

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian ini menggunakan tiga kajian terkait yang untuk mendukung pengumpulan data yang akan dilakukan. (Taupik et al., 2017) melakukan penelitian "Aplikasi Berbasis Web Inventarisasi Aset Desa Sukapura". Dalam penelitian tersebut aplikasi inventarisasi aset ini digunakan untuk membantu pemerintahan desa dalam melakukan kelola di aset tanpa lagi menggunakan Microsoft excel dan membutuhkan banyak file untuk melakukan kelola aset.

(Ashari & Juaini, 2018) melakukan penelitian "Sistem Informasi Pengolahan Data Inventaris Dan Pengadaan Barang Pada Kantor Desa Lenteng Berbasis Web". Dalam penelitian tersebut mereka membuat aplikasi inventaris yang bertujuan untuk mengurangi kesalahan-kesalahan yang terjadi pada saat pencatatan persediaan barang, transaksi permintaan barang, dan juga dapat menyediakan informasi yang akurat dan bisa dijadikan sebagai landasan bagi pimpinan instansi pemerintahan, dalam mengambil keputusan untuk kepentingan instansi pemerintahan kedepannya.

(Wiratama et al., 2022) Melakukan penelitian yang berjudul "Implementasi Sistem Informasi Inventaris Pada Kantor Desa Ketewel". Dalam penelitian tersebut bertujuan untuk membangun sebuah sistem informasi inventaris pada Kantor Desa Ketewel berbasis web yang meliputi data barang masuk, data barang keluar, pengajuan inventarisasi, jenis stocktaking, kamar, perhitungan, penyusutan dan berupa laporan barang masuk, laporan barang keluar, laporan keseluruhan dan laporan penyusutan.

Berdasarkan penelitian terkait yang sudah dipaparkan di atas, maka menyimpulkan bahwa dari ke-3 studi literatur tersebut pada intinya prinsip proses sistem yang dibuat itu sama, karena berkaitan dengan pendataan inventaris atau inventarisasi aset dan perbedaan sistem yang dibuat adalah terletak pada adanya master barang sebagai pusat data untuk mendukung proses pendataan yang mana aset-aset yang diinventarisasikan itu adalah aset milik desa yang mencakup seluruh desa matang danau, belum lagi adanya pengelolaan aset desa yang

bertujuan untuk peminjaman aset. Proses tersebut membuat sistem informasi ini berbeda dari ke-3 studi literatur yang sudah dipaparkan sebelumnya diatas. Dengan demikian, pada penelitian ini akan membangun sebuah sistem informasi inventaris desa berbasis web yang bertujuan untuk menghasilkan sebuah sistem yang dapat mempermudah proses pendataan aset atau inventarisasi aset, serta pengelolaan peminjaman aset oleh masyarakat desa matang danau kecamatan paloh kabupaten sambas.

2.2 Profil Desa Matang Danau

Desa Matang Danau merupakan desa yang terletak di Kecamatan Paloh, Kabupaten Sambas, Kalimantan Barat, Indonesia. Luas wilayahnya sebesar 4.401 Ha meliputi 3,83% wilayah kecamatan Paloh yang mana dari luas tersebut terdiri dari lahan pertanian 750 Ha, perumahan dan pekarangan 1,5 Ha, lahan perkebunan 1,75 Ha, tanah wakaf 3.339 Ha. Desa Matang Danau dipimpin oleh seorang kepala desa yang bernama Halipi. Secara administratif, wilayah Desa Matang Danau terbagi menjadi 4 (empat) dusun yang terdiri dari Dusun Pantai Laut, Dusun Mariana, Dusun Perigi Nyatu, Dusun Matang Putus serta secara keseluruhan terdapat 10 rukun warga dan 21 rukun tetangga. Desa Matang Danau dulunya merupakan penggabungan 2 (dua) desa, yang mana sekarang ini menjadi Dusun Matang Putus itu adalah Desa Matang Putus tersendiri. Kemudian setelah paska kesuruhan, sebagian penduduk pindah ke daerah lain. Akhirnya, Desa Matang Putus Menjadi dusun dan bergabung dengan Desa Matang Danau.

