

BAB II

LANDASAN TEORI

2.1. Rambu Lalu Lintas

2.1.1. Pengertian Rambu Lalu Lintas

Rambu lalu lintas menurut Keputusan Menteri Perhubungan Nomor: 61 tahun 1993 diartikan: “salah satu dari perlengkapan jalan, berupa lambang, huruf, angka, kalimat dan/atau perpaduan diantaranya sebagai peringatan, larangan, perintah atau petunjuk bagi pemakai jalan.”

Peraturan Pemerintah Republik Indonesia No : 32 Tahun 2011 menyatakan “Pengadaan, pemasangan, perbaikan, dan pemeliharaan perlengkapan jalan yang berkaitan langsung dengan pengguna dilakukan oleh menteri yang bertanggung jawab di bidang sarana dan prasarana lalu lintas dan angkutan jalan, gubernur, bupati, atau walikota sesuai dengan kewenangannya”.

Dalam keputusan Direktur Jendral Perhubungan Darat (Nomor : SK.116/AJ.404/DRJD/97) tentang penyelenggaraan rambu lalu lintas meliputi:

1. Inventarisasi tingkat pertumbuhan rambu lalu lintas.
2. Survei untuk menentukan kebutuhan rambu termasuk penentuan lokasi penempatan atau pemasangannya.
3. Perkiraan kebutuhan pasang untuk 5 tahun.
4. Penyusunan program dan pengadaan rambu.

Jenis rambu-rambu lalu lintas yang tercantum di undang-undang pemerintah terbagi menjadi 4 yaitu :

1. Rambu peringatan adalah rambu yang digunakan untuk menyatakan peringatan bahaya atau tempat berbahaya pada jalan di depan pemakai jalan.
2. Rambu larangan adalah rambu yang digunakan untuk menyatakan perbuatan yang dilarang dilakukan oleh pemakai jalan.
3. Rambu perintah adalah rambu yang digunakan untuk menyatakan perintah yang wajib dilakukan oleh pemakai jalan.
4. Rambu petunjuk adalah rambu yang digunakan untuk menyatakan petunjuk mengenai jurusan, jalan, situasi, kota tempat, pengaturan, fasilitas dan lain-lain bagi pemakai jalan.

Secara fisik rambu-rambu dibagi menjadi 2 bagian :

1. Daun rambu adalah plat aluminium atau bahan logam lainnya tempat ditempelkan/dilekatkannya rambu.
2. Tiang rambu adalah batangan logam atau bahan lainnya untuk menempelkan atau melekatkan daun rambu.

Rambu yang efektif harus memenuhi hal-hal berikut:

1. Memenuhi kebutuhan.
2. Menarik perhatian dan mendapat respek pengguna jalan.
3. Memberikan pesan yang sederhana dan mudah dimengerti.
4. Menyediakan waktu cukup kepada pengguna jalan dalam memberikan respon.

Untuk memenuhi kebutuhan tersebut, pertimbangan-pertimbangan yang harus diperhatikan dalam perencanaan dan pemasangan rambu adalah :

1. Keseragaman bentuk dan ukuran rambu
Keseragaman dalam alat kontrol lalu lintas memudahkan tugas pengemudi untuk mengenal, memahami dan memberikan respon. Konsistensi dalam penerapan bentuk dan ukuran rambu akan menghasilkan konsistensi persepsi dan respon pengemudi.

2. Desain rambu

Warna, bentuk, ukuran, dan tingkat retrorefleksi yang memenuhi standar akan menarik perhatian pengguna jalan, mudah dipahami dan memberikan waktu yang cukup bagi pengemudi dalam memberikan respon.

3. Lokasi rambu

Lokasi rambu berhubungan dengan pengemudi sehingga pengemudi yang berjalan dengan kecepatan normal dapat memiliki waktu yang cukup dalam memberikan respon.

4. Operasi rambu

Rambu yang benar pada lokasi yang tepat harus memenuhi kebutuhan lalu lintas dan diperlukan pelayanan yang konsisten dengan memasang rambu yang sesuai kebutuhan.

5. Pemeliharaan rambu

Pemeliharaan rambu diperlukan agar rambu tetap berfungsi baik.

