BAB II
TINJAUAN PUSTAKA

2.1 Literatur Review

Kumpulan jurnal dan penelitian lain yang berkaitan dengan penelitian yangakan di lakukan dapat di lihat pada tabel 2.1 berikut:

Tabel 2.1 Literatur Review

No	Nama Penulis	Metode	Aplikasi
1	Yulanda. (2019)	Data masuk, Rekap, catat jumlah laporan masuk sesuai pegawai, hitung, waterfall, web, PHP dan MySQL	Penilaian kinerja pegawai PDAM
2	Rika Rahmawati. (2018)	Kumpulkan dan rekap laporan,di simpulkan menjdi sebuah lporan utama, waterfall	Pelaporan kinerja pegawai
3	Fahmi Hakam , Noor Alis Setiyadi. (2014)	Pengumpulan data sesuai jenis data, satukan sesuai dengan jenis data, prototype.	Pengembangan sistem pencatatan dan pelaporan data di klinik Muhamadiyah

2.2 Sumber Daya Manusia

Laporan kerja sangat lah penting, namun mengumpulkan dokumentasi di berbeda area dan menyusun laporan di setiap bulan nya tidaklah mudah, maka dari itu perlu adanya dokumentasi yang akurat, akurat adalah salah satu hal yang sangat baik hubungan nya antar konsumen dengan perusahaan.

Sumber Daya Manusia merupakan motor penggerak dan aset dari sebuah perusahaan. Tanpa adanya SDM, organisasi tidak akan maju sesuai dengan yang direncanakan. Menurut (Sadili Samsudin 2010: 1) SDM atau sumber daya manusia adalah orang-orang yang merancang dan menghasilkan barang atau jasa, mengawasi mutu, memasarkan produk, mengalokasikan sumber daya finansial, serta merumuskan seluruh strategi dan tujuan organisasi. Sehingga dapat diartikan bahwa, SDM adalah aset yang dimiliki bank untuk melakukan segala aktivitas operasional bank. Dalam tugasnya, SDM diarahkan oleh sebuah manajemen yakni Manajemen Sumber Daya Manusia.

Manajemen sumber daya manusia dapat diartikan sebagai pendayagunaan sumber daya manusia di dalam organisasi, yang dilakukan melalui fungsi-fungsi perencanaan sumber daya manusia, rekrutmen dan seleksi, pengembangan sumber daya manusia, perencanaan dan pengembangan karir, pemberian kompensasi dan kesejahteraan, keselamatan dan kesehatan kerja, dan hubungan industrial (Marwansyah, 2010:3). Sehingga dapat disimpulkan bahwa, MSDM adalah suatu langkah yang dilakukan Menajer dalam mengarahkan SDM nya agar kinerja yang dihasilkan sesuai dengan standar dan mencapai tujuan perusahaan.

2.3 Manajemen Kinerja

Kata Manajemen Kinerja merupakan penggabungan dari kata manajemen dan kinerja. Manajemen berasal dari kata *to manage* yang berarti mengatur. Menurut (George R Terry) dalam bukunya *Principles of Management*, Manajemen merupakan suatu proses yang menggunakan metode ilmu dan seni untuk menerapkan fungsi-fungsi perencanaan, pengorganisasian, pengarahan dan pengendalian pada kegiatan-kegiatan dari sekelompok manusia yang dilengkapi dengan sumber daya/faktor produksi untuk mencapai tujuan yang sudah ditetapkan lebih dahulu, secara efektif dan efisien. Sedangkan menurut (John R Schermerhorn Jr) dalam bukunya Management, manajemen adalah proses yang mencakup perencanaan,

pengorganisasian, pengarahan dan pengendalian terhadap penggunaan sumber daya yang dimiliki, baik manusia dan material untuk mencapai tujuan.