Pada tahun 2020, Desa Matang Danau diklasifikasikan sebagai desa berkembang. Jumlah penduduk Desa Matang Danau pada tahun 2020 sebanyak 4.601 jiwa diantaranya 2.368 laki-laki dan 2.233 perempuan. Hal tersebut menempatkan Desa Matang Danau sebagai desa berpenduduk terbanyak kedua di Kecamatan Paloh setelah Desa Sebusus. Kepadatan penduduk di Desa Matang Danau pada tahun 2020 adalah 105 jiwa yang menjadikan sebagai desa berpendudukan terpadat di Kecamatan Paloh. Mata pencaharian Desa Matang Danau sebagian besar adalah Petani atau Nelayan dan ada juga berprofesi lain sebagai PNS/TNI/POLRI/HONORER, Usaha Industri Sedang/Kecil, Pedangan, Swasta, Jasa, Perkebunan, Buruh Bangunan dan Merantau. Di Desa Matang

Danau terdapat beberapa jejang pendidikan mulai dari taman kanak-kanak, sekolah dasar negeri, madrasah ibtidaiyah, madrasah tsanawiyah, sekolah menengah pertama yang tersebar seluruh desa (Su'udi, 2021).

2.3 Inventaris

Inventarisasi merupakan proses mengelola pengadaan atau persediaan barang yang dimiliki oleh kantor atau perusahaan dalam melakukan kegiatan operasionalnya. Tanpa adanya inventaris suatu kegiatan usaha tidak akan terlaksana, untuk itu keberadaan inventaris sangat penting.

Inventaris kantor sangatlah penting bagi kelangsungan sebuah instansi. Apabila salah satu atau beberapa perlengkapan mengalami gangguan, maka pasti akan menghambat jalannya roda perekonomian perusahaan yang biasanya berupa tidak teraturnya keorganisasian inventaris kantor atau kurangnya sebuah sistem dalam menginventarisasi perlengkapan kantor (andi, 2007).

(Arifin, M. Barnawi 2012) menerangkan bahwa “Inventarisasi adalah kegiatan pencatatan atau pendaftaran barang-barang secara tertib dan teratur untuk keperluan kepengurusan dan pencatatan ini harus disediakan instrument administrasi antara lain buku penerimaan barang, buku, pembelian barang, buku induk inventaris, buku golongan inventaris, buku bukan inventaris, buku stok barang”.

2.4 Aset Desa

Menurut Pernyataan Standar Akuntansi Pemerintahan (PSAP), aset adalah sumber daya ekonomi yang dikuasai dan/atau dimiliki oleh pemerintah sebagai akibat dari peristiwa masa lalu dan dari manfaat ekonomi dan/atau social di masa depan diharapkan dapat diperoleh, baik oleh pemerintahan maupun masyarakat, serta dapat diukur dalam satuan uang.

Siregar (2016) mengemukakan bahwa aset desa adalah sesuatu yang mempunyai nilai tukar, modal atau kekayaan. Dalam hal ini, pengertian aset desa sama maknanya dengan konsep kekayaan. Aset desa sama pengertiannya dengan kekayaan desa sebagaimana disebutkan dalam berbagai regulasi pemerintahan yang mengatur tentang desa, meskipun tidak terbatas pada kekayaan bersifat fisik.

Pasal 1 angka 11 Undang-Undang Nomor 6 Tahun 2014 Tentang Desa

menyebutkan bahwa “Aset Desa adalah barang milik desa yang berasal dari kekayaan asli desa, dibeli atau diperoleh atas beban anggaran pendapatan dan belanja desa atau perolehan hak lainnya yang sah”. Dari beberapa pengertian diatas bahwa aset desa merupakan segala sesuatu yang dimiliki oleh desa, dikelola oleh desa, dan digunakan untuk desa demi yang tercapainya tujuan Negara yaitu mensejahterakan masyarakat.

Menurut Peraturan Menteri dalam Negeri No 1 Tahun 2016 tentang pengelolaan aset desa dalam pasal 2 menyebutkan bahwa jenis-jenis aset desa terdiri atas : kekayaan asli desa, kekayaan milik desa yang dibeli atau diperoleh atas beban APBDesa, kekayaan desa yang diperoleh dari hibah dan sumbangan atau yang sejenis, kekayaan desa yang diperoleh sebagai pelaksanaan dari perjanjian/kontrak dan/ atau diperoleh berdasarkan ketentuan peraturan undang-undang, hasil kerja sama desa dan, kekayaan desa yang berasal dari perolehan lainnya yang sah.

