

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Rambu lalu lintas**

Leksmono (2008:2), mendefinikan rekayasa lalu lintas adalah bidang kajian yang mempelajari metode perancangan ruang lalu lintas jalan yang aman dan nyaman bagi pengguna jalan dan efisien dari sudut pandang pembiayaan atau penggunaan lahan. Salah satu rekayasa lalu lintas adalah perencanaan peraturan lalu lintas. Salah satu struktur peraturan lalu lintas yang mengatur pengguna jalan adalah peraturan jalan dan lalu lintas. Leksmono (2008:117) dalam bukunya yang berjudul “Rekayasa Lalu Lintas”, menyatakan bahwa aturan di jalan meliputi hal-hal sebagai berikut:

- a. Right of away ,
- b. Batas kecepatan,
- c. Rambu, sinyal dan marka,
- d. Alat pengendali, dan lain-lain.

Menurut UU No. 22 Tahun 2009 tentang Angkutan Jalan, rambu lalu lintas dapat didefinisikan sebagai bagian perlengkapan jalan yang berupa lambang, huruf, angka, kalimat, atau perpaduan yang berfungsi sebagai peringatan, larangan, perintah, atau petunjuk bagi pengguna jalan. Ada tiga jenis informasi yang digunakan oleh pemakai jalan yaitu :

1. Yang bersifat perintah atau larangan yang harus dipatuhi.
2. Peringatan suatu bahaya.
3. Petunjuk berupa arah, identifikasi tempat, fasilitas-fasilitas alat pengendalian yang efektif.

Ada beberapa rambu-rambu lalu lintas menurut bentuk dan fungsinya antara lain :

**2.1.1 Rambu peringatan** (sebagian besar berwarna dasar kuning), digunakan untuk menyatakan peringatan bahaya atau tempat berbahaya



Gambar 2.1. Rambu peringatan

**2.1.2 Rambu larangan** (sebagian besar berwarna dasar putih dan bergaris tepi merah), digunakan untuk menyatakan perbuatan yang dilarang dilakukan oleh pemakai jalan.



Gambar 2.2. Rambu larangan

**2.1.3 Rambu perintah** (sebagian besar berwarna dasar biru), digunakan untuk menyatakan perintah yang wajib dilakukan oleh pemakai jalan.



Gambar 2.3. Rambu perintah

**2.1.4 Rambu petunjuk** (sebagian berwarna dasar putih bergaris tepi biru, berwarna dasar hijau atau coklat), digunakan untuk menyatakan petunjuk mengenai jurusan, jalan, situasi, kota, tempat, pengaturan, fasilitas dan lain-lain.



Gambar 2.4. Rambu petunjuk

## 2. 2 Penelitian terdahulu

### 2. 2. 1 Petualangan Baseta

AKP Andhika Wiratama SH., S. IK. sedang memprakarsai sebuah *game* *Petualangan BASETA* yang beerasal dari Kasat Lantas Polres Jepara. Setelah melihat kegelisahan para pemohon SIM yang gagal dalam ujian teori. Sebuah *game* yang tidak hanya menghibur, tapi juga memberikan berbagai pengetahuan tentang marka jalan dan cara berkendara dengan baik dan benar. Dalam *game* PETUALANGAN BASETA, setiap pemain memiliki misi mengumpulkan poin sebanyak-banyaknya, setiap soal yang terjawab dengan benar akan memberikan poin plus(+) bagi pengguna dan setiap pelanggaran yang dilakukan akan memberikan poin minus (-) dari poin yang telah dikumpulkan

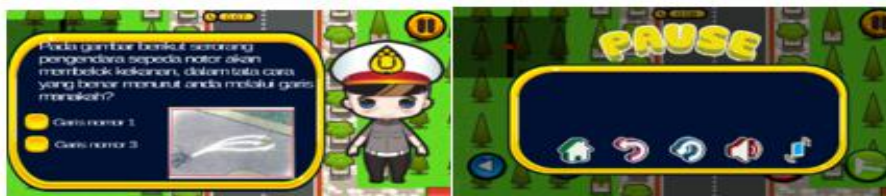
User interface :