Dari beberapa definisi manajemen yang diberikan oleh para ahli, dapat disimpulkan manajemen mencakup tiga aspek, yaitu:

1. Pertama : manajemen sebagai proses
2. Kedua : adanya tujuan yang telah ditetapkan
3. Ketiga : mencapai tujuan secara efektif dan efisien

Kata kinerja merupakan singkatan dari kinetika energi kerja yang padanannya dalam bahasa Inggris adalah *performance,* yang sering diIndonesiakan menjadi kata performa. Dengan demikian, kinerja adalah kesediaan seseorang atau kelompok orang untuk melakukan sesuatu kegiatan dan menyempurnakannya sesuai dengan tanggung jawabnya dengan hasil seperti yang diharapkan. Dari kedua kata manajemen dan kinerja, jika digabungkan menjadi satu kata baru yaitu Manajemen Kinerja (*Performance Management*). Manajemen kinerja merupakan suatu cara untuk mendapatkan hasil yang lebih baik bagi organisasi kelompok dan individu dengan memahami dan mengolah kinerja sesuai dengan target yang telah di rencanakan dengan standar dan persyaratan kompetensi yang telah ditentukan (Tohari. 2002). Manajemen kinerja bertujuan untuk menciptakan budaya pada individu dan kelompok dalam memikul tanggung jawab bagi usaha dalam meningkatkaan proses kerja dan kemampuan yang berkesinambungan dengan menggunakan presensi sebagai salah satu standar utama penilaian kedisiplinan.

2.4 Laporan Kerja

Laporan adalah bentuk penyajian dari suatu fakta mengenai hal yang berkenaan terhadap keadaan ataupun suatu kegiatan. Dan pada dasarnya suatu fakta yang disajikan tersebut ialah tanggung jawab yang ditugaskan bagi si pelapor.

Fakta yang disajikan merupakan bahan ataupun keterangan dari informasi yang memang dibutuhkan. Berdasarkan dari suatu objektif yang dialami oleh si pelapor atau dilihat, didengar, dirasakan oleh pelapor. Dan pada saat si pelapor sudah melaksanakan kegiatan atau suatu kegiatan.

2.4.1 Fungsi Laporan

Berikut ini adalah beberapa fungsi laporan yakni sebagai berikut:

1. Adanya pertanggung jawaban bagi orang yang diberi tugas (Pelapor)
2. Menjadi landasan pimpinan dalam mengambil kebijakan/keputusan
3. Menjadi alat untuk melakukan pengawasan
4. Menjadi dokumen sebagai bahan studi dan pengalaman bagi orang lain.

2.4.2 Manfaat Laporan

Didalam sebuah laporan tentunya memiliki manfaat didalamnya, berikut iniadalah manfaat dari sebuah laporan diantaranya :

1. Menjadi dasar penentuan kebijakan
2. Menjadi bahan untuk penyusunan rencana kegiatan-kegiatan berikutnya
3. Bisa mengetahui perkembangan dan proses dari peningkatan kegiatan
4. Sebagai sumber informasi

2.4.3 Macam-macam Laporan

Berikut ini macam-macam laporan menurut bentuknya*:*

1. Laporan dalam bentuk formulir
2. Laporan dalam bentuk surat
3. Laporan dalam bentuk memorandum (memo)

4. Laporan dalam bentuk naskah

5. Laporan dalam bentuk buku

2.5 Smartphone

Smartphone adalah suatu mobile *device* yang tertanam sistem operasi sehingga memiliki kemampuan tingkat tinggi dan memiliki kemampuan menyerupai komputer (wikipedia.org). *Smartphone* adalah kombinasi antara telepon genggam, *Personal Digital Assistant* (PDA), media *player, digital camera* dan *Global Positioning System*(GPS). Selain fitur-fitur tersebut, didalam *smartphone* tertanam pula *Bluetooth* dan Wi-Fi serta kemampuan untuk mengakses internet dengan baik.

Smartphone merupakan *mobile device* untuk masa depan yang menawarkan berbagai fungsi kemudahan dalam kemampuan nirkabel, daya komputasi dan penyimpanan *on board.* Sehingga secara tidak langsung *smartphone* memiliki kemampuan untuk mempermudah kegiatan manusia.

2.6 Aplikasi

Aplikasi merupakan perangkat lunak proses data yang berpacu pada sebuah komputasi. Aplikasi berasal dari bahasa inggris *application* yang berarti penerapan penggunaan. Sedangkan secara istilah, pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Sedangkan menurut kamus besar Bahasa Indonesia (2005: 52),—Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Adapun definisi aplikasi menurut para ahli adalah:

1. Aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*)atau pernyataan (*statement)* yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output* (Jogiyanto, 2005).

2. Aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu (Hendrayudi, 2005).
3. Aplikasi adalah suatu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti sistem perniagaan, game palayanan masyarakat, periklanan atau semua peroses yang hampir dilakukan manusia (Hengky W. Pramana, 2005). Dari pengertian diatas, dapat disimpulkan bahwa aplikasi merupakan *software* yang ditransformasikan ke komputer yang berisikan perintah-perintah yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data.

2.7 Android

Android adalah sebuah sistem operasi untuk perangkat mobile yang mencakup aplikasi sistem operasi, *middleware* dan aplikasi. Android SDK menyediakan fitur dan API yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android dengan menggunakan bahasa pemrograman Java. Menurut (DiMarzio 2008), Android sebagai suatu sistem operasi berbasis Java yang berjalan pada kernel Linux 2.6. Sistem ini sangat ringan dan berfitur lengkap. Android dikembangkan oleh *Open Handset Alliance*, Android membawa inovasi dengan *internet* dan keterbukaan terhadap perangkat *mobile*. Android memberikan sebuah *software* yang lengkap untuk aplikasi pada telepon seluler: sistem operasi, *middleware* dan aplikasi kunci dalam *mobile.*

2.8 Manajemen Sistem

IT *System Management* merupakan disiplin ilmu mengenai pengelolaan entitas IT yaitu, orang, proses, dan teknologi yang saling berinteraksi satu sama lain serta berkembang dalam suatu lingkungan yang disebut infrastruktur teknologi informasi.

2.9 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL(bahasa Inggris: *database management system*) atau DBMS yang multi alur, multi pengguna, dengan sekitar 6 juta instalasi di seluruh dunia. *MySQL* AB membuat *MySQL* tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan GPL.
MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan *MySQL*, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial.
SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Kehandalan suatu sistem basisdata (DBMS) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, *MySQL* mendukung operasi basis data transaksional maupun operasi basis data non-transaksional. Pada modus operasi non-transaksional, *MySQL* dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basis data *competitor* lainnya. Namun pada modus non- transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi *blogging* berbasis web (*wordpress*), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basis data transaksional, hanya saja sebagai konsekuensinya unjuk kerja *MySQL* pada modus transaksional tidak secepat unjuk kerja pada modus nontransaksional.

2.10 *Website*

Website merupakan fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada *website* disebut dengan *webpage* dan *link* dalam *website* memungkinkan pengguna bisa berpindah dari satu *page* ke *page* lain (*hypertext*), baik diantara *page* yang disimpan dalam *server* yang sama maupun *server* diseluruh dunia. *Pages* diakses dan dibaca melalui browser seperti *Netscape Navigator, Internet Explorer, Mozila Firefox, Google Chrome* dan aplikasi *browser* lainnya (Lukmanul, 2004).

Website (situs web) merupakan alamat (URL) yang berfungsi sebagai tempat penyimpanan data dan informasi dengan berdasarkan topik tertentu. URL adalah suatu sarana yang digunakan untuk menentukan lokasi informasi pada suatu *web*. Situs atau web dapat dikategorikan menjadi 2 (dua) yaitu:

1. *Web* Statis, yaitu *web* yang berisi atau menampilkan infomasi-informasi yang sifatnya statis (tetap).
2. *Web* Dinamis, yaitu *web* yang menampilkan informasi serta dapat berinteraksi dengan *user* yang sifatnya dinamis (Pardosi, 2004).

2.11 Android Studio

Android Studio merupakan *Integrated Development Environment* (IDE) resmi untuk pengembangan aplikasi Android, berdasarkan *IntelliJ IDEA*. Android berubah menjadi *platform* yang begitu cepat dalam melakukan inovasi. Hal ini tidak lepas dari pengembangan utama dibelakangnya, yaitu *Google*. Googlelah yang mengakuisisi android dan kemudian membuatkan sebuah *platform.*
Platform android terdiri dari Sistem Operasi berbasis *Linux,* sebuah GUI (*Graphic User Interface*), sebuah *web browser* dan Aplikasi Studio *End-User* yangdapat di *download* dan juga para pengembang bisa dengan leluasa berkarya

2.11.1 Sejarah Android

Pada awal mulanya, *Android Inc* merupakan sebuah perusahaan *software* kecil yang didirikan pada bulan Oktober 2003 di Palo Alto, California, USA. Didirikan oleh beberapa senior di beberpa perusahaan yang berbasis IT & *Communication,* Andy Rubin, Rich Miner, Nick Sears dan Chris White. Menurut Rubin, Android Inc Didirikan untuk mewujudkan *mobile device* yang lebih peka terhadap lokasi dan preferensi pemilik. Dengan kata lain, Android Inc ingin mewujudkan *mobile device* yang lebih mengerti pemiliknya. Android Studio menampilkan file proyek Anda dalam tampilan proyek Android.