Kekayaan asli desa itu sendiri dibagi lagi menjadi 11 macam yakni menurut PERMENDAGRI tersebut salah satunya adalah tanah kas desa, selain tanah kas desa ada juga pasar desa, pasar hewan, tambatan perahu, bangunan desa, pelelangan ikan yang dikelola oleh desa, pelelangan hasil pertanian, hutan milik desa, mata air milik desa, pemandian umum, dan lain-lainnya kekayaan asli desa.

2.5 Sistem Informasi

Sistem informasi mencakup sejumlah komponen terdiri dari manusia, komputer, teknologi informasi, dan prosedur kerja, ada sesuatu yang diproses yaitu data menjadi informasi dan dimaksudkan untuk mencapai suatu tujuan tertentu. Ada beberapa pengertian sistem informasi menurut para ahli, diantaranya: Menurut Gelinas, Oram, dan Wiggins, sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan, dan mengelola data serta menyediakan informasi kepada pengguna. Sedangkan menurut Hall, sistem informasi adalah sekumpulan rangkaian prosedur formal dimana data dikelompokkan, diproses, menjadi informasi, dan didistribusikan kepada pengguna.

Menurut John F. Nash dan Martin B. Roberts (1984) sistem adalah kumpulan komponen yang berinteraksi membentuk suatu kesatuan dan keutuhan yang kompleks di tingkat tertentu untuk mengejar tujuan yang umum. Menurut (Davis, 1997) informasi adalah data yang telah diolah ke dalam suatu bentuk yang penting bagi penerimanya dan nyata atau berupa nilai yang dapat dipahami di dalam keputusan sekarang maupun masa depan. Menurut (Jogiyanto H.M., 1997) Kualitas dari suatu informasi (*quality of information*) tergantung dari tiga hal, yaitu informasi harus akurat (*accurate*), tepat pada waktunya (*timely basis*) dan relevan (*relevance*). Menurut John F. Nash dan Martin B. Roberts (1984) suatu sistem informasi adalah suatu kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar untuk pengambilan keputusan yang cerdas.

2.6 Website

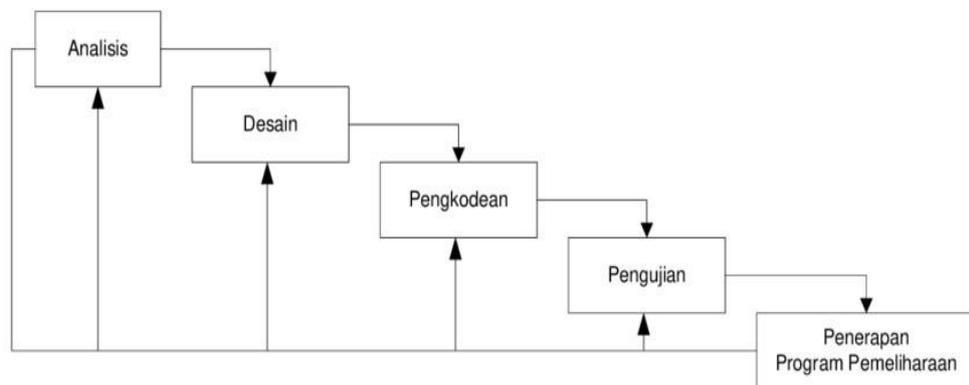
Website merupakan suatu dokumen yang berupa kumpulan halaman yang saling terhubung dan isinya terdiri dari berbagai informasi yang berbentuk teks, suara, gambar, video, dan lain sebagainya, dimana semua data tersebut disimpan pada server hosting. Untuk membuka sebuah website maka pengguna harus memiliki perangkat (komputer maupun smartphone) yang terkoneksi dengan internet. *Website* pada umumnya berbentuk dokumen dalam format *Hyper Text Markup Language* (HTML), yang dapat diakses melalui HTTP atau HTTPS, suatu protokol yang menyampaikan berbagai informasi dari server web untuk ditampilkan kepada para user atau pemakai melalui web browser. Sebuah website memiliki alamat URL yang unik dan spesifik yang disebut dengan domain. Misalnya domain Google.com, Facebook.com, dan lain-lain.