2.2. Aplikasi

Menurut Wardana (2010) mendefinisikan Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Aplikasi merupakan rangkaian kegiatan atau perintah yang dieksekusi oleh komputer. Program merupakan kumpulan instruction set yang akan dijalankan oleh pemroses, yaitu berupa software. Bagaimana sebuah sistem komputer berpikir diatur oleh program ini. Program inilah yang mengendalikan semua aktivitas yang ada pada pemroses. Program berisi konstruksi logika yang dibuat oleh manusia, dan sudah diterjemahkan ke dalam bahasa mesin sesuai dengan format yang ada pada instruction set.

Aplikasi merupakan program yang siap pakai. Program direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain.

Contoh-contoh aplikasi yaitu program pemroses kata dan Web Browser. Aplikasi akan menggunakan operating system (OS) komputer dan aplikasi yang mendukung, Wardana (2010).

2.3. Android

Menurut Nazarudin (2014) android merupakan generasi baru platform Mobile. Platform yang memberikan pengembang untuk melakukan pengembangan sesuai yang diharapkannya. Sistem operasi yang mendasari Android dilisensikan di bawah GNU, general Public Lisensi Versi 2 (GPLv2), yang lebih dikenal dengan istilah copyleft, lisensi dimana setiap perbaikan pihak ketiga harus terus dibawah syarat (terms). Android di distribusikan dibawah Lisensi Apache Software (ASL/Apache 2), yang memungkinkan untuk distribusi kedua dan seterusnya. Komersialisasi pengembang (produsen handset khususnya) dapat memilih untuk meningkatkan platform tanpa harus memberikan perbaikan mereka ke masyarakat open source. Sebaliknya, pengembang dapat keuntungan dari perangkat tambahan seperti perbaikan dan mendistribusikan ulang pekerjaan mereka di bawah lisensi apapun yang mereka inginkan. Pengembang aplikasi Android diperbolehkan untuk mendistribusikan aplikasi mereka dibawah skema lisensi apapun yang mereka inginkan.

Selain itu menurut Nazarudin (2014) terdapat berbagai macam fitur dan versi dari *platform* android. Berikut penjelasan macam-macam fitur dan versi android :

2.3.1 Fitur-fitur yang ada di android

- a. Framework Aplikasi , pengembang memiliki akses penuh framework API (Application Programming Interface) yang sama yang digunakan oleh aplikasi inti.
- b. Mesin virtual Dalvik dioptimalkan untuk perangkat mobile.
- c. Integrated browser berdasarkan engine open source WebKit

- d. Grafis yang dioptimalkan dan didukung oleh perpustakaan grafis 2D, grafis 3D berdasarkan spesifikasi OpenGL ES 1.0
- e. SQLite untuk penyimpanan data
- f. Media Support yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- g. GSM Telephony (tergantung hardware)
- h. Bluetooth, EDGE, 3G, dan WiFi
- i. Kamera, GPS, kompas, dan accelerometer
- j. Lingkungan Development yang lengkap: perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk Eclipse IDE.

2.3.2 Versi Android

1. Android Versi 1.1
Dirilis oleh Google pada bulan Maret 2009.
2. Cupcake (1.5)
Hanya selang 1 bulan Google merilis Android 1.5 (Cupcake) dengan SDK (Software Development Kit).
3. Donut (1.6)
Versi ini dirilis pada September dengan menampilkan proses pencarian yang lebih baik, penggunaan baterai indikator dan kontrol applet VPN (Virtual Private Network) atau jaringan pribadi.
4. Eclair (2.0 / 2.1)
Versi Android awal yang mulai dipakai oleh banyak smartphone, fitur utama Eclair yaitu perubahan total struktur dan tampilan user interface dan merupakan versi Android yang pertama kali mendukung format HTML5.

5. Froyo / Frozen Yoghurt (2.2)

Android 2.2 dirilis dengan 20 fitur baru, antara lain peningkatan kecepatan, fitur Wi-Fi hotspot tethering dan dukungan terhadap Adobe Flash.

6. Gingerbread (2.3)

Perubahan utama di versi 2.3 ini termasuk update UI, peningkatan fitur soft keyboard & copy/paste, power management, dan support Near FieldCommunication.

7. Honeycomb (3.0, 3.1 dan 3.2)

Merupakan versi Android yang ditujukan untuk gadget / device dengan layar besar seperti Tablet PC; Fitur baru Honeycomb yaitu dukungan terhadap prosessormulticore dan grafis dengan hardware acceleration.

