BAB 2

TINJAUAN PUSTAKA

2.1. Pengertian *Game*

Game merupakan kata dalam bahasa inggris yang berarti permainan. Permainan adalah sesuatu yang dapat dimainkan dengan aturan tertentu sehingga ada yang menang dan ada yang kalah, biasanya dalam konteks tidak serius atau dengan tujuan refreshing[6]. Suatu cara belajar yang digunakan dalam menganalisa interaksi antara sejumlah pemain maupun perorangan yang menunjukkan strategi-strategi yang rasional. Teori permainan pertama kali ditemukan oleh sekelompok ahli Matematika pada tahun 1944. Teori itu dikemukakan oleh John von Neumann and Oskar Morgenstern yang berisi:

"Permainan terdiri atas sekumpulan peraturan yang membangun situasi bersaing dari dua sampai beberapa orang atau kelompok dengan memilih strategi yang dibangun untuk memaksimalkan kemenangan sendiri atau pun untuk meminimalkan kemenangan lawan. Peraturan-peraturan menentukan kemungkinan tindakan untuk setiap pemain, sejumlah keterangan diterima setiap pemain sebagai kemajuan bermain, dan sejumlah kemenangan atau kekalahan dalam berbagai situasi". (Theory of Games and Economic Behavior (3d ed. 1953))

2.2. Permainana Kartu Kwartet

Kartu kwartet adalah permainan kartu yang banyak dimanfaatkan sebagai media pembelajaran yang terdiri atas beberapa jumlah kartu bergambar. Dari kartu tersebut tertera keterangan berupa tulisan yang menerangkan gambar tersebut. Biasanya tulisan judul (kategori) gambar ditulis paling atas dari kartu dan

tulisannya lebih diperbesar atau dipertebal, sedangkan tulisan lainnya (sub kategori) ditulis setelah judul dari kartu kwartet[7]. Ukuran dari kartu kwartet ini biasanya beragam, ada yang kurannya kecil sedang. Jumlah kartu kwartet dalam penelitian ini berjumlah 32 buah judul kartu, dan masing-masing memiliki 4 buah subjudul, seperti pada gambar 2.1:



Gambar 2.1 Kartu Kwartet

2.3. AI (*Artficial Intelligence*)

Kecerdasan buatan sering disebut juga dengan *Artificial Intelligence* (AI). AI merupakan salah satu bagian ilmu komputer yang mempelajari tentang bagaimana caranya agar komputer dapat melakukan pekerjaan seperti yang dapat dilakukan oleh manusia[8]. Menurut John McCarthy, 1956, AI digunakan untuk mengetahui dan memodelkan proses – proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia. Pada awal diciptakannya, komputer hanya difungsikan sebagai alat hitung saja. Namun seiring dengan perkembangan zaman, teknologi komputer semakin ditingkatkan dan peran komputer semakin mendominasi kehidupan umat manusia. Komputer tidak lagi

hanya digunakan sebagai alat hitung, melainkan komputer diharapkan dapat diberdayakan untuk mengerjakan hal - hal yang dapat memudahkan pekerjaan manusia.

Agar komputer bisa bertindak seperti apa yang dapat dilakukan manusia, maka komputer juga harus diberi bekal pengetahuan dan mempunyai kemampuan untuk menalar. Untuk itu AI akan mencoba untuk memberikan beberapa metode untuk membekali komputer agar komputer bisa menjadi mesin yang pintar. Dari beberapa perspektif, AI dapat dipandang sebagai berikut:

1. Dari perspektif kecerdasan, AI adalah bagaimana membuat mesin yang cerdas dan dapat melakukan hal-hal yang sebelumnya hanya dapat dilakukan manusia.
2. Dari perspektif bisnis, AI adalah sekelompok alat bantu (tools) yang berdayaguna dan metodologi yang menggunakan alat-alat bantu tersebut untuk menyelesaikan masalah-masalah bisnis.
3. Dari perspektif pemrograman, AI meliputi studi tentang pemrograman simbolik, pemecahan masalah, dan proses pencarian (search).
4. Dari perspektif penelitian, riset tentang AI dimulai pada awal tahun 1960-an, percobaan pertama adalah membuat program permainan catur, membuktikan teori, dan general problem solving.

2.4. Skala Likert

Skala Likert adalah suatu skala psikometrik yang umum digunakan dalam kuisioner, dan merupakan skala yang paling banyak digunakan dalam riset berupa survei. Skala likert menggunakan beberapa butir pertanyaan untuk mengukur perilaku individu dengan merespon 5 titik pilihan pada setiap butir pertanyaan,

sangat setuju, setuju, tidak memutuskan, tidak setuju, dan sangat tidak setuju[9]. Nama skala ini diambil dari nama Rensis Likert, yang menerbitkan suatu laporan yang menjelaskan penggunaannya. Sewaktu menanggapi pertanyaan dalam skala likert, responden menentukan tingkat persetujuan mereka terhadap suatu pernyataan dengan memilih salah satu dari pilihan yang tersedia. Biasanya disediakan lima pilihan skala dengan format seperti:

1 Sangat setuju dengan skor 5,
2 Setuju dengan skor 4
3 Ragu–ragu dengan skor 3
4 Tidak setuju dengan skor 2
5 Sangat tidak setuju dengan skor 1.