World Wide Web (WWW) atau biasa disebut dengan web merupakan salah satu sumber daya Internet yang berkembang pesat. Pertama kali sistem web dibangun hanya dengan menggunakan bahasa yang disebut HTML (*HyperText Markup Language*) dan protokol yang digunakan dinamakan HTTP (*HyperText*

Transfer Protocol). Pada perkembangan berikutnya, sejumlah skrip dan objek dikembangkan untuk memperluas kemampuan HTML yang sekarang ini terdapat banyak skrip seperti: PHP dan ASP, sedangkan contoh yang berupa objek antara lain adalah applet (*java*). Jadi sistem website atau sistem berbasis web (*Web-based application*) adalah sistem untuk menyampaikan informasi kepada pengguna yang menggunakan layanan Internet berbasis web. (Kadir, 2003).

2.7 Model Waterfall

Model Waterfall merupakan model yang bersifat sistematis dan termasuk dalam model klasik. Menurut Sommerville (2011) metode waterfall memiliki lima tahapan, yaitu analisis (*analysis*), perancangan (*design*), pengkodean (*coding*), pengujian (*testing*) dan pemeliharaan (*maintenance*). Tahapan metode waterfall dapat dilihat pada Gambar 2.1.



Gambar 2.1 Tahapan Metode Waterfall

Penjelasan tahapan-tahapan model waterfall tersebut yaitu:

1. Analysis (Analisis)

Fase ini merupakan proses analisa terhadap sistem yang sedang berjalan dengan tujuan untuk mendapatkan jawaban mengenai pengguna sistem, cara kerja sistem dan waktu penggunaan sistem, sehingga kebutuhan yang diperlukan untuk sistem baru akan didapatkan.

2. Design (Perancangan)

Perancangan merupakan proses penentuan cara kerja sistem dalam hal perancangan antarmuka, database, dan perancangan alur

program. Perancangan diperlukan untuk menggambarkan sistem baru dengan tujuan memenuhi kebutuhan pengguna.

3. Coding (pengkodean)

Tahapan coding merupakan implementasi rancangan sistem yang dibentuk menjadi suatu kode program untuk pembuatan sistem.

4. Testing (Pengujian)

Pengujian program dilakukan untuk mengetahui kesesuaian sistem berjalan sesuai prosedur atau tidak dan memastikan sistem terhindar dari error yang terjadi. Testing juga dilakukan untuk memastikan kevalidan dalam proses input sehingga dapat menghasilkan output yang sesuai.

5. Maintenance (Pemeliharaan)

Fase ini yaitu pemeliharaan dan pengembangan sistem yang berguna untuk melihat kemampuannya, mengecek jika masih ada ditemukan error atau ada penambahan fitur-fitur yang belum ada pada sistem tersebut. Pengembangan diperlukan ketika adanya perubahan dari pengguna seperti ketika ada pergantian sistem operasi, atau perangkat lainnya.

2.8 PHP (*Hypertext Preprocessing*)

PHP adalah bahasa pemrograman untuk sisi server yang dirancang secara khusus untuk web. 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. PHP bersifat *open source* sehingga dapat dipakai secara gratis dan mampu lintas *platform*, yaitu dapat berjalan pada sistem operasi Windows maupun Linux. PHP juga dibangun sebagai modul pada *web server Apache* dan sebagai *binary* yang dapat berjalan sebagai CGI.

Menurut Welling & Thomson (2003) Dalam halaman HTML dapat dimasukan kode-kode PHP yang akan dijalankan setiap kali halaman tersebut dieksekusi, kode-kode PHP akan diinterpretasikan pada server web dan menghasilkan HTML atau Output lainnya yang akan dilihat oleh pengunjung web.

2.9 XAMPP

XAMPP merupakan paket *PHP* dan *MySQL* berbasis *open source*, yang digunakan sebagai alat pembantu pengembangan sistem berbasis *PHP*. *XAMPP* mengkombinasikan beberapa paket perangkat lunak berbeda dalam satu paket (Riyanto, 2010).