Gambar 2.5. Tampilan utama dan Pilih jenis sim A dan B



Gambar 2.6. Pilih episode sim A dan Level di episode Sim A



Gambar 2.7.pertanyaan dari sim A dan Menu Pause



Gambar 2.8. Di episode sim B dan Pilih level di sim B



Gambar 2.9. Permainan episode sim B dan Pertanyaan di sim B

➤ **Kelebihan :**

1. Informasi soal sesuai dengan standard soal pada ujian teori untuk memperoleh SIM A maupun SIM C.
2. Bisa mengetahui tentang bagaimana tata cara berkendara yang tertib dan benar
3. Ketika Pemain banyak pengalaman dalam memainkan game ini, maka semakin banyak pula soal-soal dalam ujian teori SIM yang diketahui, Sehingga pengetahuan soal SIM akan meningkat karena dalam setiap jawaban akan langsung diberitahu apakah itu benar atau salah,.

### ➤ Kekurangan

1. untuk menu pengaturan masih belum berfungsi
2. untuk music tidak begitu menarik
3. untuk menu reward masih sulit untuk melihatnya harus terlebih dahulu koneksi internet.
4. untuk sebuah rintangan masih kurang jadi permainan masih terlihat mudah

### 2. 2. 2 Ancinta

Mengajari sang buah hati tentang rambu-rambu lalu lintas sejak usia dini adalah hal yang wajib dilakukan oleh orang tua. Jika orang tua lengah akan terjadi seperti anak SD yang mengendarai motor tanpa aturan dan membawa penumpang lebih dari dua.

*Ancinta Anak Cinta Lalu Lintas* merupakan salah satu permainan edukasi yang sangat mendidik dengan mengajarkan aturan dan rambu lalu lintas. Di sini, Anda akan menggunakan karakter bernama Anchita, seorang polwan yang bisa Anda dapatkan secara gratis. Anda harus membantunya untuk berhati-hati dan waspada dalam mengendarai sepedanya. Ada beberapa rintangan yang harus dilaluinya seperti tong yang ada di tengah, galian lubang, dan lainnya. Tugas Anda adalah menghindari semua rintangan tersebut agar bisa berkendara lebih lancar.

### User Interface :



Gambar 2.10. Tampilan awal game dan Tampilan misi



Gambar 2.11. Tampilan rangking dan Tampilan karakter pemain



Gambar 2.12. Tampilan permainan yang di mulai dan Tampilan pertanyaan



Gambar 2.13. Tampilan jika kita kalah

➤ **Kelebihan :**

1. Informasi soal mengenai rambu-rambu lalu lintas
2. Bisa mengetahui dan mengenali macam-macam rambu-rambu lalu lintas
3. Dengan bertambahnya pengetahuan tentang berkendara dan mengenal fungsi-fungsinya yang benar.

➤ **Kekurangan :**

1. tidak ada menu store atau menu pembelian.
2. tampilan interface untuk menu misi masih kurang menarik
3. fungsi diamond masih belum bermanfaat
4. tidak ada next level hanya sekali main saja

### 2.3 Game Edukasi

Pengertian sederhana dari game edukasi adalah game yang dirancang dengan tujuan untuk pengayaan pendidikan. Menurut Hurd dan Jenuings (2009) menyebutkan educational game adalah game yang khusus dirancang untuk mengajarkan user suatu pembelajaran tertentu, pengembangan konsep dan pemahaman dan membimbing mereka dalam melatih kemampuan mereka, serta memotivasi mereka untuk memainkannya.

Game edukasi dapat digunakan dalam memberikan pengajaran, menambah pengetahuan penggunaannya dengan cara yang menyenangkan dan menarik. Game

edukasi dirancang dan biasanya ditujukan untuk anak-anak, karena anak-anak tidak akan pernah lepas dengan yang namanya permainan atau game .