8. Ice Cream Sandwich (4.0)

Android 4.0 Ice Cream Sandwich diumumkan pada 10 Mei 2011 di ajang Google I/O Developer Conference (San Francisco) dan resmi dirilis pada tanggal 19 Oktober 2011 di Hongkong. Android versi 4.0 ini dapat digunakan di smartphone ataupun tablet. Fitur utama yang ditambahkan di Android 4.0 ialah Face Unlock, Android Beam, perubahan major User Interface, dan ukuran layar standar (native screen) beresolusi 720p (high definition).

9. Jelly Bean (4.1)

Android 4.1 Jelly Bean diumumkan pada 27 Juni 2012 pada konferensi Google I/O yang secara resmi dikenalkan ke publik sekitar Oktober 2012. Versi ini adalah yang tercepat dan terhalus dari semua versi Android. Fitur baru yang terapat di versi ini adalah meningkatkan kemudahan dan keindahan tampilan dari Ice Cream Sandwich dan memperkenalkan pengalaman pencarian Google yang baru di Android. Android 4.2 Jelly Bean juga menawarkan peningkatan kecepatan dan kemudahan Android 4.1 serta mencakup semua fitur baru seperti Photo Sphere dan desain

baru aplikasi kamera, keyboard Gesture Typing, Google Now dan lainnya.

10. Kit Kat (4.4)

Awalnya android versi ini di isukan bernama Key Lime Pie. Namun pada tanggal oktober 2013 google merilis kitkat sebagai generasi android berikutnya. Android versi ini memiliki banyak fitur & semakin memanjakan para pengguna android. Diantaranya : Immersive mode, Akses kontak langsung dari aplikasi telepon, google now launcher, dan pastinya memiliki interface UI yang baru.

11. Lollipop (5.0)

Android 5.0 merupakan versi paling baru dari sistem operasi android, Android 5.0 sendiri dianggap membawa update yang fantastis, banyak perubahan yang disertakan Google di dalamnya.

12. Marshmallow

Android versi 6.0 dikenal dengan *Marshmallow* dikenal juga dengan nama android M, os ini resmi di rilis pada 28 Mei tahun 2015 ,dari segi tampilan hampir mirip dengan android versi 5.0 lollipop ,hanya saja sudah di fitur keamanan di os ini sudah di tingkatkan.

2.4. Multimedia

Multimedia diambil dari kata multi dan media. Multi berarti banyak dan media berarti media atau perantara. Multimedia adalah gabungan dari beberapa unsure yaitu teks, grafik, suara, video dan animasi yang menghasilkan presentasi yang menakjubkan. Multimedia juga mempunyai komunikasi interaktif yang tinggi. Bagi pengguna komputer multimedia dapat diartikan sebagai informasi computer yang dapat disajikan melalui audio atau video, teks, grafik dan animasi, Turban dkk (2002).

Selain itu menurut Turban dkk (2002) disini dapat digambarkan bahwa multimedia adalah suatu kombinasi data atau media untuk menyampaikan suatu informasi sehingga informasi itu tersaji dengan lebih menarik.

Multimeida adalah kombinasi dari komputer dan video atau multimedia secara umum merupakan kombinasi tiga elemen, yaitu suara, gambar dan teks atau multimedia adalah kombinasi dari paling sedikit dua media input atau output dari data, media ini dapat audio (suara, musik), animasi, video, teks, grafik dan gambar.

Multimedia merupakan alat yang dapat menciptakan presentasi yang dinamis dan interaktif yang mengkombinasikan teks, grafik, animasi, audio dan gambar video (Robin dan Linda, 2001).

Menurut Robin dan Linda (2001) definisi lain dari multimedia adalah pemanfaatan komputer untuk membuat dan menggabungkan teks, grafik, audio, gambar bergerak (video dan animasi) menggabungkan link yang memungkinkan pemakai melakukan navigasi, berinteraksi, berkreasi dan berkomunikasi. Dalam definisi ini terkandung empat komponen penting multimedia. Pertama, harus ada komputer yang mengkoordinasikan apa yang dilihat dan didengar. Kedua, harus ada link yang menghubungkan pemakai dengan informasi. Ketiga, harus ada alat navigasi yang membantu pemakai menjelajah jaringan informasi yang saling terhubung. Keempat, multimedia menyediakan tempat kepada pemakai untuk mengumpulkan, memproses dan mengkomunikasikan informasi dengan ide.