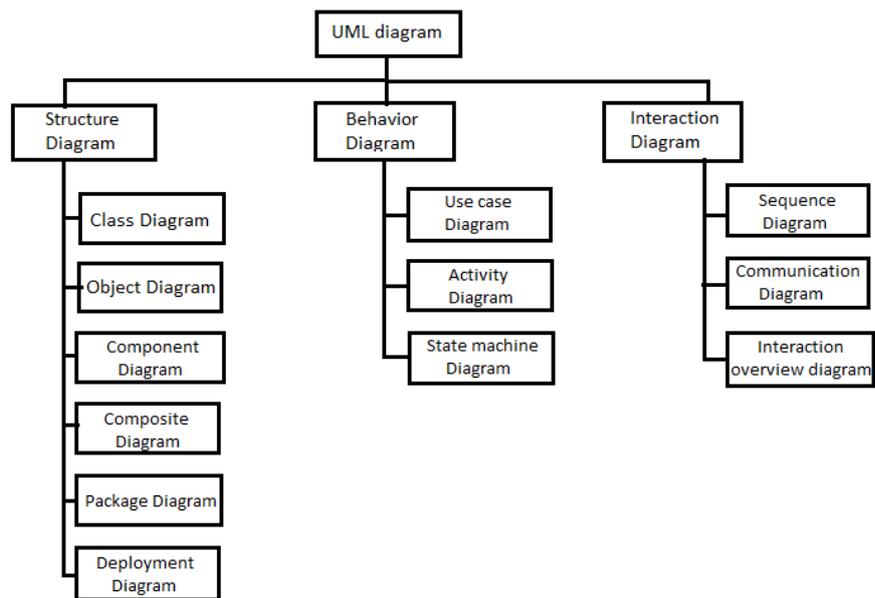
Di dalam *XAMPP* terdapat tiga paket penting yaitu *Apache* sebagai *web server*, *PHP* sebagai bahasa pemrograman dan *MySQL* sebagai *database*. *Apache* adalah *server web (web server)* yang dapat dijalankan di banyak sistem operasi. *Apache* merupakan perangkat lunak *open source* yang dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan *Apache Software Foundation*.

2.10 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah salah satu standar bahasa visual yang banyak digunakan di dunia industri untuk mengidentifikasi *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML hanya berfungsi untuk melakukan pemodelan, jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek (Rosa A. S dan M. Salahuddin, 2013).

UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat.

UML terdapat 13 diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada Gambar 2.3 dibawah ini :



Gambar 2.2 Klasifikasi Diagram UML

Berikut penjelasan singkat dari Gambar 2.2.

1. *Structure Diagram*, yaitu kumpulan diagram-diagram yang menggambarkan struktur statis dari sistem yang dimodelkan.
2. *Behavior Diagram*, yaitu kumpulan diagram-diagram yang menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi di dalam sistem.
3. *Interaction Diagram*, yaitu kumpulan diagram-diagram yang menggambarkan interaksi sistem dengan sistem lain ataupun interaksi antar subsistem dalam sebuah sistem.

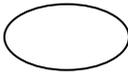
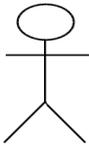
Secara garis besar, beberapa diagram utama sudah dapat menggambarkan keseluruhan sistem. Diagram tersebut antara lain *use case diagram*, *class diagram*, *sequence diagram*, dan *activity diagram*.

2.11 Use Case Diagram

Use Case Diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu

(Sukamto dan Shalahuddin, 2013). Berikut adalah simbol-simbol yang ada pada diagram *use case*.

Tabel 2.1 Simbol Diagram Use Case

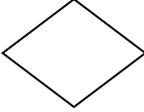
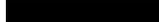
No.	Notasi	Nama	Deskripsi
1.		<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
2.		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.		<i>Association</i>	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau <i>use case</i> memiliki interaksi dengan aktor.
4.	<p><<extend>> -----></p>	<i>Extended</i> (Ekstensi)	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

No.	Notasi	Nama	Deskripsi
			ditambahkan.
5.	<pre><<include>> <-----</pre>	<i>Include</i> (Menggunakan)	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan.
6.		<i>Generalization</i> (Generalisasi)	Hubungan generalisasi dan spesialisasi (umum - khusus) antara dua buah <i>use case</i> dimana fungsi yang satu merupakan fungsi yang lebih umum dari lainnya. Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)

2.12 Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) proses bisnis dan urutan aktivitas dalam sebuah proses. *Activity diagram* sangat mirip dengan *flowchart* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status. Membuat *activity diagram* pada awal pemodelan proses cukup menguntungkan untuk membantu memahami keseluruhan proses. *Activity diagram* juga bermanfaat untuk menggambarkan *parallel behavior* atau menggambarkan interaksi antara beberapa *use case*. Berikut merupakan simbol notasi *Activity Diagram* pada Tabel 2.3 (Rosa A. S dan M. Salahuddin, 2013).