Untuk mencari nilai persentase dari setiap pertanyaan kuesioner digunakan rumus skala likert sebagai berikut :

$$P = \frac{\sum S}{skorideal} x100$$

Keterangan :

P = Tingkat persetujuan dari setiap pertanyaan dalam bentuk persentase

S = Skoring setiap jawaban dari responden yang didapatkan dari perhitungan nilai frekuensi dikalikan dengan nilai skor pada setiap jawaban

Σs = Jumlah skor yang diperoleh dari nilai skoring setiap jawaban

Skorideal = Nilai skor tertinggi dari jawaban dikalikan dengan jumlah frekuensi secara keseluruhan.

Sebagai contoh misalnya suatu perusahaan XYZ ingin mengetahui seberapa persen karyawannya setuju akan penerapan prosedur kerja yang baru kepada 100 orang karyawan dan diambil secara random.

Berilah jawaban pertanyaan berikut sesuai dengan pernyataan anda, dengan cara memberi tanda (√) pada kolom yang tersedia.

Tabel 2.1 Tabel Pertanyaan Skala Likert

No	Pertanyaan	SS	ST	RG	TS	STS
1.	Prosedur kerja yang baru itu akan segera diterapkan di perusahaan anda		√			
2.						

SS = Sangat Setuju diberi skor 5 ST = Setuju diberi skor 4

RG = Ragu-ragu diberi skor 3 TS = Tidak Setuju diberi skor 2

STS = Sangat Tidak Setuju diberi skor 1

Dari hasil kuisioner terhadap 100 pegawai didapat hasil seperti berikut :

25 Orang menjawab SS

40 Orang menjawab ST

5 Orang menjawab RG

20 Orang menjawab TS

10 Orang menjawab STS

Berdasarkan data tersebut 65 orang (40+25) atau 65% karyawan menjawab setuju dan sangat setuju. Jadi kesimpulannya mayoritas karyawan setuju adanya metode kerja baru.

Data interval tersebut juga dapat dianalisis dengan menghitung rata-rata jawaban berdasarkan skoring setiap jawaban dari responden. Berdasarkan skor yang telah ditetapkan dapat dihitung sebagai berikut :

Jumlah skor untuk 25 orang yang menjawab SS = 25 x 5= 125

Jumlah skor untuk 40 orang yang menjawab ST = 40 x 4= 160

Jumlah skor untuk 5 orang yang menjawab RG = 5 x 3 = 15

Jumlah skor untuk 20 orang yang menjawab TS = 20 x 2= 40

Jumlah skor untuk 10 orang yang menjawab STS = 10 x 1= 10

Jumlah total = 350

Jumlah skor ideal (kriterium) untuk seluruh item = 5 x 100 = 500 (seandainya semua menjawab SS). Jumlah skor yang diperoleh dari penelitian 350. Jadi berdasarkan data tersebut maka tingkat persetujuan terhadap prosedur kerja yang baru itu

(350 : 500) x 100% = 70%

dari yang diharapkan (100%). Jadi berdasarkan data yang diperoleh dari 100 responden maka rata-rata 350 terletak pada daerah setuju.

2.5. Metode Probabilitas

Probabilitas didifinisikan sebagai peluang atau kemungkinan suatu kejadian, suatu ukuran tentang kemungkinan atau derajat ketidakpastian suatu peristiwa (event) yang akan terjadi di masa mendatang. Sebagai contoh ketika kita melihat langit dalam keadaan mendung, awan berubah warna menjadi gelap,

angin lebih kencang dari biasanya serta sinar matahari tidak seterang biasanya. Dalam keadaaan ini kita dihadapkan antara 2 permasalahan, yaitu kemungkinan terjadinya hujan serta kemungkinan langit hanya mendung saja dan tidak akan turunnya hujan. Statistik yang membantu permasalahan dalam hal ini adalah probabilitas.

Manfaat probabilitas dalam kehidupan sehari-hari adalah membantu kita dalam mengambil suatu keputusan, serta meramalkan kejadian yang mungkin terjadi. Jika kita tinjau pada saat kita melakukan penelitian, probabilitas memiliki beberapa fungsi antara lain:

1 Membantu peneliti dalam pengambilan keputusan yang lebih tepat. Pengambilan keputusan yang lebih tepat dimagsudkan tidak ada keputusan yang sudah pasti karena kehidupan mendatang tidak ada yang pasti kita ketahui dari sekarang, karena informasi yang didapat tidaklah sempurna.
2 Dengan teori probabilitas kita dapat menarik kesimpulan secara tepat atas hipotesis yang terkait tentang karakteristik populasi.
3 Menarik kesimpulan secara tepat atas hipotesis (perkiraan sementara yang belum teruji kebenarannya) yang terkait tentang karakteristik populasi pada situssi ini kita hanya mengambil atau menarik kesimpulan dari hipotesis bukan berarti kejadian yang akan dating kita sudah ketehaui apa yang akan tertjadi.
4 Mengukur derajat ketidak pastian dari analisis sampel hasil penelitian dari suatu populasi.