2.4.1. Jenis Multimedia

Menurut Asyhar (2011), multimedia dapat di definisikan menjadi 2 kategori, yaitu:

- a. ***Multimedia Content Production*** adalah penggunaan beberapa media (teks, audio, *graphics*, *animation*, video dan *interactivity*) yang berbeda dalam menyampaikan suatu informasi atau menghasilkan produk multimedia seperti video, audio, musik, *film*, *game*, *entertainment*, dll. Bisa juga dikatakan sebagai penggunaan beberapa teknologi yang berbeda yang memungkinkan untuk menggabungkan media (teks, audio, *graphics*, *animation*, video, dan *interactivity*) dengan cara yang baru untuk tujuan komunikasi

- b. **Multimedia Communication** adalah penggunaan media (massa), seperti televisi, radio, media cetak dan internet untuk mempublikasikan / menyiarkan / mengkomunikasikan material periklanan, publikasi, *entertainment*, berita, pendidikan, dan lainnya.

2.4.2 Tipe Media yang disupport

Menurut Asyhar (2011) terdapat berbagai macam tipe media yang bisa *support* dalam membangun media pembelajaran. Berikut tipe-tipe media tersebut:

1. Audio

Menurut Asyhar (2011), menyatakan penyajian audio merupakan cara lain untuk memperjelas pengertian suatu informasi. Contohnya, narasi merupakan kelengkapan dari penjelasan yang dilihat melalui video. Suara dapat lebih menjelaskan karakteristik suatu gambar, misalnya musik dan suara efek (*sound effect*), maupun suara asli (*real sound*). Authoring software yang digunakan harus mempunyai kemampuan untuk mengontrol *recording* dan *playback*.

Perekaman musik yang baik memerlukan *sampling size* dan *sampling rate* yang tinggi. Beberapa macam authoring software dapat mengkonversi suara seperti format .WAV, .MID (MIDI), .VOC ATAU .INS dan dapat dihubungkan dengan sekuens dari animasi.

2. Video

Menurut Asyhar (2011), menyatakan video terdiri dari *full-motion* dan *live-video*. *Full-motion* video berhubungan dengan penyimpanan sebagai video clip, sedangkan *live-video* merupakan hasil pemrosesan yang diperoleh dari kamera.

Beberapa authoring tool dapat menggunakan *full-motion* video, seperti hasil rekaman menggunakan VCR, yang dapat menyajikan gambar bergerak dengan kualitas tinggi. File animasi memerlukan penyimpanan yang jauh lebih besar dibandingkan dengan file gambar.

3. Teks

Menurut Asyhar (2011), menyatakan teks merupakan dasar dari pengolahan kata dan informasi berbasis multimedia. Beberapa hal yang perlu diperhatikan adalah penggunaan *hypertext*, *auto-hypertext*, *text style*, *import text* dan *export text*.

4. Grafik

Menurut Asyhar (2011), menyatakan image atau grafik berarti *still image* seperti foto dan gambar. Manusia sangat berorientasi pada visual (*visual-oriented*), dan gambar merupakan sarana yang sangat baik untuk menyajikan informasi. Semua objek yang disajikan dalam bentuk grafik adalah bentuk setelah encoding dan tidak mempunyai hubungan langsung dengan waktu.

5. Animasi

Menurut Asyhar (2011), menyatakan animasi berarti gerakan image atau video seperti gerakan orang yang sedang melakukan suatu kegiatan, dll. Konsep dari animasi adalah menggambarkan sulitnya menyajikan informasi dengan satu gambar atau sekumpulan gambar. Demikian juga tidak dapat menggunakan teks untuk menerangkan informasi.

Animasi seperti halnya film, dapat berupa *frame-based* atau *cast-based*. *Frame-based animation* (animasi berbasis frame) dibuat dengan merancang setiap frame tersendiri sehingga mendapatkan tampilan akhir. *Cast-based animation* (animasi berbasis cast) mencakup pembuatan kontrol dari masing-masing objek (kadang disebut cast member atau actor) yang bergerak melintasi background. Hal ini merupakan bentuk umum animasi yang digunakan dalam game dan *object-oriented* software untuk lingkungan Window.

Dalam *authoring* software, biasanya animasi mencakup kemampuan 'recording' dan 'playback'. Fasilitas yang dimiliki oleh software animasi mencakup *integrated animation tool*, *animation clip*, *import animation*, *recording*, *playback* dan *transition effect*.