Tampilan ini disusun oleh modul untuk menyediakan akses cepat ke file sumber utama proyek Anda. Semua *file build* terlihat di tingkat atas di bawah *Gradle Scripts* dan setiap modul aplikasi berisi *folder* berikut:

a. Bermanifestasi

Berisi file *Android Manifest*.xml.

b. Java

Berisi file kode sumber Java, termasuk kode uji Unit.

c. Res

Berisi semua sumber daya non-kode, seperti tata letak XML, string UI Struktur proyek Android pada disk berbeda dari representasi yang rata ini. Untuk melihat struktur file proyek yang sebenarnya, pilih *Project from the Project dropdown*. Anda juga dapat menyesuaikan tampilan file proyek agar fokus pada aspek spesifik pengembangan aplikasi Anda. Misalnya, memilih tampilan masalah dari proyek anda yang menampilkan tautan ke file sumber yang berisi kesalahan pengkodean dan sintaks yang dikenali, seperti tag penutup elemen XML yang hilang dalam file tata letak. File proyek dalam tampilan Masalah, menampilkan file tata letak dengan masalah.

d. Antarmuka Pengguna

Bilah alat memungkinkan Anda melakukan berbagai tindakan, termasuk menjalankan aplikasi dan meluncurkan alat Android. Bilah navigasi membantu anda menavigasi proyek anda dan membuka file untuk di edit. Ini memberikan tampilan struktur yang lebih kompak yang terlihat di jendela *Project.* Jendela *editor* adalah tempat anda membuat dan memodifikasi kode. Bergantung pada tipe *file* saat ini, *editor* bisa berubah. Misalnya, saat melihat *file layout*, *editor* menampilkan *Layout Editor*. *Tool window bar* berjalan di sekitar bagian luar jendela IDE dan berisi tombol yang memungkinkan Anda untuk memperluas atau merobohkan jendela alat individual.

Jendela alat memberi Anda akses ke tugas-tugas tertentu seperti manajemen proyek, pencarian, kontrol versi, dan banyak lagi. Anda dapat memperluas mereka dan runtuh mereka. Bilah status menampilkan status proyek anda dan IDE itu sendiri, serta peringatan atau pesan apa pun.

2.12 NOX App Player

Nox player merupakan aplikasi *emulator* Android. Mungkin ada yang belum tau mengenai istilah *emulator*, khususnya *emulator* Android. Jadi secara umum pengertian dari Nox adalah aplikasi untuk menjalankan game atau aplikasi yang berbasis Android di PC (baik itu berupa komputer ataupun laptop). Bagi para *gamers*, Nox sangat disarankan untuk dipergunakan sebagai aplikasi bermain game-game yang ada di Android. *Emulator* Android lain yang juga bagus untuk gaming yaitu *Koplayer* dan *Bluestacks*. Masing-masing *emulator* memiliki kelebihannya sendiri-sendiri.

Beberapa fitur Nox yang lain :

1. *Check for update* untuk memeriksa apakah ada pembaharuan versi pada Nox.
2. *Screenshot* untuk mengambil gambar dari tampilan yang ada di Nox.
3. *Fullscreen* untuk merubah tampilan Nox *player* menjadi sebesar ukuran *monitor.*

4. *Increase* dan *Decrease volume*, 2 tombol pengatur suara untuk menaikkandan menurunkan suara.
5. Add APK file untuk memasang atau *install* APK.
6. *Rotate* untuk merubah tampilan Nox yang biasanya memanjang dari kiri ke kanan (posisi yang sering digunakan untuk bermain game) menjadi berdiri.
7. *Video recorder* untuk merekam segala aktifitas pada Nox ke dalam bentuk video.
8. *Multi-instance* manager untuk membuka Nox player baru. Biasanya hal ini digunakan jika Anda mempunyai akun lebih dari satu tetapi ingin membukanya di waktu yang bersamaan.