Berikut ini merupakan rumus 2.1 yaitu rumus probabilitas dalam menghitung peluang suatu kejadian[10]:

$$P(A) = n(A) / n(S)$$

Dengan:

n(A) = banyaknya kejadian A yang diharapkan muncul

n(S) = banyaknya ruang sampel

Sebagai contoh, sebuah kartu dipilih secara acak dari satu set kartu bridge. Tentukan peluang munculnya kartu yang terpilih bukan kartu As

Pembahasan :

Misal A = kejadian terpilihnya kartu As

A' = kejadian terpilihnya kartu bukan As

n(S) = 52

n(A) = 4

P(A) = n(A) / n(S)

= 4 / 52

= 1 - P(A)

= 48 / 52

Jadi, peluang munculnya kartu yang terpilih bukan kartu As = 48 / 52

2.6. Pemograman Berorientasi Objek

Pemograman berorientasi objek (*Obhect oriented programming – OOP*) merupakan paradigma pemograman yang berorientsikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya,[11]

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

Pemrograman berorientasiobjek menekankan konsep berikut:

1. kelas, kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh 'class of dog' adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi oobjek. Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.
2. Objek, membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer. Objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.
3. Abstraksi , kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak

yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.

4. Enkapsulasi, memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi ijin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.
5. Polimorfisme melalui pengiriman pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan; metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim. Contohnya, bila sebuah burung menerima pesan "gerak cepat", dia akan menggerakan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut polimorfisme karena sebuah variabel tungal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama. Hal ini berlawanan denganbahasa fungsional yang mencapai polimorfisme melalui penggunaan fungsi kelas-pertama.

6. Inheritas- Mengatur polimorfisme dan enkapsulasi dengan mengijinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada - objek-objek ini dapat membagi (dan memperluas) perilaku mereka tanpa haru mengimplementasi ulang perilaku tersebut (bahasa berbasis-objek tidak selalu memiliki inheritas).
7. Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sebagai contoh anggap kita memiliki sebuah departemen yang memiliki manager, sekretaris, petugas administrasi data dan lainnya. Misal manager tersebut ingin memperoleh data dari bag administrasi maka manager tersebut tidak harus mengambilnya langsung tetapi dapat menyuruh petugas bag administrasi untuk mengambilnya. Pada kasus tersebut seorang manager tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi manager bisa mendapatkan data tersebut melalui objek petugas adminiistrasi. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek-objek yang ada karena setiap objek memiliki deskripsi tugasnya sendiri.

2.7. Bahasa Pemograman Java

Java adalah bahasa pemograman yang dapat dijalankan di berbagai berbagai komputer termasuk telepon genggam. Dikembangkan oleh Sun Microsystems dan dirilis tahun 1995[12]. Java berbeda dengan JavaScript. JavaScript adalah bahasa scripting yang digunakan oleh web browser seperti pada gambar 2.2:



Gambar 2.2 Logo Java

Bahasa pemrograman Java pertama lahir dari The Green Project, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan versi yang dinamakan Oak. Proyek ini dimotori oleh Patrick Naughton, Mike Sheridan,James Gosling dan Bill Joy, beserta sembilan pemrogram lainnya dari Sun Microsystems. Salah satu hasil proyek ini adalah mascot Duke yang dibuat oleh Joe Palrang.

Nama Oak, diambil dari pohon oak yang tumbuh di depan jendela ruangan kerja "Bapak Java", James Gosling. Nama Oak ini tidak dipakai untuk versi release Java karena sebuah perangkat lunak lain sudah terdaftar dengan merek dagang tersebut, sehingga diambil nama penggantinya menjadi "Java". Nama ini diambil dari kopi murni yang digiling langsung dari biji (kopi tubruk) kesukaan Gosling. Konon kopi ini berasal dari Pulau Jawa. Jadi nama bahasa pemrograman Java tidak lain berasal dari kata Jawa (bahasa Inggris untuk Jawa adalah Java).

2.8. UML (*Unified Modeling Language*)

Unified Modeling Language (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya[13]. UML adalah metodologi untuk mengembangkansistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut UML mulai diperkenalkan oleh Objek Management Group, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (tool) desain berorientasi objek dari IBM

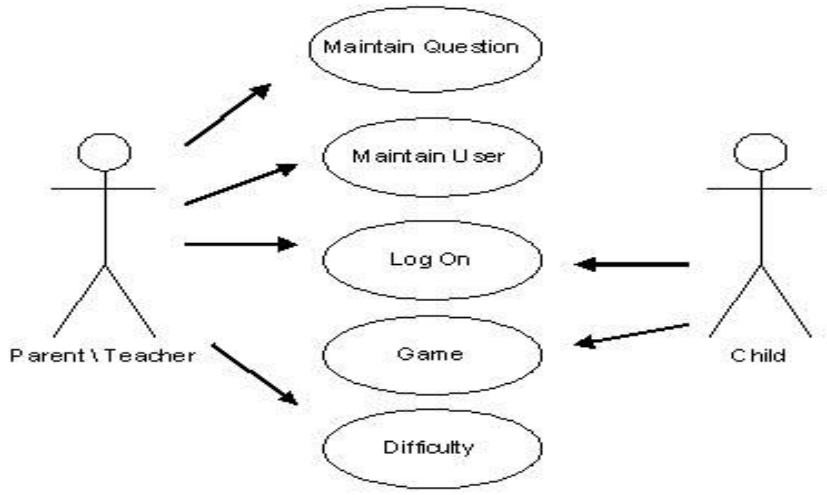
UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson.Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi.Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.

2.8.1. Use *Case*

Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. Use case diagram terdiri atas diagram untuk use case dan actor. Actor merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi.

Use case merepresentasikan operasi-operasi yang dilakukan oleh actor. Use case digambarkan berbentuk elips dengan nama operasi dituliskan

didalamnya. Actor yang melakukan operasi dihubungkan dengan garis lurus ke use case, seperti pada gambar 2.3:



Gambar 2.3 Use Case Diagram

2.8.2. Use *Case Scenario*

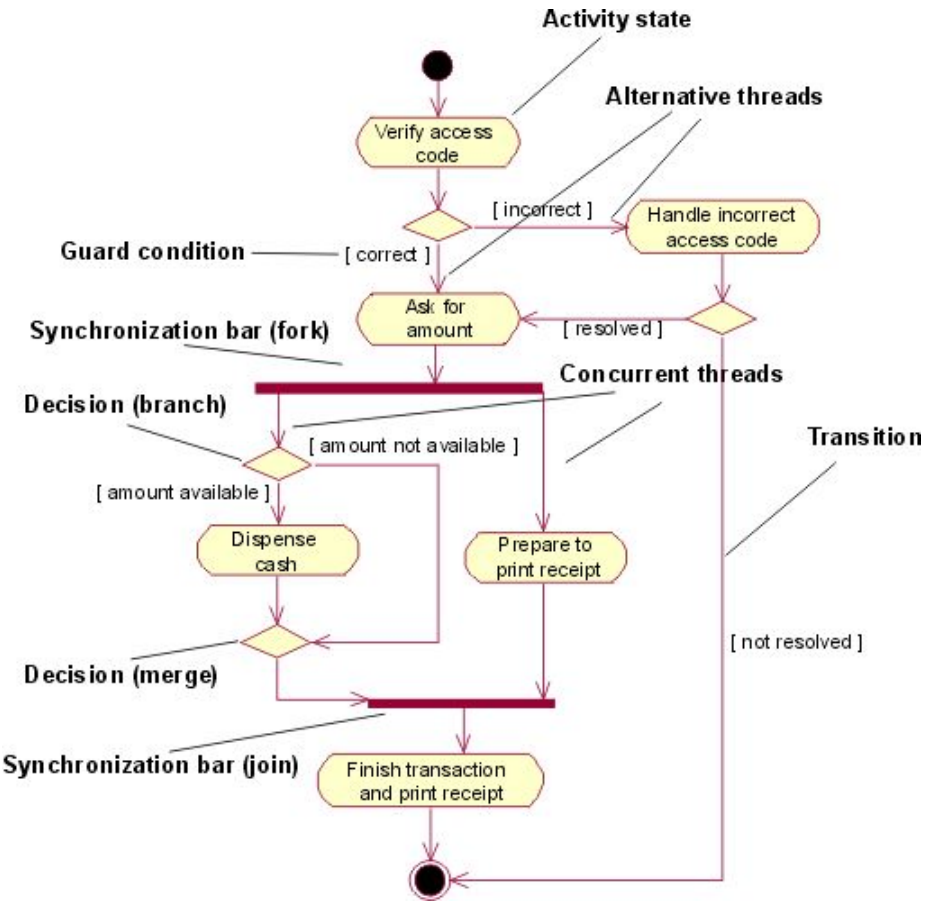
Use Case Scenario adalah sebuah dokumentasi terhadap kebutuhan fungsional dari sebuah sistem. Setiap use case harus dijelaskan alur prosesnya melalui sebuah use case skenario. Use case skenario berisi:

1. Nama use case yaitu penamaan use case yang menggunakan kata kerja
2. Deskripsi yaitu penjelasan mengenai tujuan use case dan nilai yang akan didapatkan oleh aktor
3. Kondisi sebelum (pre-condition) yaitu kondisi-kondisi yang perlu ada sebelum use case dilakukan.
4. Kondisi sesudah (post-condition) yaitu kondisi-kondisi yang sudah dipenuhi ketika uses case sudah dilaksanakan
5. Alur dasar (basic flow) yaitu alur yang menceritakan jika semua aksi yang dilakukan adalah benar atau proses yang harusnya terjadi