2.5. Animasi

Menurut Vaughan (2004) dalam Iwan Binanto (2010), animasi adalah usaha untuk membuat presentasi statis menjadi hidup. Animasi merupakan perubahan visual sepanjang waktu yang memberi kekuatan besar pada proyek multimedia dibuat. Animasi berasal dari kata “*toanimate*” yang artinya menggerakkan. Animasi sendiri merupakan suatu teknik menampilkan gambar berurutan sedemikian rupa sehingga penonton merasakan adanya ilustrasi gerakan (motion) pada gambar yang ditampilkan. Animasi adalah hasil dari proses menampilkan obyek-obyek gambar sehingga gambar yang ditampilkan akan tampak hidup. Tidak hanya menghidupkan, animasi juga memberikan karakter kepada obyek-obyek tersebut (Binanto, 2010).

Animasi adalah gambar-gambar yang bergerak dengan kecepatan, arah dan cara tertentu. Jadi, dalam pembentukan animasi ada tipe data pemrograman yang baru, yaitu waktu. Dengan animasi, maka penyajian informasi menjadi lebih hidup dan menarik (Oetomo,2006).

Menurut Vega (2004), animasi dapat kita nikmati dengan cara menggunakan sekumpulan gambar yang berubah sedikit demi sedikit, yang ditampilkan secara berurutan dengan kecepatan tertentu, sehingga menimbulkan kesan bergerak pada rentetan gambar yang diam. Kehalusan animasi bisa ditentukan melalui banyaknya gambar yang ditampilkan dalam 1 detik atau disebut frame per second. Biasanya dalam animasi standart, digunakan patokan 18 sampai 24 gambar perdetik.

2.6. Metode *Rational Unified Process* (RUP)

Menurut Rosa A.S M. Shalahudin (2016:124) Unified Process atau dikenal juga dengan proses iterative dan incremental merupakan sebuah proses pengembangan perangkat lunak yang dilakukan secara iteratif (berulang) dan incremental (bertahap dengan progress menaik). *Rational Unified Process* (RUP) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (iterative), focus pada arsitektur (architecture-centric), lebih diarahkan berdasarkan penggunaan kasus (use case driven). RUP merupakan

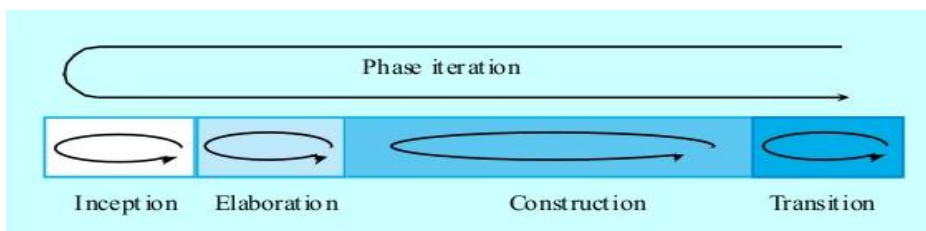
proses rekayasa perangkat lunak dengan pendefinisian yang baik (well defined) dan penstrukturan yang baik (well structured). RUP menyediakan pendefinisian struktur yang baik untuk alur hidup proyek perangkat lunak. RUP adalah sebuah produk proses perangkat lunak yang dikembangkan oleh *rational software* yang diakuisisi oleh IBM dibulan february 2003. Proses pengulangan/iteratif pada RUP secara global dapat dilihat pada gambar 2.1 berikut :



Gambar 2.1 Proses Iteratif RUP

2.6.1. Fase RUP

RUP memiliki empat buah tahap atau fase yang dapat dilakukan pula secara iteratif. Berikut ini adalah gambar 2.2 alur hidup RUP.



Gambar 2.2 Alur Hidup RUP

Berikut ini penjelasan untuk setiap fase pada RUP, yaitu :

2.6.1.2. Inception (Permulaan)

Tahap ini lebih pada pemodelan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*).

Berikut adalah tahap yang dibutuhkan pada tahap ini:

- a. Memahami ruang lingkup dari proyek (termasuk pada biaya, waktu, kebutuhannya, risiko dan lain sebagainya).
- b. Membangun kasus bisnis yang dibutuhkan.

Hasil yang diharapkan dari tahap ini adalah memenuhi *lifecycle objective milestone* (batasan/tonggak objektif dari siklus) dengan kriteria berikut:

- a. Umpan balik dari pendefinisian ruang lingkup, perkiraan biaya, dan perkiraan jadwal.
- b. Kebutuhan dimengerti dengan pasti (dapat dibuktikan) dan sejalan dengan kasus primer yang dibutuhkan.
- c. Kredibilitas dari perkiraan biaya, perkiraan jadwal, penentuan skala prioritas, risiko dan proses pengembangan.
- d. Ruang lingkup purwarupa (*prototype*) yang akan dikembangkan.
- e. Membangun garis dasar dengan membandingkan perencanaan aktual dengan perencanaan yang direncanakan.