Pengertian sederhana dari game edukasi adalah game yang dirancang dengan tujuan untuk pengayaan pendidikan. Menurut Hurd dan Jenuings (2009) menyebutkan educational game adalah game yang khusus dirancang untuk mengajarkan user suatu pembelajaran tertentu, pengembangan konsep dan pemahaman dan membimbing mereka dalam melatih kemampuan mereka, serta memotivasi mereka untuk memainkannya.

Game edukasi dapat digunakan dalam memberikan pengajaran, menambah pengetahuan penggunaannya dengan cara yang menyenangkan dan menarik. Game edukasi dirancang dan biasanya ditujukan untuk anak-anak, karena anak-anak tidak akan pernah lepas dengan yang namanya permainan atau game .

## 2.4 Smartphone

### 2.4.1 Pengertian Smartphone

Menurut Yang et al. (2007), *smartphone* merupakan *cellphone* yang menggabungkan fungsi - fungsi *Personal Digital Assistant* (PDA) seperti kalender, *personal schedule*, *address book* dan memiliki kemampuan untuk mengakses internet, membuka email, membuat dokumen, bermain game, serta membuka aplikasi lainnya.

Schmidt (2009) juga mengemukakan bahwa istilah *smartphone* merupakan istilah yang digunakan untuk mendeskripsikan *mobile device* yang menggabungkan fungsi *cellphone*, *PDA*, *audio player*, *digital camera*, *camcorder*, *Global Positioning System* (GPS) *receiver*, dan *Personal Computer* (PC).

### 2.4.2 Sistem Operasi Smartphone

Menurut Fling (2009), yang disadur dari bukunya yang berjudul *Mobile Design and Development*, terdapat beberapa sistem operasi pada *smartphone* yang umum digunakan yaitu:

1. *Symbian*
2. *Windows Mobile*
3. *Palm OS*
4. *Linux*
5. *Android*

## 2.5 Android

**Android** adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Keunggulan utama Android adalah gratis dan *open source*, yang membuat smartphone Android dijual lebih murah dibandingkan dengan Blackberry atau iPhone meski fitur (hardware) yang ditawarkan Android lebih baik. Beberapa fitur utama dari Android antara lain WiFi hotspot, Multi-touch, Multitasking, GPS, accelerometers, support java, mendukung banyak jaringan (*GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE & WiMAX*) serta juga kemampuan dasar handphone pada umumnya.

Ada beberapa jenis android diantaranya :

- Android Alpha Rilis sekitar awal 2007
- Android Beta Rilis pada bulan November 2007
- Android Versi 1.0 (Angel Cake) Rilis pada tanggal 23 September 2008
- Android Versi 1.1 (Bettenberg) Rilis 9 Februari 2009
- Android Versi 1.5 (Cupcake) Rilis 30 April 2009
- Android Versi 1.6 (Donut) Rilis 15 September 2009
- Android Versi 2.1 (Eclair) 12 Januari 2010
- Android Versi 2.2–2.2.3 (Froyo) 20 Mei 2010
- Android Versi 2.3.3–2.3.7 (Gingerbread) 9 Februari 2011
- Android Versi 3.2 (Honeycomb) 15 Juli 2011
- Android Versi 4.0.3–4.0.4 (Ice Cream Sandwich) 16 Desember 2011
- Android Versi 4.3 (Jelly Bean) 24 Juli 2013
- Android Versi 4.4 (KitKat) Rilris 31 Oktober 2013
- Android Versi 5.0 (Lollipop) 15 Oktober 2014
- Android Versi 6.0 (Marshmallow ) Rilis 30 September 2015



### 2.5.1 Sejarah Android

Seperti halnya *software* yang lain, *Android* juga mengalami perkembangan setiap waktunya yang dapat dilihat dari jumlah versinya dimana terdapat perbedaan antara versi yang satu dengan yang sesudahnya.

