	Relasi antar node.

2.3.8. Pengujian

Pengujian adalah suatu proses pelaksanaan suatu program dengan tujuan menemukan suatu kesalahan. Suatu kasus test yang baik adalah apabila test tersebut mempunyai kemungkinan menemukan sebuah kesalahan yang tidak terungkap.

1. Pengujian Alpha

Pengujian fungsional yang digunakan untuk menguji sistem yang baru adalah metode pengujian alpha. Pengujian alpha dilakukan dengan menggunakan metode *black box*. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak.

Black box testing merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, *tester* dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetesan pada spesifikasi fungsional program (khan, 2011).

Black Box cenderung untuk menemukan hal-hal berikut :

- a. Fungsi yang tidak benar atau tidak ada.
- b. Kesalahan antarmuka (*interface errors*)
- c. Kesalahan pada struktur data dan akses basis data
- d. Kesalahan performansi (*performance errors*)
- e. Kesalahan inisialisasi dan terminasi

2. Pengujian Beta

Pengujian beta merupakan suatu bentuk pengujian yang dilakukan oleh pengguna, dengan membuat kuisisioner tentang aplikasi yang dibuat (Salamah dan Khasanah, 2017).

Adapun metode penilaian pengujian yang digunakan yaitu metode kuantitatif berdasarkan data sampel dari user. Untuk mengetahui tanggapan dan penilaian pengguna terhadap sistem ini, disebarkan kuisisioner kepada responden berdasarkan target *user* dan dilakukan prosentase dengan menggunakan rumus sebagaimana persamaan 2.1.

$$Y = P/Q * 100\% \quad (2.1)$$

Keterangan :

P= Banyaknya jawaban responden tiap soal

Q= Jumlah responden

Y= Nilai prosentase