Jika pada akhirnya tahap ini target yang diinginkan tidak dicapai maka dapat dibatalkan atau diulang kembali setelah dirancang ulang agar kriteria yang diinginkan dapat dicapai. Batas/tonggak objektif digunakan untuk mendeteksi apakah sebuah kebutuhan akan sistem dapat diimplementasi atau tidak.

2.6.1.2. Elaboration (Perluasan/Perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat

dibuat atau tidak. Mendeteksi risiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (*prototype*).

2.7. Software Pendukung dan Bahasa Pemrograman

2.7.1 Adobe Flash Professional CS6

Menurut Andi (2013) *Adobe Flash Professional CS6* merupakan perangkat lunak multiguna yang dapat dimanfaatkan untuk berbagai macam animasi dengan berbagai fitur canggih yang ada didalamnya dan juga dapat menggambar, membuat animasi, hingga digunakan untuk membuat *game* yang menarik dan berkualitas. Area kerja *Adobe Flash Professional CS6* dirancang secara khusus agar ruang kerja yang digunakan dapat diatur dan lebih mudah dipahami oleh pemakai pemula maupun para desainer *flash* yang telah berpengalaman.

Adobe Flash Professional CS6 adalah program yang cukup kompleks sehingga tidak mungkin untuk menjabarkannya secara lengkap. Disini hanya akan diberikan sedikit pengertian dan fungsi dasar dari *Adobe Flash Professional CS6*.

2.7.2 Adobe Integrated Runtime

Menurut Wahana Komputer (2014) *Adobe AIR* adalah sebuah cross operating system runtime yang dikembangkan oleh Adobe sehingga memungkinkan pengembang memanfaatkan keterampilan mereka (seperti Flash, Flex, HTML, Javascript, dan PDF) untuk membangun RIA (*Rich Internet Application*) dan kontennya ke dalam platform baru.

2.7.3 Action Script 3.0

Menurut Madcoms (2012), mendefinisikan bahwa *actionscript* yaitu bahasa pemrograman pada *adobe flash player* dan *adobe AIR*. Dimana bahasa pemrograman ini bisa interaktif, menangani data, dan banyak digunakan pada

Flash, *Flex*, dan *AIR* baik konten maupun aplikasi. *Actionscript* dapat di eksekusi oleh *Actionscript VirtualMachine (AVM)*, yang mana ini merupakan bagian dari *Flash* dan *AIR*. Bahasa pemrograman ini akan familiar dengan para *develover* dengan kemampuan yang minim terhadap OOP (*Object Oriented Programming*).

Selain itu menurut Madcoms (2012), *actionscript 3.0* mempunyai kapabilitas yang lebih bagus dari pada *Actionscript* sebelumnya. Yaitu di desain untuk memfasilitasi kreasi atau pekerjaan yang mempunyai kompleksitas yang tinggi dalam membangun aplikasi dengan data set yang besar dan *ObjectOriented, code bases* yang bisa dipakai berulang kali. Bahasa pemrograman ini dapat mengeksekusi 10 kali lebih cepat dari pada *actionscript* sebelumnya.

2.8. Adobe Photoshop

AdobePhotoshop adalah salah satu *software* pengolah *grafik* yang banyak digunakan oleh para *desainer grafis* dan *webdi* seluruh dunia. Tampilannya yang mudah difahami, kelengkapan fasilitas yang ditawarkan, serta kemudahan memperoleh fasilitas pendukung dari berbagai sumber menjadikan *Photoshop* menjadi pilihan paling handal bagi para desainer, HenkyPrihatna (2005).