6. Alur alternatif (alternatif flow) yaitu alur yang menceritakan aksi alternatif, yang berbeda dari alur dasar

2.8.3. Activity *Diagram*

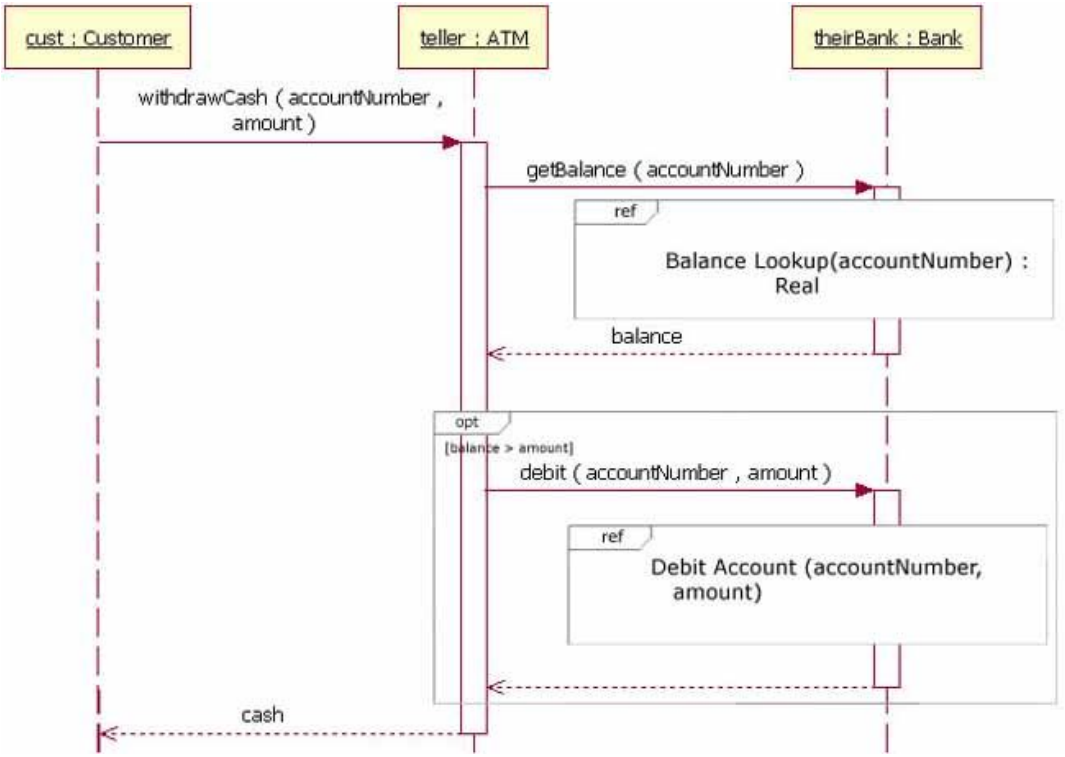
Diagram aktivitas atau dalam bahasa inggris activity diagram adalah representasi grafis dari seluruh tahapan alur kerja. Diagram ini mengandung aktivitas, pilihan tindakan, perulangan dan hasil dari aktivitas tersebut. Pada pemodelan UML, diagram ini dapat digunakan untuk menjelaskan proses bisnis dan alur kerja operasional secara langkah demi langkah dari komponen suatu sistem, seperti pada gambar 2.4:



Gambar 2.4 Activity Diagram

2.8.4. Sequence *diagram*

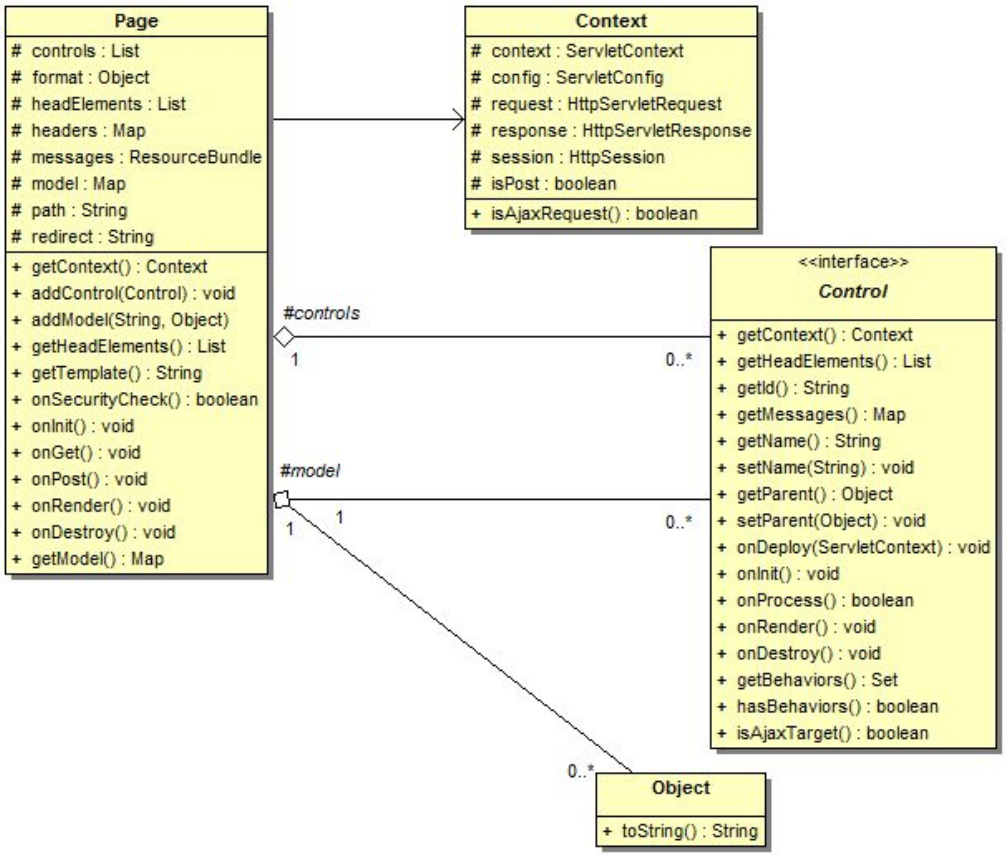
Sequence diagram menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari use case. Interaksi yang terjadi antar class, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi, seperti pada gambar 2.5:



Gambar 2.5 Sequence Diagram

2.8.5. Class *Diagram*

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. Class diagram menunjukkan hubungan antar class dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan, seperti pada gambar 2.6:



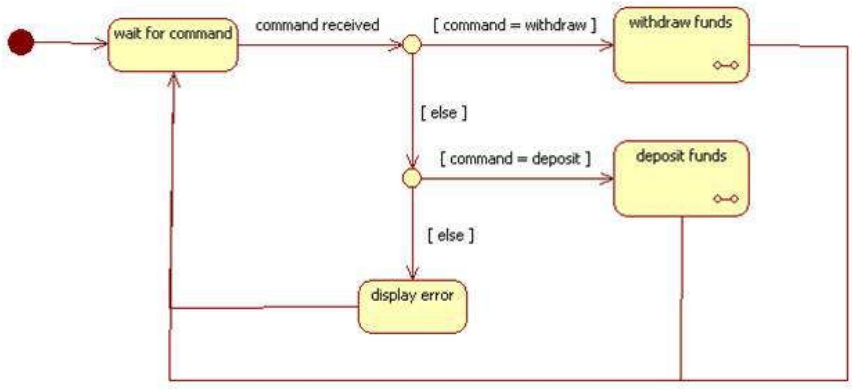
Gambar 2.6 Class Diagram

2.8.6. State *Chart Diagram*

State Chart diagram adalah diagram yang menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) suatu keadaan (dari satu state ke state lainnya) suatu obyek pada sistem sebagai akibat dari stimulans yang diterima, yang berfungsi sebagai:

1. Untuk memodelkan behavior/methode (lifecycle) sebuah kelas atau object
2. Memperlihatkan urutan kejadian sesaat (state) yang dilalui sebuah object, transisi dari sebuah state ke state lainnya

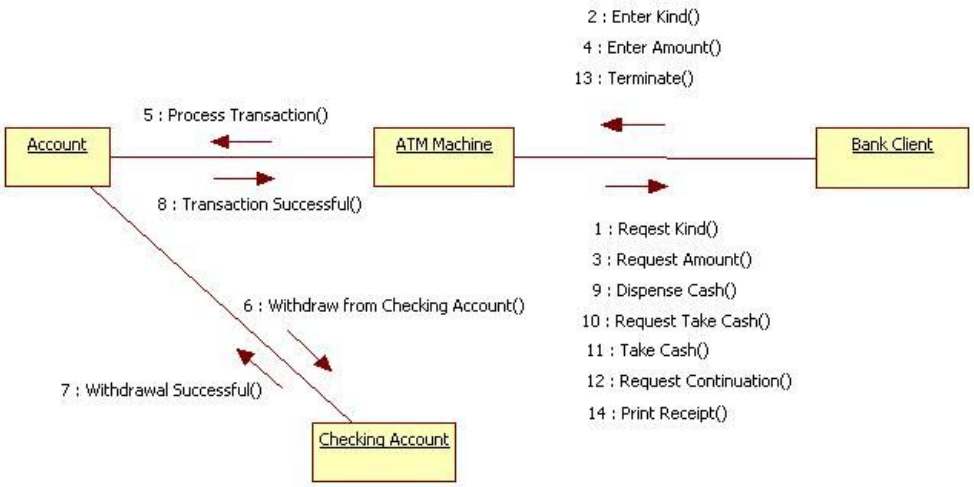
Berikut ini gambar 2.7 yang merupakan contoh dari state chart diagram:



Gambar 2.7 State Chart Diagram

2.8.7. Collaboration *Diagram*

Collaboration diagram dipakai untuk memodelkan interaksi antar objek di dalam sistem.Berbeda dengan sequence diagram yang lebih menonjolkan kronologis dari operasi-operasi yang dilakukan, collaboration diagram lebih fokus pada pemahaman atas keseluruhan operasi yang dilakukan oleh objek, seperti pada gambar 2.8:



Gambar 2.8 Colaboration Diagram

2.9. Sistem Operasi Android

Android merupakan platform yang berjalan diatas sistem operasi Linux. Android pertama kali dikembangkan Google dan Open Handset Alliance. Bahasa pemrograman dasar yang dipakai untuk pengembangan aplikasi di atas Android adalah Java dengan pemanfaatan Google developed Java libraries[14].

2.9.1. Sejarah Android

Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru, sehingga Sistem Operasi Android terdiri dari beberapa versi, antara lain:

1. Android versi 1.1

   Pada 9 Februari 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, voice search (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

2. Android Cupcake (1.5)

   Hanya selang 1 bulan Google merilis Android 1.5 (Cupcake) dengan SDK (Software Development Kit). Terdapat beberapa pembaharuan termasuk juga penambahan beberapa fitur dalam seluler yakni kemampuan merekam dan menonton video dengan modus kamera, menggugah video ke youtube dan gambar ke Picasa langsung dari telepon, dukungan bluetooth A2DP, kemampuan terhubung secara otomatis ke headset Bluetooth , animasi layar, dan keyboard pada layar yang dapat disesuai dengan system.