Tabel 2.2 Simbol *Activity Diagram*

No	Notasi	Nama	Deskripsi
1		Status Awal (<i>initial node</i>)	Status awal aktivitas, sebuah diagram aktivitas memiliki status awal.
2		Status Akhir (<i>final node</i>)	Status akhir yang dilakukan sistem.
3		Aktivitas (<i>activity</i>)	Aktivitas yang dilakukan oleh sistem, biasanya diawali oleh kata kerja.
4		Percabangan (<i>decision</i>)	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
5		Penggabungan (<i>join</i>)	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
6		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.13 Sequence Diagram

Sequence diagram menggambarkan alur objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Oleh karena itu, untuk menggambarkan *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu (Sukamto dan Shalahuddin, 2013). Berikut adalah simbol-simbol yang ada pada *sequence diagram*.

Tabel 2.3 Simbol *Sequence Diagram*

No	Notasi	Nama	Deskripsi
1		<i>Actor</i>	Orang Proses, atau Sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat. Biasanya dinyatakan dengan menggunakan kata benda di nama aktor.
2		<i>Object</i>	Menyatakan objek yang berinteraksi pesan.
3		Garis Hidup	Menyatakan kehidupan suatu objek.

2.14 *Class Diagram*

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem (Rosa A. S dan M. Salahuddin, 2013). *Class diagram* memiliki apa yang disebut atribut dan operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Atribut dan metode dapat memiliki salah satu sifat sebagai berikut:

1. *Private* (-), hanya dapat digunakan oleh *class* yang memilikinya
2. *Public* (+), dapat digunakan oleh *class* lain.
3. *Protected* (#), hanya dapat dipanggil oleh *class* yang bersangkutan jumlah suatu anak yang mewarisinya.

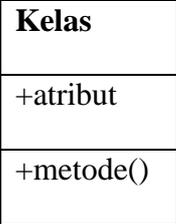
Nilai kardinalitas atau *multiplicity* sebuah *class* menunjukkan jumlah suatu objek yang dapat berhubungan dengan objek lain. Berikut nilai kardinalitas atau *multiplicity* pada Tabel 2.5 (Tohari, Hamim 2014) dan notasi *class diagram* pada Tabel 2.6 (Rosa A. S dan M. Salahuddin, 2013).

Tabel 2.4 Jenis-jenis *Multiplicity*

No	Indikator	Keterangan
1	0 .. 1	Nol atau satu
2	1	Hanya satu
3	0 .. *	Nol atau lebih
4	1 .. *	Satu atau lebih

Berikut adalah simbol-simbol yang ada pada *class diagram*.

Tabel 2.5 Simbol *Class Diagram*

No	Notasi	Nama	Deskripsi
1		<i>Class</i>	Menggambarkan konsep dasar pemodelan sistem.
2		Asosiasi (<i>Assosiation</i>)	Sebuah garis solid antara dua <i>class</i> , ditarik dari <i>class</i> sumber ke <i>class</i> target lebih spesifik, digunakan dalam struktur pewarisan.
3		Ketergantungan (<i>Dependency</i>)	Relasi antara dua elemen jika perubahan definisi sebuah elemen (<i>supplier</i> atau sumber) dapat menyebabkan perubahan pada elemen lainnya (<i>Client</i> atau target).

2.15 Pengujian Sistem

Pengujian adalah proses menjalankan suatu program dengan tujuan

menemukan kesalahan (*error*). Berdasarkan pengertian tersebut, aktivitas yang terjadi dalam pengujian perangkat lunak terdiri dari pengujian kode program hingga kegiatan percobaan terhadap perangkat lunak yang sudah berfungsi.

Pengujian merupakan suatu teknik yang digunakan menguji apakah sebuah perangkat lunak yang dihasilkan telah sesuai dengan yang diharapkan atau belum. Menurut Pressman (2002), pengujian adalah proses eksekusi suatu program untuk menemukan kesalahan sebelum digunakan oleh pengguna akhir (*end-user*). Pengujian bertujuan untuk memastikan hasil implementasi yang telah dilakukan apakah sudah memenuhi semua kebutuhan dan apakah sistem yang telah dibangun dapat berfungsi sesuai dengan yang diharapkan sebelumnya.

2.15.1 Pengujian *Black Box*

Black Box testing berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineers* untuk memperoleh set kondisi input yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program (Pressman, 2010).