Keunikan dari nama sistem operasi (OS) *Android* adalah dengan menggunakan nama makanan hidangan penutup (*Dessert*). Selain itu juga nama-nama OS *Android* memiliki huruf awal berurutan sesuai abjad; *Cupcake*, *Donut*, *Eclair*, *Froyo*, *Gingerbread*, *Honey Comb*, *Ice Cream*, *Jelly Bean*, *Kit Kat*, *Lollipop* dan *Marshmallow*. Seperti penjelasan berikut:

#### a. Android Beta

Pertama kali dirilis pada 5 November 2007, kemudian pada 12 November 2007 *Software Development Kit* (SDK) dirilis oleh Google.

#### b. Android 1.0 Astro

Pertama kali dirilis pada 23 September 2008. Sebenarnya *Android* versi pertama ini akan dinamai dengan nama “Astro” tapi karena alasan hak cipta dan trademark nama ”Astro” tidak jadi disematkan pada versi pertama dari OS *Android* ini. HTC Dream adalah ponsel pertama yang menggunakan OS ini.

#### c. Android 1.1 Bender

Pertama kali dirilis pada 9 Februari 2009. Versi *Android* kedua ini juga mengalami masalah penamaan yang sama dengan versi pertamanya. Pada awalnya *Android* ini akan diberi nama “Bender” akan tetapi karena alasan melanggar trademark, nama “Bender” tidak jadi disematkan pada versi *Android* ini. Awalnya versi OS *Android* ini dirilis untuk perangkat T-Mobile G1 saja. Versi ini merupakan update untuk memperbaiki beberapa bugs, mengganti API dan menambahkan beberapa fitur. Berikut penampakan *home screen OS Android v.1.1*.

#### d. Android 1.5 Cupcake

Pertama kali dirilis pada 30 April 2009. Nah, mulai versi *Android* ini penamaan menggunakan nama makan pencuci mulut (*dessert*) mulai digunakan, karena ini merupakan versi yang ketiga maka penamaan diawali dengan huruf “C” dan jadilah “Cupcake” menjadi nama resmi dari versi OS *Android* ketiga ini. OS ini berbasis pada kernel Linux 2.6.27 dan menambahkan beberapa update serta UI baru dari versi *Android* sebelumnya. Mulai terdapat “widget” yang dapat dibesar kecilkan. Kemudian ditambah kemampuan untuk meng-upload video dan gambar ke Youtube dan Picasa.

e. Android 1.6 Donut

Dirilis pertama kali pada 15 September 2009. Terdapat peningkatan pada fitur pencarian dan UI yang lebih user friendly. Pada versi ini juga sudah mendukung teknologi CDMA/EVDO, 802.1x, VPNs. Kemudian support layar dengan resolusi WVGA. Berikut penampakan *Android* v1.6 Donut.

f. Android 2.0/2.1 Éclair

Dirilis pertama kali pada 9 Desember 2009. Terjadi penambahan fitur untuk pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1. Beberapa versi updatenya antara *Android* v.2.0 kemudian v2.0.2 dan terakhir v.2.1.

g. Android 2.2 Froyo (Froze Yoghurt)

Dirilis pertamakali pada 20 Mei 2010 pada *smartphone* Google Nexus One. Pada versi ini sudah support terhadap Adobe Flash Player 10.1. Peningkatan pada kecepatan membuka dan menutup aplikasi, serta penggunaan *SD Card* sebagai tempat penyimpanan aplikasi. Ketika *Android* Froyo hadir mulai muncul banyak diskusi yang membahas mengenai persaingan antara *Android* dengan *iOS* yang akan semakin ketat di masa yang akan datang. Beberapa versi update yang dirilis antara lain *Android* v.2.2.1 hingga v.2.2.3.

h. Android 2.3 Gingerbread

Pertama kali diperkenalkan pada 6 Desember 2010. Terjadi banyak peningkatan pada versi *Android* yang satu ini dibandingkan dengan versi sebelumnya. Dirancang untuk memaksimalakan kemampuan aplikasi dan game. Serta mulai digunakannya *Near Field Communication* (NFC). Perbaikan terhadap dukungan layar resolusi *WXGA* dan di atasnya. Beberapa versi update yang dirilis antara lain v.2.3.3 hingga v.2.3.7. Sampai saat ini *Android* Gingerbread merupakan versi *Android* yang memiliki pengguna terbanyak dibandingkan dengan seri *Android* lainnya, yaitu mencapai 65% dari seluruh versi *Android* yang dirilis.