3. Android 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. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus, kamera, camcorder, dan galeri yang diintegrasikan : CDMA/EVDO, 802.1x, VPN, Gestures dan Text to speech engine, kemampuan dial kontak, teknologi text to change speech (tidak tersedia dalam ponsel), pengadaan resolusi VWGA.

4. Android Eclair (2.0 / 2.1)

Versi ini dirilis 3 desember 2009 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. Perubahan yang lain yaitu : pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru , daftar kontak yang baru, dukungan flash untuk kamera 3.2 MP, digital zoom, dan Bluetooth 2.1.

5. Android Froyo / Frozen Yoghurt (2.2)

Versi ini dirilis 20 mei 2010, android 2.2 dirilis dengan 20 fitur baru, antara lain peningkatan kecepatan kinerja, fitur Wi-Fi hotspot tethering dan dukungan terhadap Adobe Flash 10.1, aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 Javascript engine yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD card, kemampuan WiFi Hotspot portable, dan kemampuan auto update dalam aplikasi Android Market.

6. Android Gingerbread (2.3)

Versi ini dirilis 6 desember 2010. Perubahan utama di versi 2.3 ini termasuk update UI, peningkatan fitur soft keyboard & copy/paste, power

management, dan support Near Field Communication, peningkatan kemampuan permainan (gaming), layanan antarmuka (User Interface) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (reverb, equalization, headphone virtualization dan bass boost), dan dukungan jumlah kamera yang lebih dari satu.

7. Android 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 prosessor multicore dan grafis dengan hardware acceleration.

8. Android Ice Cream Sandwich (4.0)

Android 4.0 Ice Cream Sandwich diumumkan pada 10 Mei 2011 di ajang Google 26 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), perangkat tambahan fotografi, mencari email secara offline dan berbagai informasi dengan menggunakan NFC.

9. Android Jelly Bean (4.1)

2.9.2. Fitur-fitur di Android

Berikut ini adalah fitu-fitur yang ada pada Android:

1. Framework Aplikasi yang mendukung penggantian komponen dan reusable
2. Mesin virtual Dalvik dioptimalkan untuk perangkat mobile

3. Integrated browser berdasarkan engine open source WebKit
4. Grafis yang dioptimalkan dan didukung oleh library grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (Opsional akselerasi hardware).
5. SQLite untuk penyimpanan data
6. Media Support yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
7. GSM Telephony (tergantung hardware)
8. Bluetooth, EDGE, 3G, dan WiFi (tergantung hardware)
9. Kamera, GPS, kompas, dan accelerometer (tergantung hardware)
10. Lingkungan Development yang lengkap dan kaya termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk Eclipse IDE.
11. Dukungna Perangkat Tambahan, Android dapat memanfaatkan kamera, layar sentuh, accelerometers, magnetometers, GPS, akselerasi 2D (dengan perangkat orientasi, scalling,konversi format piksel), dan akselerasi grafis 3D.
12. Multi-Touch, kemampuan layaknya handset modern yang dapat menggunakan dua jari atau lebih untuk berinteraksi dengan perangkat.
13. Market, handphone memiliki tempat penjualan aplikasi, market pada android merupakan katalog aplikasi yang dapat di download dan di-install melalui internet.

2.10. Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*)[15]. Berikut ini adalah sifat dari Eclipse:

1. Multi-platform: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
2. Mulit-language: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
3. Multi-role: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.
4. Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

2.10.1. Sejarah Eclipse

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak IBM Visual Age for Java 4.0. Produk ini diluncurkan oleh IBM pada tanggal 5 November 2001, yang menginvestasikan sebanyak US$ 40 juta[1] untuk

pengembangannya. Semenjak itu konsursium Eclipse Foundation mengambil alih untuk pengembangan Eclipse lebih lanjut dan pengaturan organisasinya.

2.10.2. Arsitektur Eclipse

Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah *kernel*, yang mengangkat *plug-in*. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari *plug-in* yang sudah diinstal. Ini merupakan basis dari Eclipse yang dinamakan *Rich Client Platform* (RCP). Berikut ini adalah komponen yang membentuk RCP:

1. *Core platform*
2. OSGi
3. SWT (*Standard Widget Toolkit*)
4. JFace
5. *Eclipse Workbench*

Secara standar Eclipse selalu dilengkapi dengan JDT (*Java Development Tools*), *plug-in* yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan PDE (*Plug-in Development Environment*) untuk mengembangkan *plug-in* baru. Eclipse beserta *plug-in*-nya diimplementasikan dalam bahasa pemrograman Java.

Konsep Eclipse adalah IDE yang terbuka (open), mudah diperluas (extensible) untuk apa saja, dan tidak untuk sesuatu yang spesifik. Jadi, Eclipse tidak saja untuk mengembangkan program Java, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal plug-in yang dibutuhkan. Apabila ingin mengembangkan program C/C++ terdapat plug-in CDT (C/C++ Development Tools). Selain itu, pengembangan secara visual bukan hal

yang tidak mungkin oleh Eclipse, plug-in UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat plug-in sesuai dengan keinginannya.