2.9. UML (Unified Modeling Language)

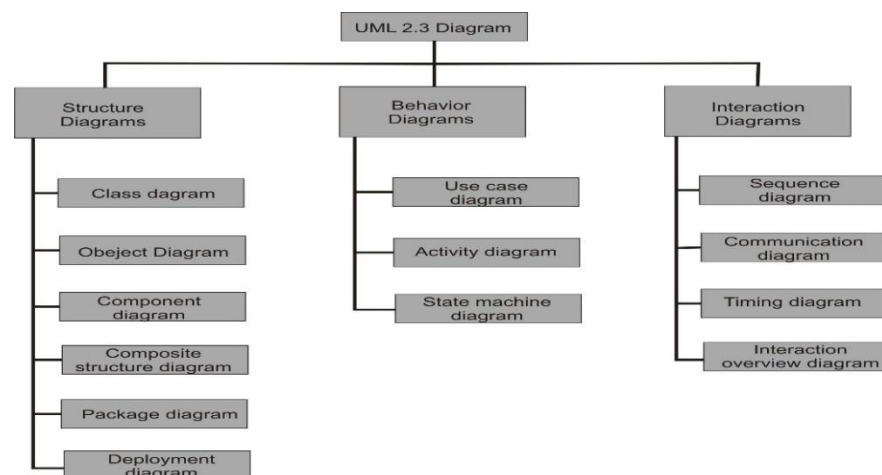
Menurut Rosa dan Shalahuddin (2015) Unified Modelling Language (UML) adalah salah satu standar bahasa yang banyak di gunakan di dunia industri untuk mendefenisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek dengan menggunakan diagram dan teks-teks pendukung.

2.9.1. Diagram UML

Menurut Rosa dan Shalahuddin (2015) pada UML 2.3 terdiri dari 13 macam diagram yang di kelompokkan dalam 3 kategori.

- 1) *Structure diagrams* yaitu kumpulan diagram yang di gunakan untuk menggambarkan suatu struktur statis dari sistem yang di modelkan.
- 2) *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
- 3) *Interaction diagram* yaitu kumpulan diagram yang di gunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

Lebih jelasnya dapat di lihat pada gambar 2.3 berikut.



Gambar 2.3 Diagram UML

(sumber : buku rekayasa perangkat lunak terstruktur dan berorientasi objek,2015)

Berikut penjelasan dari jenis-jenis diagram tersebut antara lain:

1. *Class Diagram*

Menggambarkan struktur sistem dari segi pendefenisian kelas-kelas yang akan di buat untuk membangun sistem

2. *Object Diagram*

Menggambarkan struktur sistem dari segi penanaman objek dan jalannya objek dalam sistem.

3. *ComponentDiagram*

Menunjukkan organisasi dan ketergantungan di antara kumpulan komponen dalam sebuah sistem.

4. *Composite StructureDiagram*

Menggambarkan struktur dari bagian – bagian yang saling terhubung maupun mendeskripsikan struktur pada saat berjalan (*runtime*) dari *instance* yang saling terhubung.

5. *Package Diagram*

Menyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam diagram uml.

6. *Deployment Diagram*


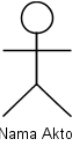

Menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi.

7. *Use Case Diagram*

Pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan di buat dapat di lihat pada tabel

Tabel 2.1. Simbol – simbol *use case diagram*

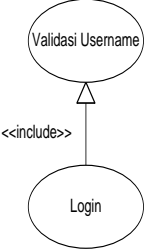
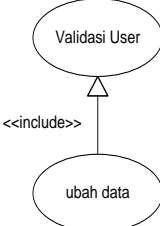
(sumber : buku rekayasa perangkat lunak terstruktur dan berorientasi objek, 2015)

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

Tabel 2.1. Lanjutan Simbol-simbol *use case diagram*

<p>Ekstensi / extend</p> <p style="text-align: center;">«extends»</p> <p style="text-align: center;">-----></p>	<p>Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya use case tambahan memiliki nama depan yang sama dengan use case yang sama dengan use case yang ditambahkan, misalnya :</p> <div style="text-align: center;"> <pre> graph TD A(Validasi sidik jari) -.-> «extends» B(Validasi User) B -.-> «extends» C(Validasi username) </pre> </div> <p>arah panah mengarah pada use case yang ditambahkan; biasanya use case yang menjadi extend-nya merupakan jenis yang sama dengan use case yang menjadi induknya.</p>
<p>Generalisasi/ <i>generalization</i></p> <p style="text-align: center;">—————></p>	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi user yang lebih umum dari lainnya, misalnya:</p> <div style="text-align: center;"> <pre> graph TD A(Ubah Data) --> B(Mengolah data) C(hapus data) --> B </pre> </div> <p>Arah panah mengarah pada use case yang menjadi generalisasinya (umum).</p>