i. Android 3.0/3.1 Honey comb

Pertama kali diperkenalkan pada 22 Februari 2011 dan *Motorola Xoom* adalah yang pertama kali menggunakannya. *Android* versi ini merupakan OS yang didesain khusus untuk pengoptimalan penggunaan pada tablet PC.

j. Android 4.0 ICS (Ice cream Sandwich)

Pertama kali dirilis pada 19 Oktober 2011. *Smartphone* yang pertama kali menggunakan OS *Android* ini adalah Samsung Galaxy Nexus. Secara teori semua perangkat seluler yang menggunakan versi *Android* sebelumnya, Gingerbread, dapat di-update ke *Android* Ice Cream Sandwich.

k. Android versi 4.1 (Jelly Bean)

*Android Jelly Bean* yang diluncurkan pada acara Google I/O lalu membawa sejumlah keunggulan dan fitur baru. Penambahan baru diantaranya meningkatkan input keyboard, desain baru fitur pencarian, UI yang baru dan pencarian melalui *Voice Search* yang lebih cepat. Tak ketinggalan *Google Now* juga menjadi bagian yang diperbarui. *Google Now* memberikan informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu-lintas, ataupun hasil pertandingan olahraga. Sistem operasi *Android Jelly Bean 4.1* muncul pertama kali dalam produk tablet Asus, yakni Google Nexus 7.

l. Android versi 4.2 (Jelly Bean)

Fitur *photo sphere* untuk panorama, *daydream* sebagai *screensaver*, *power control*, *lock screen widget*, menjalankan banyak user (dalam tablet saja), *widget* terbaru.

m. Android Versi 4.3 (Jelly Bean)

Adanya Fitur Emoji Keyboard, update aplikasi kamera, lebih responsif dan smooth.

n. Android Versi 4.4 (Kit Kat)

*Android 4.4 KitKat* API level 19. Google mengumumkan *Android 4.4 KitKat* (dinamai dengan izin dari Nestle dan Hershey) pada 3 September 2013, dengan tanggal rilis 31 Oktober 2013. Kitkat merupakan merk sebuah cokelat yang dikeluarkan Nestle. Rilis berikutnya setelah KitKat diperkirakan banyak pengamat akan diberi nomor 5.0 dan dinamai 'Key Lime Pie'.

o. Android Versi 5.0 (Lollipop)

*Android Lollipop 5.0 dan 5.1.1* API level 21. Versi Lollipop pertama kali diperkenalkan pada 25 Juni 2014 pada saat Google I/O conference. Versi Lollipop mulai diluncurkan sebagai OTA Update pada 12 November 2014.

p. Android Versi 6.0 (Marshmallow)

*Android* versi 6.0 atau *android Marshmallow* merupakan nama terbaru di keluarga *OS android*. Google akhirnya resmi menamai versi *android* teranyarnya dengan nama Marshmallow dan menggunakan API level 23 pada versi preview. Sejauh ini *Android marshmallow* baru versi preview dan belum ada release versionnya. Dikabarkan versi ini baru akan rilis untuk konsumen umum sekitar bulan september atau oktober tahun ini.

### 2.5.2 Keunggulan Android

Menurut Meier (2011), ada beberapa keunggulan dan fitur-fitur yang unik yang ada di *Android*, yaitu:

1. Aplikasi Google Maps

Sekarang ini aplikasi Google Maps yang disediakan untuk mobile sangat populer dan *Android* menawarkan aplikasi Google Maps yang bisa digunakan kembali untuk aplikasi lain.