2.10.3. Perkembangan Eclipse

Sejak tahun 2006, Eclipse Foundation mengkoordinasikan peluncuran Eclipse secara rutin dan simultan yang dikenal dengan nama *Simultaneous Release*. Setiap versi peluncuran terdiri dari Eclipse Platform dan juga sejumlah proyek yang terlibat dalam proyek Eclipse. Tujuan dari sistem ini adalah untuk menyediakan distribusi Eclipse dengan fitur-fitur dan versi yang terstandarisasi. Hal ini juga dimaksudkan untuk mempermudah deployment dan maintenance untuk sistem enterprise. Adapun versi eclipse yang telah diluncurkan yaitu :

1. Eclipse 3.0
2. Eclipse 3.1
3. Callisto
4. Europa
5. Ganymede
6. Galileo
7. Helios

2.10.4. Software *Development Kit* (SDK)

Software Development Kit (SDK) adalah suatu kit atau library dari bahasa pemrograman untuk pengembangan atau pembangunan suatu perangkat lunak dan biasanya SDK terdiri dari kumpulan *tools* yang dibutuhkan. Misalnya bahasa pemrograman java, mempunyai SDK yang berisi suatu library yang dapat digunakan untuk membuat suatu aplikasi berbasis java.

2.10.5. Java *Development Kit* (JDK)

Java Development Kit (JDK) adalah sekumpulan perangkat lunak yang dapat kamu gunakan untuk mengembangkan perangkat lunak yang berbasis Java, Sedangkan JRE adalah sebuah implementasi dari Java Virtual Machine yang benar-benar digunakan untuk menjalankan program java.Biasanya, setiap JDK berisi satu atau lebih JRE dan berbagai alat pengembangan lain seperti sumber kompiler java, bundling, debuggers, development libraries dan lain sebagainya. Perbedaan JDK dengan SDK (Software Development Kit) yaitu JDK adalah sebuah SDK tetapi sebuah SDK tidak harus menjadi sebuah JDK.

2.10.6. Android *Development Tool* (ADT)

Android Development Tools (*ADT*) adalah *plugin* untuk Eclipse *Intergrated Development Environment* (*IDE*) yang dirancang untuk memberikan lingkungan yang terpadu di mana untuk membangun aplikasi Android.

ADT memperluas kemampuan Eclipse untuk membiarkan para *developer* lebih cepat dalam membuat proyek baru Android, membuat aplikasi UI, menambahkan komponen berdasarkan Android *Framework* API, *debug* aplikasi dalam pengunaan Android SDK, dan membuat file APK untuk mendistribusikan aplikasi.

Mengembangkan aplikasi di Eclipse dengan ADT sangat dianjurkan dan merupakan cara tercepat untuk memulai membuat aplikasi android, karena banyak kemudahan-kemudahan sebagai *tools* yang terintegrasi seperti, custom XML editor, dan *debug panel ouput*. Selain itu ADT memberikan dorongan luar biasa dalam mengembangkan aplikasi Android.

2.11. Library LibGDX

Libgdx merupakan framework java yang menyediakan *cross-platform* API untuk game dan pengembang aplikasi *real-time.* Secara umum, Libgdx memberikan performa yang tinggi dan *framework* pengembang game cross-platform yang digunakan sebagai dasar untuk engine dan game. Pernyataan Libgdx sebagai *framework* mengijinkan kita untuk fokus pada pondasi yang kokoh, malahan mencoba untuk mengimplementasikan arsitektur *game engine* yang terakhir dan yang terhebat[16].

2.12. InkScape

Inkscape adalah sebuah perangkat lunak editor gambar vektor yang bersifat perangkat lunak bebas dibawah lisensi GNU GPL. Tujuan utama dari Inkscape adalah menjadi perangkat grafik mutakhir yang memenuhi standar XML, SVG, dan CSS[17].

Inkscape bersifat cross-platform dan dapat dijalankan pada Mac OS X (biasanya dibawah aplikasi X11, walaupun *toolkit* GTK+ yang digunakan dapat dikompilasikan untuk beroperasi secara langsung dibawah *Quartz*), sistem operasi berbasis *Unix*, dan *Microsoft Windows*. Implementasi SVG dan CSS di *Inkscape* belum sempurna, misalnya *Inkscape* tidak mendukung animasi SVG, dan font SVG, walaupun dukungan dasar untuk pembuatan font SVG telah diimplementasikan pada versi 0.47. *Inkscape* bersifat multibahasa, terutama untuk antarmuka dan script rumit, sesuatu yang sering terlewatkan pada sebagian besar editor grafik vektor komersil.

2.13. Audacity

Audacity adalah aplikasi pemberi efek suara yang terbaik yang pernah ada di dunia open source. Aplikasi ini dibangun dengan pustaka *WxWidgets* sehingga dapat berjalan pada berbagai sistem operasi. Dengan *Audacity*, pengguna bisa mengoreksi data suara tertentu, atau sekedar menambahkan berbagai efek yang disediakan. Selain itu, pengguna juga dapat berkreasi dengan suara yang dimilikinya. Kelebihan dari aplikasi ini adalah fitur dan kestabilannya. Pustaka yang digunakan juga tidak terlalu banyak dan waktu tunggunya juga tidak terlalu lama. Kekurangan dari aplikasi ini adalah antarmuka penggunanya (user interface) yang sedikit kaku apabila dibandingkan dengan aplikasi sejenis di sistem operasi lain[18].