Tabel 2.1. Lanjutan Simbol-simbol *use case diagram*



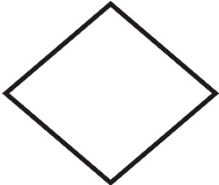


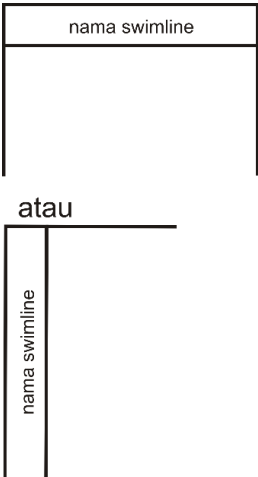
<p>Menggunakan <i>/include/uses</i></p> <p>--<<include>>--></p> <p>--<<uses>>--></p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan pada <i>use case</i> ini.</p> <p>Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>:</p> <ul style="list-style-type: none"> • <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, misal pada kasus berikut:  <pre> graph BT Login((Login)) -- "<<include>>" --> ValidasiUsername((Validasi Username)) </pre> <ul style="list-style-type: none"> • <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:  <pre> graph BT ubahData((ubah data)) -- "<<include>>" --> ValidasiUser((Validasi User)) </pre> <p>Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>
--	--

8. Activity Diagram

Menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak dapat di lihat pada tabel 2.3 berikut

Tabel 2.2 Simbol-Simbol *Activity Diagram*

(sumber : buku rekayasa perangkat lunak terstruktur dan berorientasi objek,2015)

Simbol	Deskripsi
<p>Status awal</p> 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
<p>Aktivitas</p> 	Aktivitas yang di lakukan sistem, aktivitas biasanya diawali dengan kata kerja
<p>Percabangan/decision</p> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
<p>Penggabungan/join</p> 	Asosiasi penggabungan dimana lebih dari satu aktivitas di gabungkan menjadi satu
<p>Status akhir</p> 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
<p>Swimlane</p> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

9. State Machine Diagram

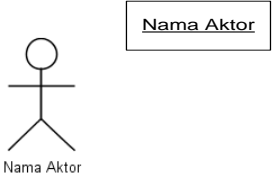
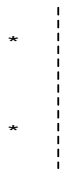
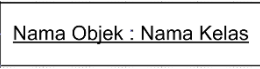

Di gunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek.

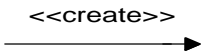
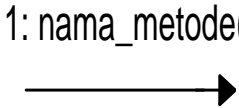

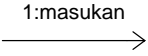
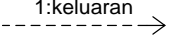
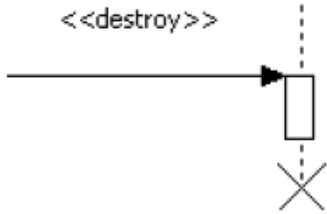
10. Sequence Diagram

Menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek dapat di lihat pada tabel 2.4

Tabel 2.3 Simbol – simbol *sequence diagram*

(sumber : buku rekayasa perangkat lunak terstruktur dan berorientasi objek, 2015)

Simbol	Deskripsi
<p>Aktor</p> <p style="text-align: center;">atau</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.</p>
<p>Garishidup / lifeline</p> 	<p>Menyatakan kehidupan suatu objek.</p>
<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktuaktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.</p>

<p>Pesantipe create</p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
<p>Pesantipe call</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>  <p>arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p>
<p>Pesantipe send</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
<p>Pesantipe return</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
<p>Pesantipe destroy</p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada <i>destroy</i>.</p>

11. *Communication* Diagram

Menggambarkan interaksi antar objek /bagian dalam bentuk urutan pengiriman pesan

12. *Timing* Diagram

Fokus dalam penggambaran terkait batasan waktu

13. *Interaction Overview* Diagram

Menggambarkan sekumpulan urutan aktivitas.

2.10. *Storry Board*

Menurut Diartono(2008) storyboard merupakan suatu pemetaan elemen-elemen multimedia dalam setiap layar program multimedia. Storyboard ini merupakan sebuah panduan bagi para programmer dan graphics designer dalam membangun suatu proyek multimedia, karena dalam storyboard ini digambarkan mengenai elemen-elemen apa saja yang digunakan dalam setiap rancangan layar yang akan dibangun. Storyboard juga berfungsi sebagai media penghubung antara programmer design dan client sebelum aplikasi dibangun. Sebuah storyboard yang telah dibuat dapat dijelaskan layar-layar yang akan dibuat secara berurutan. Sebuah storyboard harus terdiri dari :

- a. Nama proyek, modul dan nomor halaman atau nomor layar.
- b. Gambar sketsa layar atau halaman beserta rincian objek-objek yang ada dalam layar, seperti halnya teks, gambar, animasi, suara, narasi, video, warna, o, serta ukuran gambar.