Black Box Testing berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengujian 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 (Mustaqbal dan Firdaus, 2015)

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

2.15.2 Pengujian *System Usability Scale (SUS)*

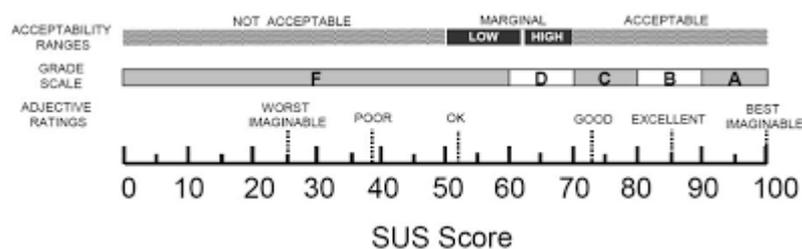
Usability adalah analisa kualitatif yang menentukan seberapa mudah user menggunakan antarmuka suatu aplikasi (Nielsen, 2012). Suatu aplikasi disebut *usable* jika fungsi-fungsinya dapat dijalankan secara efektif, efisien, dan memuaskan. Secara umum *Usability* merupakan tingkat kualitas dari suatu

sistem yang mudah digunakan, mudah dipelajari dan mendorong pengguna untuk menggunakan sistem sebagai alat bantu dalam menyelesaikan suatu tugas atau pekerjaan, hal ini berpengaruh pada *user experience* (UX). Pengujian ini dilakukan dengan harapan mencapai tujuan dimana pengguna mendapatkan pengalaman yang baik selama menggunakan website. Pengujian Usability dijadikan sebagai tolak ukur kualitas pengalaman pengguna ketika berinteraksi dengan suatu sistem seperti website, android maupun aplikasi perangkat lunak lainnya yang dioperasikan oleh pengguna. *System Usability Scale* (SUS) merupakan kuesioner yang dapat digunakan untuk mengukur usability sistem komputer menurut sudut pandang subyektif pengguna (Brooke, 2013). SUS diciptakan oleh John Brooke pada tahun 1986 dan dahulu digunakan untuk menguji sistem elektronik kantor. SUS bersifat *quick and dirty*. Sistem ini menggunakan 10 pertanyaan dengan tanggapan berupa lima poin skala *Likert* untuk masing-masing pertanyaan (Brooke, 1996). Penilaian dari responden berupa skala 1 yang berarti “Sangat Tidak Setuju”, sampai skala 5 yang berarti “Sangat Setuju”. Setiap item pertanyaan memiliki skor kontribusi antara 0 sampai 4. Dalam pengolahannya, *sistem usability scale* memiliki tata perhitungan tersendiri diantaranya untuk item bernomor ganjil (1,3,5,7,9) skor kontribusinya tersebut dikurangi 1. Sedangkan untuk bernomor genap (2,4,6,10) skor kontribusinya tersebut 5 dikurangi nilai item. Total dari SUS tersebut dikalikan 2.5 sehingga mendapatkan hasil yang maksimal. Hasil dari SUS berupa skor antara 0 sampai 100, dengan semakin besar skor SUS yang dihasil berarti semakin baik *usability*-nya. Berikut ini merupakan instrumen kuisisioner *sistem usability score* pada gambar 2.3.

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	<input type="checkbox"/>				
	1	2	3	4	5
2. I found the system unnecessarily complex	<input type="checkbox"/>				
	1	2	3	4	5
3. I thought the system was easy to use	<input type="checkbox"/>				
	1	2	3	4	5
4. I think I would need the support of a technical person to be able to use this system	<input type="checkbox"/>				
	1	2	3	4	5
5. I found the various functions in this system were well integrated	<input type="checkbox"/>				
	1	2	3	4	5
6. I thought there was too much inconsistency in this system	<input type="checkbox"/>				
	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	<input type="checkbox"/>				
	1	2	3	4	5
8. I found the system very cumbersome to use	<input type="checkbox"/>				
	1	2	3	4	5
9. I felt very confident using the system	<input type="checkbox"/>				
	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	<input type="checkbox"/>				
	1	2	3	4	5

Gambar 2.3 *The Sistem Usability Scale Questionnaire*

Kemudian hasil dari skos SUS akan diinterpretasikan dengan ukuran *acceptable* atau *not acceptable*. Untuk menetapkan syarat suatu sistem masuk dalam kriteria *acceptable* skor lebih dari 70, *marginal high* 64-70, *marginal low* 51-63, dan skor kurang dari 51 dinyatakan *not acceptable* (Brooke, 2013). Berikut merupakan interpretasi skors SUS pada gambar .



Gambar 2.4 *Sistem Usability Scale Score*