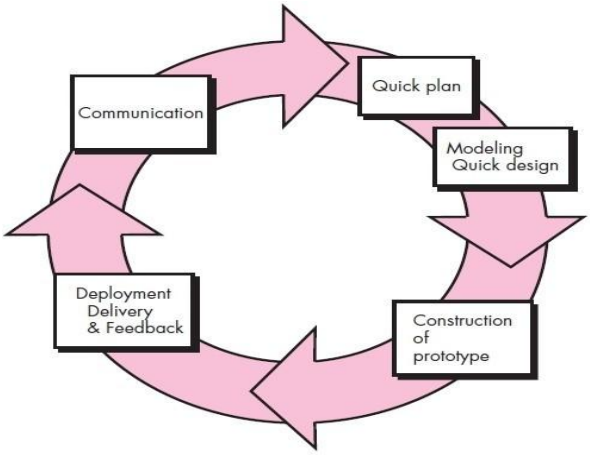
2.13 Metode Pengembangan sistem

Dalam suatu pengembangan perancangan sistem ini perlu adanya metode yang digunakan untuk pedoman dalam pengerjaannya, dengan mengikuti metode tersebut maka perancangan sistem ini bisa mencapai suatu tujuan yang baik seperti penulis harapkan. Metode yang penulis gunakan dalam perancangan sistem ini adalah metode *Prototype*. Metode *Prototype* merupakan suatu paradigma baru dalam metode pengembangan perangkat lunak dimana metode ini tidak hanya sekedar *evolusi* dalam dunia pengembangan perangkat lunak, tetapi juga merevolusi metode pengembangan perangkat lunak yang lama yaitu sistem sekuensial yang biasa dikenal dengan nama SDLC atau *waterfall development model*.

Prototyping adalah pengembangan yang cepat dan pengujian terhadap model kerja *(prototipe*) dari aplikasi baru melalui proses interaksi dan berulang-ulang yang biasa digunakan ahli sistem informasi dan ahli bisnis.

Prototyping disebut juga desain aplikasi cepat (*rapid application design*/RAD) karena menyederhanakan dan mempercepat desain sistem (O'Brien, 2005).

Model *Prototype* dan tahapannya dapat dilihat pada gambar 2.1 berikut:



Gambar 2.1 Model *prototype*

Tahap pertama adalah *communication* dan pengumpulan data awal yaitu tahap suatu perencanaan yang akan di lakukan, mulai dari menciptakan dan melaksanakan proses untuk memastikan bahwa perencanaan tersebut berkualitas tinggi, terpercaya, efisiensi biaya. Tahap kedua adalah *quick plan* yaitu analisis terhadap kebutuhan pengguna. Tahap ketiga adalah *modelling quick design* yaitu pembuatan *design* secara umum untuk selanjutnya di kembangkan kembali. Tahap keempat adalah *construction of prototype* adalah pembuatan perangkat prototype termasuk pengujian dan penyempurnaan. Tahap kelima adalah *deployment, delivery* dan f*eedback* adalah tahap penyerahan system kepada pengguna dan umpan balik.

2.14 *Unified Modelling Language* (UML)

Unified Modeling Language adalah bahasa standar yang digunakan untuk menjelaskan dan memvisualisaikan artifak dari proses analisis dan disain berorientasi objek. UML menyediakan standar pada notasi dan diagram yang bisa digunakan untuk memodelkan suatu *system.*

UML dikembangkan oleh 3 pendekar "berorientasi objek", yaitu Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. UML menjadi bahasa yang bisa digunakan

untuk berkomunikasi dalam perspektif objek antara *user* dengan developer, antara *developer* dengan developer, antara *developer* analis dengan*developer* disain, dan antara developer disain dengan *developer* pemrograman. UML memungkinkan *developer* melakukan permodelan secara *visual*, yaitu penekanan pada penggambaran, bukan didominasi oleh narasi. Permodelan *visual* membantu untuk menangkap struktur dan kelaukan dari objek, mempermudah penggambaran interaksi antara elemen dalam *system,* dan mempertahankan konsistensi antara disain dan implementasi dalam pemrograman.