2. Background services and applications  
Background services memungkinkan developer untuk membuat sebuah aplikasi yang menggunakan event, berjalan tanpa terlihat bahwa aplikasi tersebut sedang berjalan ketika aplikasi lain sedang dijalankan.
3. Sharing data dan *interprocess communication* (IPC)  
Dengan menggunakan Content Provider, *Android* memungkinkan pertukaran pesan, melakukan pengolahan, dan berbagi data.
4. Semua aplikasi yang dibuat sama *Android* tidak membedakan antara aplikasi yang asli dengan yang dikembangkan oleh pihak ketiga (*third parties*).

### 2.5.3 Jenis Aplikasi Android

Menurut Meier (2011), sebagian besar aplikasi yang dibuat dalam *Android* akan termasuk dalam salah satu kategori berikut:

1. Foreground  
Sebuah aplikasi yang berguna hanya ketika ada pada *foreground* dan tidak efektif ketika tidak terlihat. Contoh umumnya adalah *games* dan *map mashup*.
2. Background  
Sebuah aplikasi dengan interaksi yang terbatas, kecuali ketika dikonfigurasi, sebagian besar masa waktunya tersembunyi. Contohnya adalah aplikasi call screening dan *Short Message Service* (SMS) *auto-responder*.
3. Intermittent  
Sebuah aplikasi yang dapat jalan di *foreground* dan *background*, tetapi sering kali aplikasi ini akan dibentuk dan kemudian dijalankan tersembunyi (*background*) dan memberitahu pengguna di saat yang tepat. Contohnya adalah *media player*.
4. Widget  
Aplikasi yang dibuat hanya ditampilkan sebagai widget pada screen home.

### 2.5.4 jenis-jenis Komponen Android

Di android sendiri mempunyai komponen ,jumlah komponennya sebanyak 4 di antara lain:

#### 1. Activity

Bagian dari sebuah aplikasi yang di pakai untuk berintraksi dengan pengguna aplikasi. Sebuah activity mempunyai user interface, misalnya aplikasi Gmail saat menampilkan inbox atau music player saat menampilkan daftar lagu, atau pada deretan menu sebuah game

2. Service  
Service tidak memiliki user interface, namun berjalan secara background. Misalnya music player, sambil memutar lagu pengguna aplikasi tetap bisa menjelaskan aplikasi lainnya.
3. Broadcast Receiver  
Bagian ini di pakai untuk menerima isyarat dari system android, misalnya saat batrai low atau screen off
4. Content Providers  
Dengan Content Providers, data sebuah aplikasi bisa diakses atau di gunakan dari aplikasi lainnya, misalnya informasi kontak bisa diakses oleh aplikasi apa saja yang memiliki izin akses.

## 2.6 Android Studio

*Android Studio* adalah sebuah IDE yang bisa digunakan untuk pengembangan aplikasi *Android*, dan dikembangkan oleh Google. *Android Studio* merupakan pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA. *Android Studio* direncanakan untuk menggantikan Eclipse ke depannya sebagai IDE resmi untuk pengembangan aplikasi *Android*

### 2.6.1 Android Software Development Kit (SDK)

*Android SDK* adalah tool API (application Programming Interface) yang diperlukan untuk mulai mengembangkan aplikasi pada platform *Android* menggunakan bahasa pemrograman Java. *Android* merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, middleware dan aplikasi kunci yang release oleh Google. Saat ini di sediakan *Android SDK* (Software Development Kit) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform *Android* menggunakan bahasa pemrograman java (Safaat H 2011 : 5).

### 2.6.2 Android Development Tools (ADT)

*Android Development Tools* (ADT) yang lebih dikenal plugin *Eclipse*. plugin ini yang membuat *eclipse* dapat membuat project yang berbasis *android*. ADT adalah plugins di *eclipse* yang harus kita instal sehingga *Android SDK* dapat dihubungkan dengan IDE Eclipse yang digunakan sebagai tempat coding aplikasi *android* nantinya. ADT adalah kepanjangan dari *Android Development Tolls* yang menghubungkan antara IDE Eclipse dengan *Android SDK* (Safaat H 2011 : 17).