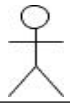
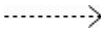
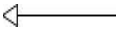
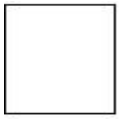
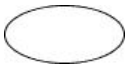
2.14.1 Use case Diagram

Use case diagram atau dalam bahasa Indonesia dikenal dengan diagram *use case* merupakan salah satu diagram yang ada dalam UML (*Unifield Modeling Language*) yang digunakan untuk memodelkan aspek perilaku sistem dari sistem yang akan dibuat dan untuk merekam persyaratan fungsional sebuah sistem
Dua hal utama yang harus ada pada *use case* adalah :

1. Aktor merupakan orang, proses, atau aplikasi lain yang berinteraksi dengan aplikasi yang akan dibuat di luar aplikasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsi-fungsi/proses-proses yang disediakan aplikasi sebagai unit-unit yang saling bertukar pesan/berinteraksi antar unit/prosesatau aktor.

Simbol-simbol *use case* diagram dapat di lihat pada tabel 2.2 berikut:

Tabel 2.2 Simbol *Use case* Diagram

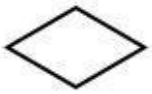
NO	SIMBOL	NAMA	KETERANGAN
1		*Actor*	Merupakan kesatuan eksternal yang berinteraksi dengan sistem
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
3		*Generalitation*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
4		*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.
5		*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use case*	Uraian sekelompok yang sling terkait membentuk sistem

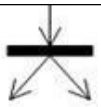
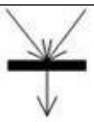
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah elemen- elemennya (sinergi)
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumberdaya komputasi

2.14.2 Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Simbol-simbol *Activity Diagram* dapat di lihat pada table 2.3 berikut:

Tabel 2.3 Simbol Activity Diagram

No	SIMBOL	NAMA	KETERANGAN
1		Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
2		Aktivitas	Aktivitas yang dilakukan sistem(kata kerja)
3		Percabangan / Decision	Percabangan pilihan aktivitas.
4		Penggabungan / Join	Penggabungan aktivitas yang lebih dari satu
5		Status Akhir	Status akhir yang di lakukan sistem
6		Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

7		Fork	Kegiatan yang di lakukan parallel
8		Join	Menunjukkan kegiatan yang di gabungkan

Activity Diagram yang disediakan oleh UML melengkapi *use case* yang telah dibuat sebelumnya memberikan representasi grafis dari aliran– aliran interaksi di dalam suatu skenario yang sifatnya spesifik. Mirip dengan diagram alir, suatu diagram aktifitas menggunakan sebuah kotan yang berisi lengkung untuk menggambarkan fungsi tertentu yang ada dalam suatu sistem yang akan dikembangkan, sementara itu tanda panah menggambarkan aliran didalam sistem dan seterusnya.

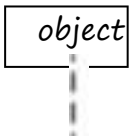
2.14.3 Squence Diagram

Sequence Diagram adalah suatu diagram yang menjelaskan interaksi objek dan menunjukkan (memberi tanda atau petunjuk) komunikasi diantara objek-objek tersebut. *Sequence diagram* digunakan untuk menjelaskan perilaku pada sebuah skenario dan menggambarkan bagaimana entitas dan sistem berinteraksi, termasuk pesan yang dipakai saat interaksi. Semua pesan digambbarkan dalam urutan pada eksekusi. *Sequence diagram* berkaitan erat dengan *Use Case* Diagram, yang mana 1 *Use Case* akan menjadi 1 *Sequence Diagram*.

Simbol-simbol sequence diagram dapat di lihat pada tabel 2.4 berikut ini:

Tabel 2.4 Simbol Squence Diagram

NO	SIMBOL	NAMA	KETERANGAN
1		*Life line*	Objek entitas, antar muka yang saling berinteraksi
2		*Actor*	Merupakan kesatuan eksternal yang berinteraksi dengan

			sistem
3		*Massage(call)*	Menggambarkan alur massage yang merupakan kejadian objek pengirim *life line* ke objek penerima *life line*
4		Massage(return)	Menggambarkan alur pemanggilan massage ke objek pemanggil dn tanda bahwa menyelesaikan prosesnya
5		object	Adalah instance dari sebuah class yang di tuliskan tersusun secara horizontal diikuti

a. Tujuan *Sequence* Diagram

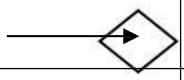
Berikut ini tujuan dari sequence digram adalah:

1. Menghubungkan r*equirement* kepada tim teknis karena diagram ini dapat lebih mudah untuk di kolaborasi menjadi model design.
2. Merupakan diagram yang paling relevan untuk menguraikan model deskripsi *use case* menjadi spesifikasi *design*.
3. Analisa dan Desain, memfokuskan pada identifikasi metode didalam sebuah sistem.
4. Menganalisa, mendesain dan memfokuskan pada identifikasi sebuah metode yang digunakan sistem.
5. *Sequence* diagram ini dipakai untuk menejelaskan dan memodelkan *use case.*
6. berfungsi untuk memodelkan sebuah logika dari sebuah method operasi, *function* ataupun prosedur.
7. berfungsi untuk memodelkan logika dari *service.*

2.14.4 *Class* Diagram

Class Diagram adalah diagram yang menunjukkan class-class yang ada dari sebuah system dan saling berhubungan, dan diagram ini menggambarkan alur struktur statis dari sebuah sistem. Oleh karena itu Class Diagram di anggap sebagai tulang punggungatau kekuatan dasar dari hamper setiap metode berorientasi objek termasuk UML. Symbol class diagram dapat di lihat pada table 2.5 berikut:

Table 2.5 simbol class diagram

NO	SIMBOL	NAMA	KETERANGAN
1	class +atribut +operasi()	Class	Simbol untuk membangun sebuah program dengan objek terdiri dari 3 bagian, bagian atas adalah kelas, bagian tengah adalh atribut, bagian bawah adalah metode dari kelas tersebut.
2		Generalisasi	Menandakan adany generalisasi dari kelas input untuk menghasilkan data yang di butuhkan
3		Asosiasi berarah	Relasi antar kelas dengan makna kelas yang satu di gunakan oleh kelas yang lain, assosiasi biasanya juga di sertai dengan *multiplicity*
4		kebergantungan	Relasi antar kelas dengan makna ketergantungan antar kelas
5		Agresi	Relasi antar kelas dengan makna semua bagian (*whole-part*)

2.15 *Black Box Testing*

Metode pengujian *black-box* dilakukan untuk menguji rute yang tampil pada sistem dengan cara membandingkannya dengan perhitungan manual. Teknik yang paling lazim digunakan dalam pengujian adalah pengujian kotak hitam (*Black- box*). Pengujian *Blackbox* dirancang untuk memvalidasi kebutuhan fungsional tanpa peduli dengan kerja internal dari program. Beberapa cara untuk memilih data pengujian untuk metode *black box* adalah sebagai berikut :

1. *Easy values*, yaitu data yang mudah diperiksa.
2. *Typical realistic value*, yaitu mencoba program dengan data pengujian untuk melihat bagaimana program menggunakannya. Data ini harus cukup sederhana sehingga hasilnya dapat dihitung secara manual.
3. *Extreme values*, banyak program error pada suatu batas range dari aplikasi.
4. *Ilegal values*, yaitu suatu data / nilai yang tidak diperbolehkan maupun data yang tidak berguna. Meskipun dirancang untuk mengungkap kesalahan, pengujian *black box* digunakan untuk memperlihatkan bahwa fungsi-fungsi perangkat lunak dapat beroperasi, bahwa input diterima dengan baik dan output dihasilkan dengan tepat, dan integritas informasi eksternal (seperti file data) dipelihara.

Dalam pengujian *black box* terdapat beberapa sifat *black box* testing yaitu :

1. *Robustness testing* (menguji kekuatan sistem yang ada) untuk menjamin sistem dengan memasukan data-data yang abnormal.
2. *Performance testing* (menguji kinerja sistem yang ada) menguji *software* bagian dari sistem/orientasi kepada *hardware.*
3. *Endurance testing* (menguji daya tahan sistem) menguji daya tahan terhadap *software* apakah sistem tersebut dapat bertahan dari gangguan-gangguan yang mengganggu.
4. *Behavior testing* (menguji prilaku sistem apakah sudah sesuai dengan permintaan) menguji tingkah laku sistem berdasarkan polanya “didesain sebagai mana mulanya.