### 2.6.3 Android Virtual Device (AVD)

AVD yang merupakan emulator untuk menjalankan program aplikasi *android* yang kita buat, AVD ini nantinya yang kita jadikan sebagai tempat test dan menjalankan aplikasi *android* yang kita buat, AVD berjalan di Virtual Machine (Safaat H 2011 : 19).

## 2.7 UML (Unified Modeling Language)

*Unified Modeling Language* merupakan salah satu alat bantu yang dapat digunakan dalam **bahasa pemrograman** yang berorientasi objek, saat ini UML akan mulai menjadi standar masa depan bagi industri pengembangan sistem/perangkat lunak yang berorientasi objek sebab pada dasarnya UML digunakan oleh banyak perusahaan raksasa seperti IBM, *Microsoft*, dan sebagainya.

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu :

1. *Use Case Diagram* untuk memodelkan proses bisnis.
2. *Conceptual Diagram* untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. *Sequence Diagram* untuk memodelkan pengiriman pesan (message) antar *objects*.
4. *Collaboration Diagram* untuk memodelkan interaksi antar *objects*.
5. *State Diagram* untuk memodelkan perilaku *objects* di dalam sistem.
6. *Activity Diagram* untuk memodelkan perilaku *Use Cases* dan *objects* di dalam *system*.
7. *Class Diagram* untuk memodelkan struktur kelas.
8. *Object Diagram* untuk memodelkan struktur *object*.
9. *Component Diagram* untuk memodelkan komponen *object*.
10. *Deployment Diagram* untuk memodelkan distribusi aplikasi.

Empat macam diagram yang paling sering digunakan dalam pembangunan aplikasi berorientasi *object*, yaitu *use case diagram*, *sequence diagram*, *activity diagram*, dan *class diagram*.


### 2.7.1 Use Case Diagram

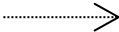
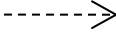



*Use case diagram* digunakan untuk memodelkan bisnis proses *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang

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

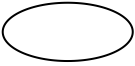


Seorang aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

Tabel 2.1 Simbol *Use Case Diagram*

No	Gambar	Nama	Keterangan
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .

2		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
3		<i>Include</i>	Menspesifikasikan bahwa use case sumber secara eksplisit.
<b>No</b>	<b>Gambar</b>	<b>Nama</b>	<b>Keterangan</b>
4		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
5		<i>Association</i>	Apa yang menghubungkan antara objek.
6		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas

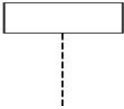




7		<i>Use case</i>	Dskripsi dari urutan aksi – aksi yang ditampilkan sistem menghasilkan suatu hasil yang terukur.
<b>No</b>	<b>Gambar</b>	<b>Nama</b>	<b>Keterangan</b>
8		<i>Collaboration</i>	Interaksi aturan- aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
9		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan

### 2.7.2 Sequence Diagram

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline vertical*.

Tabel 2.2 Simbol *Sequence Diagram*

No	Gambar	Nama	Keterangan
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi






### 2.7.3 Activity Diagram

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. *Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu. Untuk

mengilustrasikan proses-proses paralel (fork dan join) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.





Tabel 2.3 Simbol *Activity diagram*


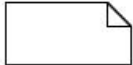
No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
No	Gambar	Nama	Keterangan
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

### 2.7.4 State Diagram

State diagram digunakan untuk mendeskripsikan perilaku sistem. State diagram mendeskripsikan semua kondisi yang mungkin muncul sebagai sebuah object begitu pula dengan event.

Tabel 2.4 Simbol *State diagram*

No	Gambar	Nama	Keterangan
1		Status awal/Kondisi awal	Status awal alur sebuah objek, sebuah diagram status memiliki sebuah status awal
2		Status	Status yang dialami objek selama hidupnya
3		Status akhir/kondisi akhir	Kondisi akhir alur hidup objek, sebuah diagram status memiliki sebuah status akhir
4		Transisi	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbarui satu atau lebih nilai atribut
5		Asosiasi	Apa yang menghubungkan antara objek satu

			dengan objek lainnya
6		Node	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi