BAB II
LANDASAN TEORI

II.1 *Game*

Game merupakan salah satu media hiburan yang paling populer untuk semua kalangan usia. Sejak pertama kali ditemukan sampai saat sekarang, teknologi *game* telah mengalami kemajuan yang terbilang pesat. Hal ini ditandai dengan berkembangnya jenis *game*, produk, alat dan jenis interaksi *game* dengan penggunaan yang semakin beragam bentuknya.

II.1.1 Pengertian *Game*

Dalam kamus bahasa Indonesia *game* adalah permainan. Sebuah permainan adalah sebuah sistem dimana pemain terlibat dalam konflik buatan, disini pemain berinteraksi dengan sistem dan konflik dalam permainan merupakan rekayasa atau buatan, dalam permainan terdapat peraturan yang bertujuan untuk membatasi perilaku pemain dan menentukan permainan. Permainan merupakan sebuah aktivitas rekreasi dengan tujuan bersenang-senang, mengisi waktu luang, atau berolahraga santai. Permainan biasanya dilakukan sendiri atau bersama-sama [5].

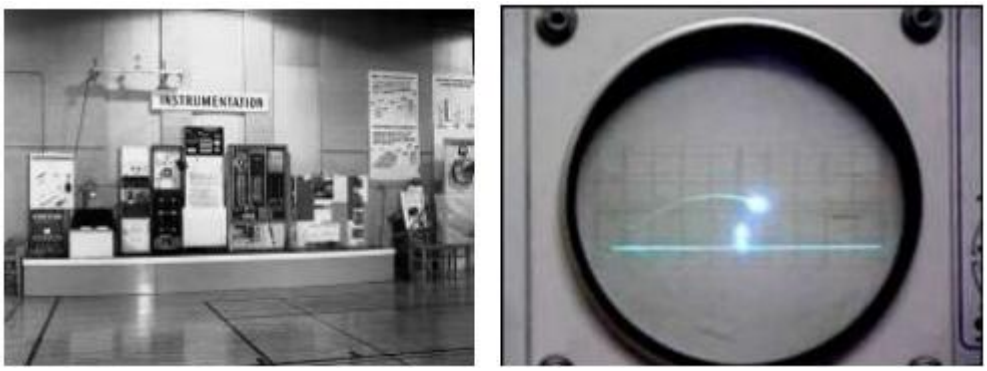
Permainan video (video game) adalah permainan yang menggunakan interaksi antarmuka pengguna melalui gambar yang dihasilkan oleh piranti video. Permainan video umumnya menyediakan system penghargaan misalnya skor yang dihitung berdasarkan tingkat keberhasilan yang dicapai dalam menyelesaikan tugas-tugas yang ada di dalam permainan. Sistem elektronik yang digunakan untuk menjalankan permainan video disebut platform, contohnya adalah komputer pribadi dan konsol permainan [5].

II.1.2 Sejarah Singkat *Game*

Perjalanan terciptanya *video game* memiliki kisah yang panjang, perkembangannya sangat dipengaruhi oleh kemajuan teknologi yang terjadi. *Video games* tidak akan terbentuk jika tidak ada penemuan dari perangkat lunak komputer. Pada masa perang komputer dibutuhkan untuk melakukan proses kalkulasi yang rumit dan melakukannya dengan cepat. Komputer pertama yang muncul sebagai satu kesatuan yang lengkap dan melakukan fungsi tersebut adalah

komputer *Colossus*, yang dirangkai oleh Alan M. Turing dari Inggris pada tahun 1943. Dari konsep teknologi komputer muncullah cikal bakal *game* hadir kedunia [6].

Pada tahun 1952 seorang mahasiswa Universitas Cambridge bernama A.S. Gouglas membuat permainan OXO (*Tic Tac Toe*) dalam versi grafik. Permainan ini ia kembangkan ketika hendak mendemonstrasikan tesisnya tentang interaksi antara manusia dan komputer. Pada tahun 1958 William Higin Botham mendesain sebuah *game* dengan judul *Tennis For Two*. *Tennis for Two* merupakan sebuah *game* analog komputer dikendalikan dengan sebuah knob dan sebuah tombol. Grafisnya ditampilkan secara sederhana dalam sebuah alat primitif yang disebut sebagai *oscilloscope*, sebuah alat yang biasanya digunakan sebagai indikasi sinyal elektrik dapat di lihat di [6]:



Gambar II-1 *Osciloscop* [6]

Pada pertengahan 1960 Joseph Weizenbaum seorang profesor ilmu komputer dari MIT menciptakan sebuah Virtual Psychiatrist yang dinamakan ELIZA. Konsep awalnya adalah artificial intelligence yang merupakan pengembangan dari karya Alan Turing. Eliza dapat menjawab pertanyaan-pertanyaan yang dilontarkan kepadanya seperti seorang psikiater sesungguhnya. ELIZA diciptakan untuk dapat menjawab berbagai pertanyaan yang dilontarkan, secara nyata ELIZA dapat merespon dengan baik berbagai pertanyaan tersebut, sehingga ELIZA dapat dirasakan real oleh para penggunanya. Oleh karena itu, ELIZA menjadi *the First Virtual Character of the Information Age* pada masanya. ELIZA masuk dalam bagian sejarah game dikarenakan sifatnya yang menghibur

dan interaktif terhadap pemakainya, walaupun secara teknik belum ada *gameplay* yang jelas dalam permainannya atau penggunaannya [6]:

Di akhir 1960 seorang mahasiswa MIT bernama Steve Russell menciptakan sebuah *game* yang menjadi inspirasi *game arcade* pertama di dunia, *game* tersebut dinamakan *Spacewar*. *Spacewar* dapat disebut sebagai awal dari *game* komersial di dunia dalam platform komputer *game*. Hingga saat ini perkembangan *game* komputer berkembang sangat pesat seiring dengan perkembangan teknologi komputer yang mendukung tampilan grafis 3D dan *gameplay* yang lebih realistis [6].

II.1.3 Klasifikasi Game

Klasifikasi *Game* biasanya digunakan untuk mengkatagorikan sebuah *game* berdasarkan dari interaksi *gameplay* daripada perbedaan tampilan atau narasi. Sebuah jenis *game* dapat didefinisikan dengan kumpulan dari sebuah tipe permainan dari *game* tersebut. *Game* dapat diklasifikasikan menjadi dua, yaitu:

II.1.3.1 Berdasarkan Jenis *Platform* yang Digunakan

Berikut ini beberapa jenis *game* berdasarkan jenis *platform* yang digunakan [7]:

1. *Arcade games*, yaitu yang sering disebut ding-dong di Indonesia, biasanya berada di daerah/tempat khusus dan memiliki *box* atau mesin yang memang khusus di desain untuk jenis *video games* tertentu dan tidak jarang bahkan memiliki fitur yang dapat membuat pemainnya lebih merasa masuk dan menikmati, seperti pistol, kursi khusus, sensor gerakan, sensor injakkan dan stir mobil (beserta transmisinya tentunya).
2. PC *Games*, yaitu *video game* yang dimainkan menggunakan *Personal Computers*.
3. *Console games*, yaitu *video games* yang dimainkan menggunakan *console* tertentu, seperti *Playstation2, Playstation 3*, XBOX 360, dan Nintendo Wii.
4. *Handheld games*, yaitu *game* yang dimainkan di *console* khusus *video game* yang dapat dibawa kemana-mana, contoh Nintendo DS dan Sony PSP.
5. *Mobile games*, yaitu *game* yang dapat dimainkan atau khusus untuk *mobile phone*.

II.1.3.2 Berdasarkan *Genre* Permainannya

Jika dilihat berdasarkan *genre* permainannya maka *game* terbagi dalam beberapa jenis, diantaranya adalah [7]:

1. *Action Game*
   a. *Action Adventure Game*

      Genre game yang berfokus pada eksplorasi dan biasanya mempunyai unsur item *gathering*, penyelesaiaan puzzle simpel dan pertarungan. Contoh game: *The Legend Of Zelda series* dan *Metroid series.*

   b. *Stealth Game*

      Termasuk dalam *genre* terbaru, biasanya digolongkan dalam mata-mata yang bisa melakukan aksinya secara rahasia. Contoh *game*: *Metal Gear*.

   c. *Survival Horror Game*

      Genre game yang berusaha membuat pemain menjadi tegang dan takut dengan elemen-elemen horror. Contoh *game*: *Resident Evil series*.

   d. *Beat'em Up Game*

      Genre game combat dimana satu orang melawan banyak musuh yang telah disediakan. Contoh g*ame*: *Dynasty Warrior series* dan *Final Fight*.

   e. *Fighting Game*

      Game pertarungan dua pemain dengan jurus-jurus yang biasa dikeluarkan dengan menekan beberapa tombol pada keyboard dengan urutan tertentu. Contoh *game*: *Street Fighter dan Tekken series*.

   f. *Maze Game*

      Genre game yang membutuhkan kecepatan berpikir dan bereaksi serta berunsur ketepatan menavigasi. Contoh *game*: Pac-Man dan Bomberman.

   g. *Platform Game*

      Genre Game dengan *gameplay* berlari, melompat, mengayun dan sebagainya. Contoh *game*: *Donkey Kong* dan *Ray Man*.

2. *Shooter*
   a. *First Person Shooter Game*

      Genre game yang mengutamakan *shooting* dan *combat* dari perspektif langsung mata karakter yang bertujuan untuk memberikan pemain

perasaan berada ditempat itu dan bisa fokus menembak. Contoh *game*: *Army Men*.

b. *Massively Multiplayer Online First Person Shooter Game*

Genre game yang mengkombinasikan *gameplay first person shooter* dengan dunia *virtual* dimana banyak player juga ikut bermain melaui internet. Contoh *game*: *Counter Strike Online*.

c. *Third Person Shooter Game*

Genre game yang sama seperti *first person shooter game* yaitu mengutamakan *shooting* dan *combat* dari perspektif karakter yang bertujuan untuk memberikan pemandangan yang lebih luas dan gerakan yang lebih banyak. Contoh *game*: *Max Payne*.

d. *Tactical Shooter Game*

Genre yang mengutamakan perencanaan dan kerja sama tim untuk memenangkan *game*. Contoh *game*: *TomClancy's Ghost Recon series*.

e. *Light Gun* Game

Genre dengan lebih banyak pada *arcade* dengan peralatan tertentu seperti senjata mainan yang mempunyai sensor khusus terhadap layar. Contoh *game*: *Time Crisis* dan *Duck Hunt*.

f. *Shoot'em Up Game*

Genre dengan ciri khas gambar 2D dan *scrolling playing area*. Contoh *game*: *Star Fox series*.

3. *Adventure Game*

Adventuregame dikategorikan sebagai *gameplay* yang mengharuskan pemain memecahkan bermacam-macam teka-teki melaui interaksi dengan orang lingkungan dalam *game* tersebut.

a. *Text Adventure*

Pemain akan menggunakan keyboard untuk mengetikkan berupa perintah dan komputer akan menganalisa perintah tersebut lalu menjalankan karakter sesuai perintah tersebut.

b. *Graphical Adventure Game*

Genre yang merupakan perkembangan dari *text adventure*.Pemain dapat menggunakan mouse untuk menggerakkan karakter.

c. *Visual Novel Game*

Genre yang memberikan keleluasaan untuk memilih jalan ceritanya.

d. *Interactive Movie Game*

Genre game dengan rangkaian live action dari karakter yang dimainkan pemain. Contoh *game*: *Space Ace*.

e. *Dialog Game*Pada *genre* ini, pemain akan mengalami kemajuan tergantung pada apa yang mereka katakan. Contoh *game*:*Law And Order,The Vangeful Heart*.

4. *Role Playing Game*

Role playing game adalah *game* yang memiliki *gameplay* dimana karakter milik *player* akan berpetualang dengan *skill combat* dalam cerita *game*.

a. *Action Role Playing Game*

Genre game yang memasukkan unsur *action game* dan *action adventure game*. Contoh *game*: Diablo 1 & 2.

b. *Massively Multyplayer Online Role Playing Game*

Konsep dari *genre* ini terkombinasi dengan *genre-genre* lainnya yang berupa fantasi.Contoh *game*: *Rising Force Online*.

c. *Tactical Role Playing Game*

Dalam *genre* ini, pemain akan diberikan giliran masing-masing untuk menentukan langkah-langkah yang akan dilakukan oleh karakter. Contoh *game*: *Final Fantasy Tactics*.

5. *Simulation Game*

Genre ini bertujuan untuk memberikan pengalaman simulasi kepada pemain.

a. *Construction and Management Simulation Game*

Genre ini merupakan bagian dari *economic simulation game*. Contoh *game*: *Sims City series*.

b. *Economic Simulation Game*

Genre ini berupa simulasi keadaan ekonomi dimana pemain mengontrol keadaan ekonomi dari *game* tersebut. Contoh *game*: *Monopoly Tycoon*.

c. *God Game*

Dalam *genre* ini tidak ada tujuan akhir yang membuat pemain memenangkan *game*. Contoh *game*: *The Sims series*.

d. *Government Simulation Game*

Genre game yang memasukkan unsur kepolisian, pemerintahan atau politik sebuah negara.

6. *Strategy Game*

Genre strategy game berfokus pada *gameplay* dimana dibutuhkan pemikiran yang tepat agar dapat meraih kemenangan.

a. *Real Time Strategy*

Dalam *real time strategi*, *action* dilakukan dalam waktu yang bersamaan oleh masing-masing pihak dimana *action* dimainkan per-ronde atau bergiliran. Contoh *game*: *Warcraft series*.

b. *Tactical Game*

Dalam *genre* ini pemain harus menggunakan bermacam-macam taktik dan strategi untuk mencapai kemenangan. Contoh *game*: *Dark Omen*.

c. 4X Game

Genre ini berarti penjelajahan, menjajah dan memusnahkan. Contoh *game*: *Galactic Civizations*.

d. Artillery *Game*

Genre game ini biasanya mengikutkan perang dengan tank atau tentara militer. Contoh *game*: *Tanarus*.

7. *Vehicle Simulation Game*

Genre ini merupakan simulasi yang memberikan pemain sebuah pengalaman realistik dalam mengendarai kendaraan tertentu.

a. *Flight Game*

Dalam *genre* ini, pemain tidak hanya bersimulasi mengontrol pesawat terbang tetapi juga bisa bertarungt di udara. Contoh *game*: Falcon 4.0.

b. *Racing Game*

Genre yang menempatkan pemain sebagai *driver* dengan kendaraan seperti mobil. Contoh *game*: *Need For Speed series*.

c. *Space Game*

Genre ini bersifat pertarungan di angkasa luar. Contoh *game*: *Star Wars*.

d. *Train Game*

*Genre*ini mensimulasikan yang berhubungan dengan trasnportasi kereta. Contoh *game*: *Rail Simulator*.

8. *Sports Game*

Genre ini merupakan bentuk *game* simulasi yang memberikan pemain sebuah pengalaman dalam bermain olahraga tertentu di PC atau *game* console. Contoh *game* dari *genre* ini adalah seri *Winning Eleven*, seri *ProEvolution Soccer*, seri NBA, seri FIFA, John Madden NFL, Lakers vs Celtics, Tony Hawk Pro Skater, dll.

II.1.4 Unsur *Game*

Dalam sebuah *game* terdapat unsur-unsur yang melengkapinya. Berikut beberapa unsur utama dari sebuah *game*, yaitu [8]:

1. Representasi

Representasi yang dimaksud dari sebuah *game* adalah bersifat *subjektif* bukan bersifat *objektif*. Sehingga bila seseorang memainkan *game* tembakan, maka makna yang terkandung dari menghancurkan musuh adalah bergantung pada bagaimana persepsi pemain terhadap *game* yang sedang dimainkannya. Representasi pada *game* lebih bersifat fantasi dari pada ilmiah. Selain itu representasi tujuan (*objective*) dari sebuah *game* bersifat tidak nyata (*unreal*) sehingga sama sekali tidak dikehendaki pemain secara fisik mewujudkan secara langsung situasi yang didapatnya dari *game*.

2. Interaksi

Realitas dapat di representasikan secara statis atau dinamis. Sebuah *game* dikehendaki untuk memiliki sifat interaktif dalam upaya untuk merepresentasikan realitas. Interaktif dan keterlibatan pemain untuk

menentukan sendiri representasi realitas merupakan komponen utama dari suatu *game*. Interaksi diperlukan untuk menjadi penghubung keterlibatan individu pemain kedalam situasi yang diciptakan oleh *game*. Sehingga tantangan yang harus diatasi suatu *game* bukan secara teknis tetapi harus diselesaikan sesuai dengan reaksi dari pemain. Dalam hal ini interaksi akan merubah tantangan yang sifatnya pasif menjadi aktif. Pemain harus menciptakan sendiri solusi untuk menyelesaikan tantangan berdasar intuisi yang dimilikinya.

3. Konflik

   Konflik muncul sebagai akibat dari interaksi.Pemain dihadapkan pada satu tujuan yang harus dicapai. Namun sejumlah rintangan disediakan untuk menghalangi pemain untuk mencapai tujuan dengan cara yang mudah. Bila rintangan tersebut bersifat statis maka termasuk dalam kategori *puzzle*. Bila rintangan bersifat aktif atau dinamis dan reaktif terhadap respon yang diberikan oleh pemain maka inilah yang dimaksud dengan *game*.

4. Keamanan

   Adanya konflik dimungkinkan untuk menimbulkan resiko yang membahayakan. *Game* dibuat hanya untuk sekedar memberikan pengalaman psikologis dari suatu konflik sehingga *game* harus sepenuhnya bersifat aman dan tidak membahayakan langsung secara fisik kepada pemainnya. Sehingga walaupun berjam-jam pemain memainkan *game* peperangan, maka secara psikologis dia akan mendapatkan pengalaman kalah dan menang namun begitu keluar dari dunia yang direpresentasikan oleh *game* tidak ada satupun efek langsung secara fisik kepada pemain tersebut.

II.2 *Game* Edukasi

Penerapan *game* edukasi bermula dari perkembangan *video game* yang sangat pesat dan menjadikan *game* edukasi sebagai media efektif yang interaktif dan banyak dikembangkan di perindustrian. Menurut Randel *game* sangat berpotensi untuk menumbuhkan kembali motivasi belajar anak yang mengalami penurunan. Berdasarkan penelitian yang dilakukan oleh Randel pada tahun 1991 tercatat bahwa pemakaian *game* sangat bermanfaat pada materi-materi yang

berhubungan dengan matematika, fisika dan kemampuan berbahasa (seperti studi sosial, biologi dan logika) [9]:

II.2.1 Pengertian *Game* Edukasi

Game yang memiliki kontent pendidikan lebih dikenal dengan istilah *game* edukasi. *Game* berjenis edukasi ini bertujuan untuk memancing minat belajar anak terhadap materi pelajaran sambil bermain, sehingga dengan perasaan senang diharapkan anak bisa lebih mudah memahami materi pelajaran yang disajikan. Jenis ini sebenarnya lebih mengacu kepada isi dan tujuan game, bukan jenis yang sesungguhnya [9].

II.2.2 Kriteria *Game* Edukasi

Menurut Foremen beberapa kriteria yang harus diterapkan dalam aplikasi sebuah *game* edukasi adalah [9]:

1. Individualization

   Materi pembelajaran (pengetahuan) dibuat sesuai dengan kebutuhan individual dari pembelajar, sedangkan game mengadopsi level individual dari pemain
2. Feedback Active

   Adanya feedback yang sesuai dengan cepat untuk memperbaiki pembelajaran dan mengurangi ketidaktahuan pembelajar terhadap materi yang disampaikan, sedangkan game menyediakan feedback dengan cepat dan konstektual.
3. Active learning

   Adanya kecenderungan untuk menyertakan pelajar secara aktif dalam menciptakan penemuan dan pengetahuan baru yang membangun, sedangkan game menyediakan suatu lingkungan yang membantu terjadinya penemuan baru tersebut.
4. Motivation

   Pelajar termotivasi dengan reward yang diberikan dalam aktivitas permainan, sedangkan *game* melibatkan pengguna berjam-jam untuk mencapai tujuan.
5. Social

   Pengetahuan merupakan suatu proses partisipasi sosial, sedangkan game dapat dimainkan dengan orang lain (seperti *game multiplayer*) atau melibatkan komunitas dari pecinta *game* yang sama.

6. *Scaffolding*

   Pelajar secara berangsur-angsur ditantang dengan tingkat kesulitan yang makin tinggi dan dapat melangkah lebih maju untuk mencapai kemenangan dari permainan, sedangkan game dibangun secara multi level, pemain tidak bisa bergerak ke level yang lebih tinggi sampai dia mampu menyelesaikan permainan di level yang ada.

7. *Transfer*

   Pelajar mengembangkan kemampuan untuk mentransfer pengetahuan dari satu orang ke orang yang lain, sedangkan *game*, sedangkan *game* mengijinkan pemain untuk menstransfer informasi dari suatu konteks ke konteks yang lain

8. Assessment

   Setiap individu mempunyai kesempatan untuk menilai pelajaran mereka sendiri atau membandingkannya dengan orang lain.

II.3 Multimedia

II.3.1 Sejarah Multimedia

Istilah multimedia pertama kali di kenal pada dunia teater, yang mempertunjukan pagelaran dengan menggunakan gerak, musik, dan video untuk menambah dramatisasi suatu cerita. Sekarang multimedia dikenal dengan perpanduan dari hasil gambar atau image, grafik, teks, suara, TV, dan animasi sehingga menjadi suatu karya yang dapat dinikmati secara audio visual [10].

Pada umumnya orang mengenal multimedia sebagai sistem dari komputer personal (PC) yang berkembang pesat dewasa ini. Dalam perkembangannya saat ini multimedia digunakan sebagai media untuk pengajaran, pelatihan, pembuatan manufaktur, dan digunakan untuk kegiatan promosi penjualan [10].

Sistem multimedia dimulai pada akhir 1980-an, sejak permulaan tersebut hampir setiap pemasok perangkat keras dan lunak melompat ke multimedia.Pada tahun1994 diperkirakan ada lebih dari 700 produk dan sistem multimedia dipasaran [11].

II.3.2 Pengertian Multimedia

Multimedia dapat diartikan sebagai penggunaan beberapa media yang berbeda untuk menggabungkan dan menyampaikan informasi dalam bentuk teks,

audio, grafik, animasi, dan video.Adapun definisi multimedia menurut beberapa ahli, yaitu [11]:

1. Kombinasi dari komputer dan video. (Rosch, 1996)
2. Kombinasi dari tiga elemen, yaitu suara, gambar,dan teks. (McComick,1996)
3. Kombinasi dari paling sedikit dua media input atau output. Media ini dapat berupa audio (suara, musik), animasi, video, teks, grafik dan gambar. (Turban dkk, 2002)
4. Alat yang dapat menciptakan presentasi yang dinamis dan interaktif dari kombinasi teks, grafik, animasi, audio dan video. (Robin dan Linda, 2001)
5. Multimedia adalah pemanfaatan komputer untuk membuat danmenggabungkan teks, grafik, audio, video, dengan menggunakan tool yang memungkinkan pemakai berinteraksi, berkreasi, dan berkomunikasi.dalam konteks komputer. (Hofstetter, 2001)

II.3.3 Jenis Multimedia

Multimedia terbagi menjadi dua jenis, yaitu [11] :

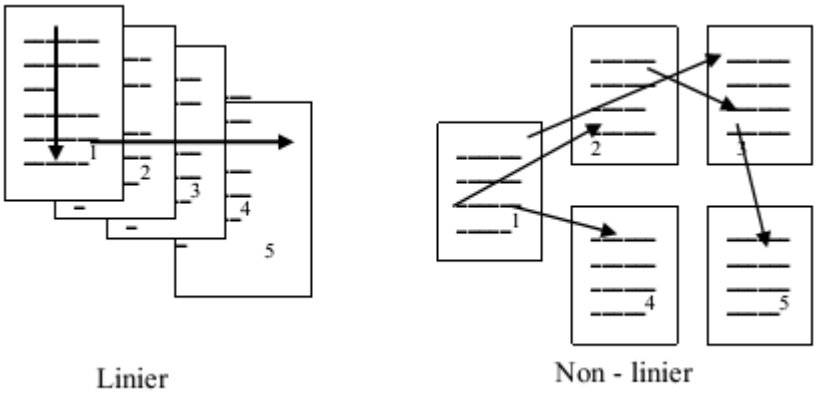
1. Multimedia Linier

   Multimedia linier adalah suatu multimedia yang tidak dilengkapi oleh alatpengontrol apapun yang dapat dioperasikan oleh pengguna. Multimedia iniberjalan sekuensial (berurutan), contohnya: TV dan film.

2. Multimedia Interaktif

   Multimedia interaktif dilengkapi oleh alat pengontrol yang dapat dioperasikanoleh pengguna, sehingga pengguna dapat memilih apa yang dikehendakiuntuk proses selanjutnya. Contohnya: video game, pembelajaran interaktif,dan lain-lain.

Untuk lebih jelasnya perbedaan antara multimedia linear dan non-linear dapat di lihat pada [11]:



Gambar II-2 Jenis Multimedia [11]

II.3.4 Tujuan Multimedia

Menurut Sutopo, multimedia dapat digunakan untuk bermacam-macam bidang pekerjaan, tergantung dari kreativitas untuk mengembangkannya setelah mengetahui defenisi dari multimedia serta elemen-elemen multimedia yang ada, serta aplikasi-aplikasi yang saat ini digunakan pada bidang kehidupan manusia, maka dapat diketahui bahwa tujuan dari penggunaan multimedia adalah sebagai berikut [11] :

1. Multimedia dalam penggunaannya dapat meningkatkan efektivitas dari penyampaian suatu informasi.
2. Penggunaan multimedia dalam lingkungan dapat mendorong.
3. Partisipasi, keterlibatan serta eksplorasi pengguna tersebut.
4. Aplikasi multimedia dapat meransang panca indera, karena dengan penggunaannya multimedia akan meransang beberapa indera penting manusia, seperti : penglihatan, pendengaran, aksi maupun suara.

II.3.5 Hubungan Konsep Multimedia dengan Aplikasi *Game* Edukasi

Konsep multimedia dengan aplikasi *game* edukasi memiliki hubungan yang cukup erat. Dalam pendidikan, multimedia digunakan untuk memproduksi pelatihan berbasis computer dan buku referensi seperti ensiklopedia yang memungkinkan pengguna melalui serangkaian modul presentasi, teks tentang topik tertentu, dan gambar yang terkait dalam berbagai format informasi. *Informal Edutainment* adalah istilah yang digunakan untuk menjelaskan menggabungkan

pendidikan dengan hiburan, terutama hiburan multimedia. Pembelajaran teori dalam dekade terakhir telah berkembang dramatis karena pengenalan multimedia [11]. Oleh karena itu,aplikasi *game* edukasi tidak pernah terlepas dari konsep multimedia.

II.4 Matematika

II.4.1 Soal Cerita Dalam Pembelajaran Matematika

Bentuk soal yang digunakan untuk mengukur kemampuan siswa dalam pembelajaran matematika dapat berupa soal cerita atau soal non cerita. Soal cerita yang dimaksud erat kaitannyadengan masalah yang ada dalam kehidupan siswa sehari-hari, sehingga yang dimaksud dengan soal cerita matematika adalah soal matematika yang terkait dengan kehidupan sehari-hari untuk dicari penyelesaiannya menggunakan kalimat matematika yang memuat bilangan, operasi hitung (+, –, ×, :), dan relasi (=, <, >, ≤,≥). Soal cerita semacam ini penting untuk diberikan kepada siswa guna melatih perkembangan proses berpikir mereka secara berkelanjutan dalam rangka mencapai standar kompetensi yang telah ditetapkan, sehingga keberadaannya sangat diperlukan [12].

II.4.2 Tujuan Pembelajaran Soal Cerita di SD

Tujuan pembelajaran soal cerita di Sekolah Dasar sebagai berikut [12]:

1. Melatih siswa berpikir deduktif.
2. Membiasakan siswa untuk melihat hubungan antara kehidupan sehari-hari dengan pengetahuan matematika yang telah mereka peroleh di sekolah.
3. Memperkuat pemahaman siswa terhadap konsep matematika tertentu, maksudnya dalam menyelesaikan soal cerita siswa perlu mengingat kembali konsep-konsep matematika yang telah dipelajarinya sehingga pemahaman terhadap konsep-konsep tersebut semakin kuat.

II.4.3 Macam-macam soal cerita dalam matematika

Macam-macam soal cerita dalam matematika dilihat dari segi macam operasi hitung yang terkandung dalam soal cerita dibedakan sebagai [12]:

1. Soal cerita satu langkah (one-step word problems) adalah soal cerita yang di dalamnya mengandung kalimat matematika dengan satu jenis operasi hitung (penjumlahan atau pengurangan atau perkalian atau pembagian).

2. Soal cerita dua langkah (*two-step word problems*), adalah soal cerita yang didalamnya mengandung kalimat matematika dengan dua jenis operasi hitung.
3. Soal cerita lebih dari dua langkah (*multi-step word problems*), adalah soal cerita yang didalamnya mengandung kalimat matematika dengan lebih dari dua jenis operasi hitung.

II.4.4 Strategi Pemecahan Masalah dan Penyelesaian Soal Cerita

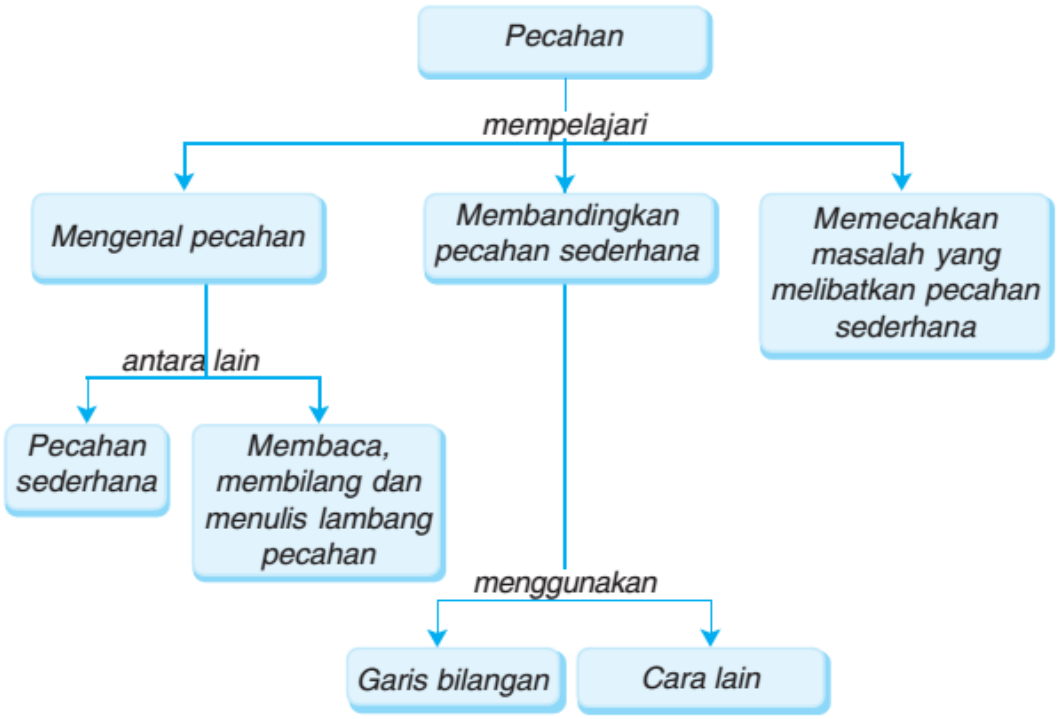
Dalam pembelajaran matematika ada soal pemecahan masalah dan ada soal bukan pemecahan masalah. Soal cerita matematika umumnya erat kaitannya dengan kehidupan sehari-hari. Soal tersebut penting sekali diberikan kepada siswa Sekolah Dasar, karena pada umumnya soal cerita tersebut dapat digunakan untuk melatih siswa dalam menyelesaikan masalah. Oleh karena itu dalam menyelesaikan soal cerita dapat digunakan strategi penyelesaian masalah, walaupun soal ceritamatematika belum tentu merupakan soal pemecahan masalah. Kemampuan yang diperlukan untuk menyelesaikan soal cerita tidak hanya kemampuan keterampilan (*skill*) dan mungkin algoritma tertentu saja melainkan kemampuan lainnya yaitu kemampuan menyusun rencana dan strategi yang akan digunakan dalam mencapai penyelesaian [12].

Eicholz mengemukakan bahwa langkah-langkah yang diperlukan dalam menyelesaikansoal cerita sebagai berikut [12]:

1. Memahami apa yang ditanyakan
2. Menemukan data yang dibutuhkan
3. Merencanakan apa yang harus dilakukan
4. Menemukan jawaban melalui komputasi (penghitungan)
5. Mengoreksi kembali jawaban.

II.4.5 Pecahan Sederhana

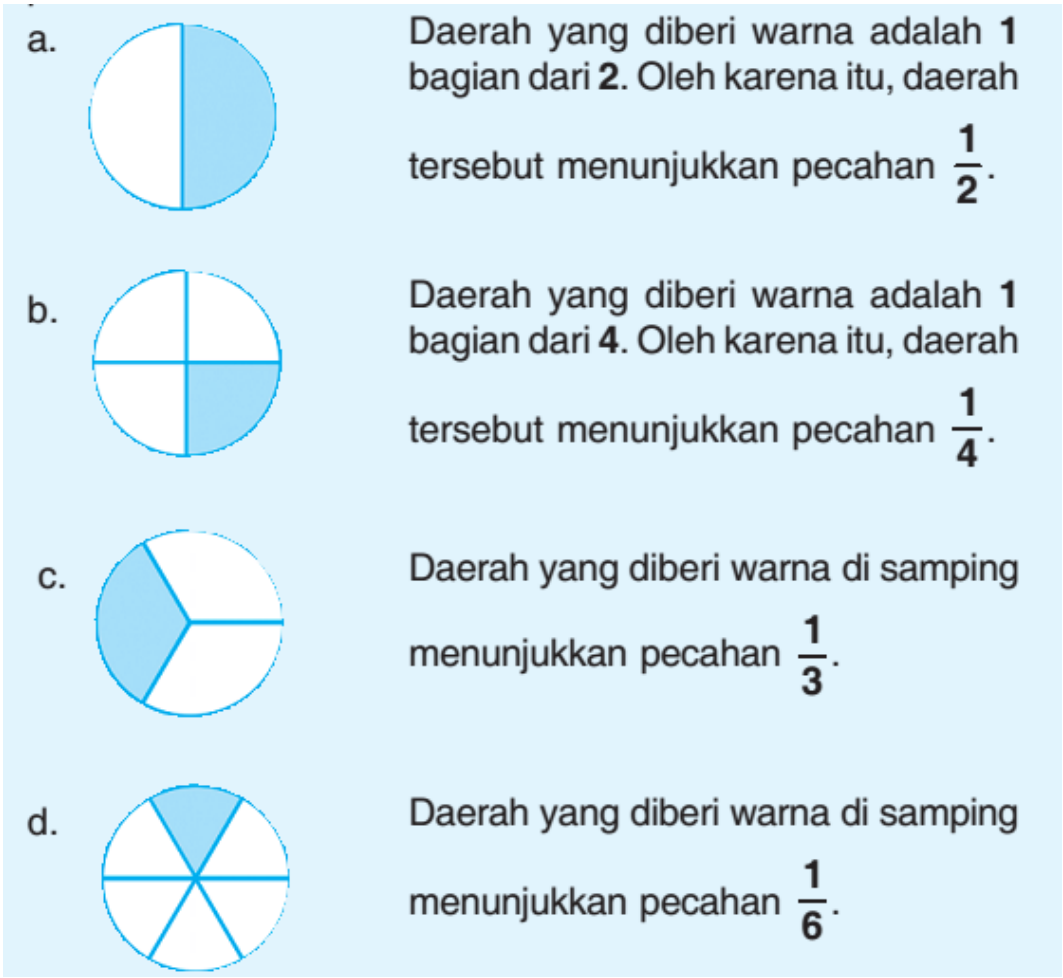
Pada penelitian ini materi yang di ambil untuk pembangunan aplikasi di ambil dari buku cerdas berhitung matematika 3 yaitu tentang pecahan. Adapun peta konsep dari pembelajaran pecahan sederhana pada buku tersebut adalah sebagai berikut [13]:



Gambar II-3 Peta Konsep Pembelajaran Pecahan Sederhana

II.4.5.1 Mengenal Pecahan

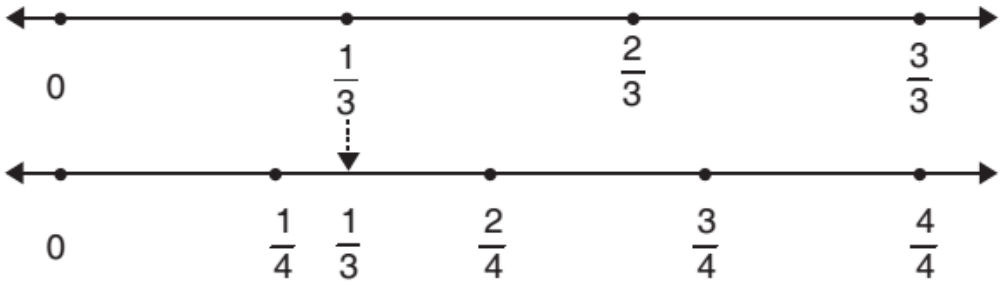
Pada materi ini siswa diperkenalkan pada pembelajaran pecahan dalam bentuk gambar yang dibagi dalam bentuk sama besar, adapun contoh soalnya sebagai berikut [13]:



Gambar II-4 Pengenalan Pecahan Sederhana

II.4.5.2 Membandingkan Pecahan Sederhana

Untuk membandingkan dua pecahan dapat kamu tentukan nilainya, apakah lebih besar daripada (>), lebih kecil daripada (<), ataukah sama dengan (=). Adapun cara untuk menyelesaikan nya dapat dilihat pada contoh soal berikut [13]:



Gambar II-5 Garis Bilangan Pecahan

Setelah memerhatikan gambar di atas, maka kita dapat menentukannilai suatu bilangan pecahan.Pecahan 1/ 3 terletak di sebelah kanan 1/4, maka 1/3 lebih besar daripada 1/4, dapat di tulis 1/3 > 1/4.Begitu pula sebaliknya pecahan 1/3 terdapat di sebelah kiri pecahan 2/4, maka 1/3 lebih kecil daripada 2/4, dan dapat ditulis 1/3 < 2/4.

II.4.5.3 Memecahkan masalah yang melibatkan masalah pecahan sederhana

Pada materi ini siswa mulai diperkenalkan pada materi pecahan sederhana dalam bentuk soal cerita. Adapun contoh soalnya adalah sebagai berikut [13]:

a. Ibu membeli kue. Bagian yang 1/4 dimakan Kak Jo dan yang 2/4 dimakan Ayah. Siapakah yang makan bagian kue lebih banyak?

   Jawab :

   Bagian yang dimakan kak Jo adalah 1/4 , bagian yang dimakan Ayah adalah 2/4, Jadi ayah makan bagian kue lebih banyak atau lebih besar dari pada kak Jo. Dapat di tulis 1/4 < 2/4 atau 2/4 > 1/4.

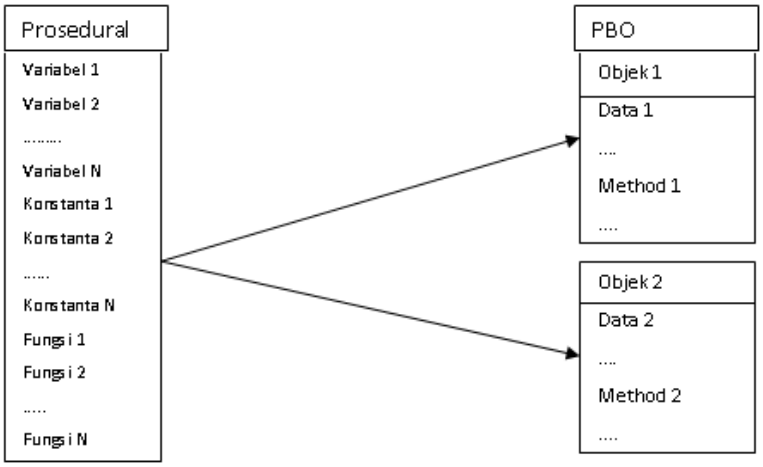
II.5 Pemrograman Berorientasi Objek (*Object Oriented Programing*)

Pemrograman berorientasi objek merupakan teknik dalam menyelesaikan masalah yang kerap muncul dalam pengembangan perangkat lunak. Teknik ini merupakan titik kulminasi dalam menemukan cara yang efektif dalam membangun sistem dan menajsi metode yang paling banyak dipakai oleh para pengembang perangakat lunak saat ini. Orientasi objek merupakan teknik pemodelan sistem riil yang berbasis objek.Inti dari konsep ini adalah objek yang merupakan model dari sistem nyata [14].

II.5.1 Perbedaan Pemrograman Berorieantasi Objek dengan Pemrograman Prosedural

Pada model pemrograman tradisional atau procedural (*Process-Oriented Model*), semua data dan kode digabung menjadi satu bagian dalam satu program. Untuk program-program sederhana yang membutuhkan hanya beberapa buah baris kode, pengguanaan model ini tentu tidak begitu menjadi masalah. Permasalah baru akan timbul pada saat program tersebut berkembang kearah yang lebih besar dan kompleks, yang mungkin akan membutuhkan ratusan atau bahkan ribuan kode. Alasan inilah yang menjadi benih diciptakannya model

pemrograman baru, yaitu model pemrograman berorientasi objek (*Object-Oriented Mode*l). Pada model ini, komponen-komponen penyusun program secara konseptual akan dipecah menjadi bagian-bagian tersendiri yang disebut dengan objek. Berikut ini gambar yang mengilustrasikan perbedaan yang terdapat antara pemrograman prosedural dan pemrograman berorientasi objek [15].



Gambar II-6 pemrograman prosedural dan berorientasi objek [15]

Seperti yang dapat anda lihat pada gambar di atas, dalam pemrograman berorientasi objek, setiap objek akan memiliki data (sifat, berupa variabel maupun konstanta) dan *method* (perillaku atau kemampuan melakukan sesuatu, berupa fungsi). Jadi, objek dapat di definisikan sebagai sesuatu entitas yang memiliki data dan *method*. Sebagai contoh, manusia adalah objek yang memiliki data-data (nama, jenis kelamin, tinggi badan, berat badan, dan sebagainya) dan juga *method*(misalnya : cara bicara, cara berjalan, cara marah, dan sebagainya) [15].

II.5.2 Karakteristik Pemrograman Berorientasi Objek

Adapun karakteristik yang dimiliki pemrograman berorientasi objek yaitu [14] :

a. *Objects*

   Objek adalah entitas yang memiliki atribut, karakter (*behaviour*) dan kadang kala disertai kondisi (state).

b. *Class*

Kelas adalah penggambaran satu objek yang memiliki atribut dan behavior yang sama. Kelas mirip dengan tipe data pada pemrograman non-objek, tapi lebih komperehensif karena terdapat struktur sekaligus karakteristiknya.Kita dapat membentuk kelas baru yang spesifik dari kelas *general*-nya.Kelas dan objek merupakan jantung dari pemrograman berorientasi objek.

c. *Abstraction*

Abstraksi adalah pengabstrakan atau penyembunyian kerumitan dari suatu proses. Sebagai contoh, orang tidak perlu berpikir bahwa mobil adalah kumpulan dari puluhan atau ribuan bagian-bagian kecil yaitu mesin, kabel, baut, roda, dan sebagainya. Orang hanya perlu berpikir bahwa mobil adalah sebuah objek yang telah memiliki perilaku spesifik, yang dapat digunakan sebagai alat treansportasi, sehingga orang hanya perlu menggunakannya tanpa harus mengetahui kerumitan proses yang terdapat pada mobil tersebut.

d. *Encapsulation*

Ciri penting lainnya dari OOP adalah *encapsulation*. *Encapsulation* adalah sebuah proses dimana tidak ada akses langsung ke data yang diberikan,bahkan tersembunyi. Jika seseorang ingin mendapat data, maka ia harus berinteraksi dengan objek perantara yang bertanggung jawab atas data tersebut.

e. *Polymorphism*

Polymorphism atau polimorfisme adalah kemampuan dua buah objek berbeda untuk merespon pesan permintaan yang sama dalam suatu cara yang unik. Contohnya, fungsi cetak akan menghasilkan keluaran berbeda ketika dijalankan pada mesin pencetak dan layar monitor. Dalam OOP, penerapan tipe *polymorphism* melalui proses yang disebut *overloading*. Hal tersebut mengizinkan pengimplementasian metode yang berbeda pada sebuah objek yang mempunyai nama sama.

f. *Inheritance*

Penggunaan *inheritance* dalam OOP untuk mengklasifikasikan objek dalam program sesuai karakteristik umum dan fungsinya. Hal ini sama dengan konsep pewarisan sifat pada manusia. Jika manusia memiliki keturunan, maka beberapa sifat dari orang tuanya terkadang menurun pada anaknya. Analogi

tersebut kemudian berlaku pada OOP dalam *inheritance* ini. Konsep ini kemudian menjadikan pemrograman lebih mudah karena memungkinkan pemrogram untuk mengkombinasikan karakteristik umum kedalam objek orang tua (*superclass*) dan mewariskan karakteristik ini ke objek anaknya (*subclass*).

II.6 *Unified Modeling Language* (UML)

UML merupakan bahasa standar untuk penulisan *blueprint software* yang digunakan untuk visualisasi, spesifikasi, pembentukan dan pendokumentasian alat-alat dari sistem perangkat lunak. UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat. Bahasa pemodelan merupakan bagian terpenting dari metode [16]. Ini merupakan bagian kunci tertentu untuk komunikasi. Jika anda ingin berdiskusi tentang desain dengan seseorang, maka anda hanya membutuhkan bahasa pemodelan bukan proses yang digunakan untuk mendapatkan desain [16].

II.6.1 *Use Case* Diagram

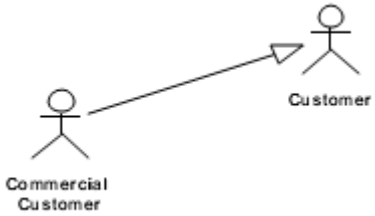
Menggambarkan sejumlah *external actors* dan hubungannya ke *use case*yang diberikan oleh sistem. *Use case* adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari simbol *use case*s namun dapat juga dilakukan dalam activity diagram. *Use case* digambarkan hanya yang dilihat dari luar oleh actor (keadaan lingkungan sistem yang dilihat user) dan bukan bagaimana fungsi yang ada di dalam system [16].

Komponen-komponen pembentuk diagram *use case* adalah sebagai berikut [16]:

1. *Actor*

   Pada dasarnya *actor* bukanlah bagian dari use case diagram, namun untuk dapat terciptanya suatu use case diagram diperlukan beberapa *actor* dimana *actor* tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. Sebuah *actor* mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari

sistem atau keduanya menerima dan memberi informasi pada sistem, *actor* hanya berinteraksi dengan *use case* tetapi tidak memiliki kontrol atas *use case*. *Actor* digambarkan dengan *stick man* [16].



Gambar II-7 *Actor* [16]

2. *Use Case*

*Usecase*adalah gambaran fungsionalitas dari suatu sistem, sehingga customer atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.



Gambar II-8 *Use Case* [16]

3. Relasi dalam *Use Case*

Ada beberapa relasi yang terdapat pada *use case* diagram:

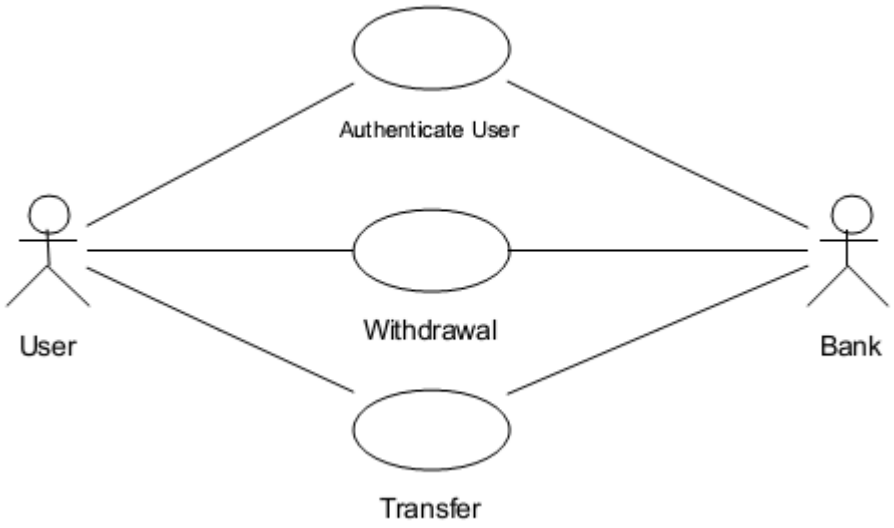
a. *Association*, menghubungkan link antar element.
b. *Generalization*, disebut juga i*nheritance*(pewarisan), sebuah elemen dapatmerupakan spesialisasi dari elemen lainnya.
c. *Dependency*, sebuah element bergantung dalam beberapa cara ke element lainnya.
d. *Aggregation*, bentuk assosiation dimana sebuah elemen berisi elemen lainnya.

Tipe relasi/ stereotype yang mungkin terjadi pada *use case* diagram:

a. <<include>>, yaitu kelakuan yang harus terpenuhi agar sebuah event dapat terjadi, dimana pada kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya.
b. <<extends>> , kelakuan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan alarm.

c. <<communicates>>, mungkin ditambahkan untuk asosiasi yang menunjukan asosiasinya adalah *communicates association*. Ini merupakan pilihan selama asosiasi hanya tipe ralationship yang dibolehkan antara *actor* dan *use case*.

Contoh Dari *Use Case* diagram dapat dilihat pada [16] :



Gambar II-9 Diagram *Use Case* [16]

II.6.2 *Class* Diagram

Class adalah deskripsi sekelompok *object* dari *property* (atribut), sifat(operasi),relasi antar objectdan skematik yang umum. *Class* merupakan template untuk membentuk object. Setiap object merupakan contoh dari beberapa *class* dan *object* tidak dapat menjadi contoh lebih dari satu *class* [16] :

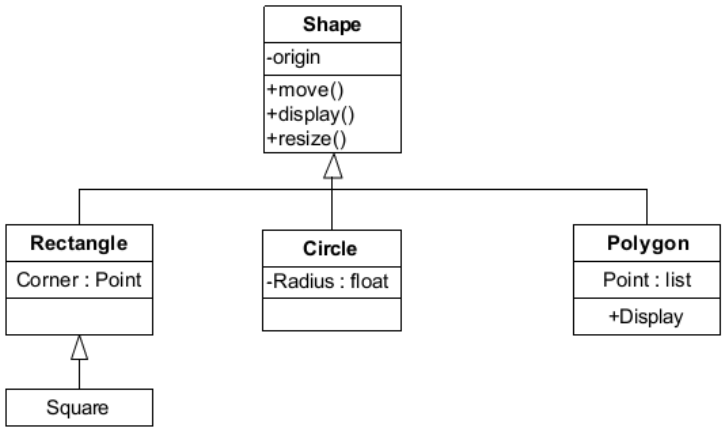
Class memiliki tiga area pokok yaitu Nama, Atribut, Metoda. Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. *Private*, tidak dapat dipanggil dari luar classyang bersangkutan
2. *Protected*, hanya dapat dipanggil oleh classyang bersangkutan dan anak-anak yang mewarisinya.
3. *Public*, dapat dipanggil oleh siapa saja

Antara *class* yang satu dan *class* yang lainnya dapat memiliki hubungan atau relasi antar kelas sebagai berikut :

1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *clas s*lain, atau yang harus mengetahui eksistensi *class* lain. Panah *navigability* merupakan penunjuk arah *query* antar *class*.
2. Agregasi, yaitu hubungan yang menyatakan bagian ("terdiri atas..") dari suatu *class*.
3. Pewarisan, yaitu hubungan hirarki antar *class*. *Class*dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-passing dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence* diagram.

Contoh dari *class* dan relasi *class* dapat dilihat pada [16]:



Gambar II-10 Diagram *Class* [16]

II.6.3 *Activity* Diagram

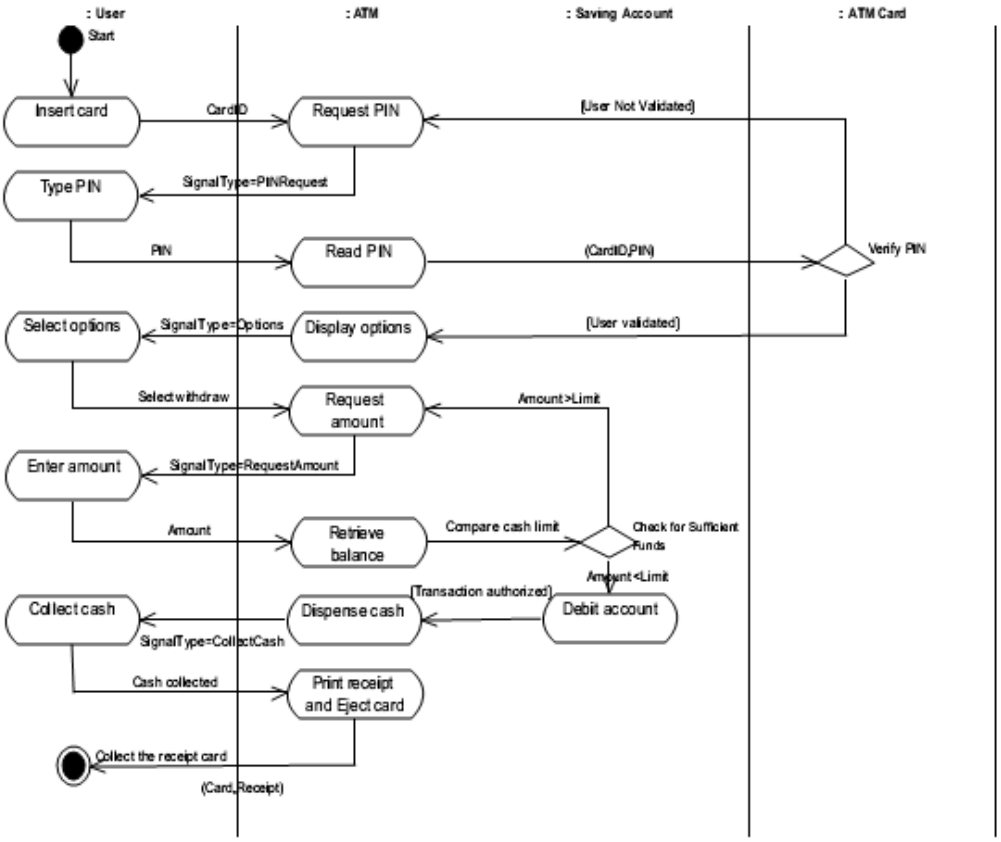
Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

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 jelas, 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.

Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan 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.

Contoh *activity* diagram dapat dilihat pada [16]:



Gambar II-11 *Activity* Diagram [16]

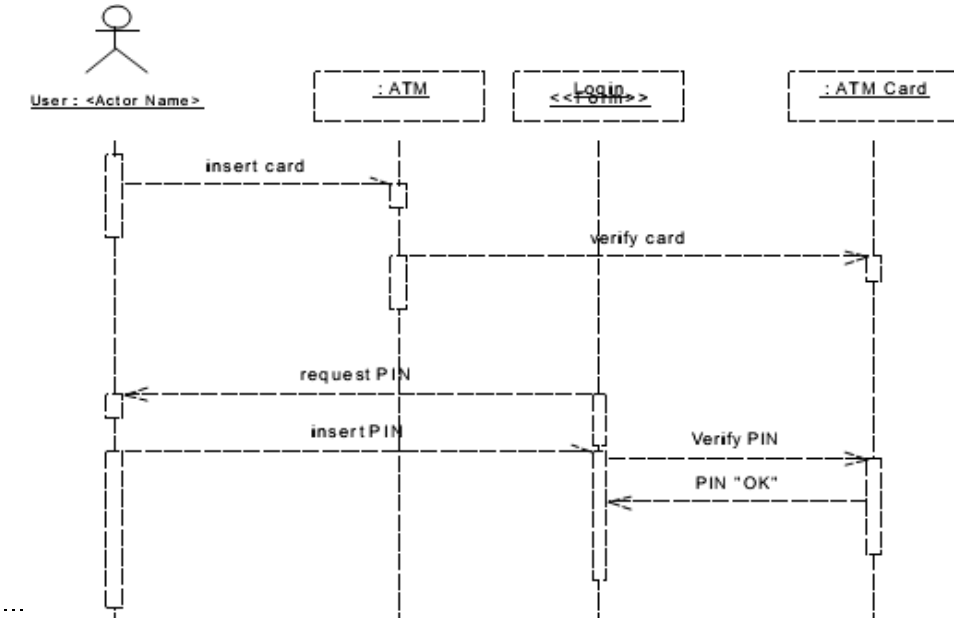
II.6.4 *Sequence* Diagram

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 *life line vertikal*. *Messages* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *messages* akan dipetakan menjadi operasi/metoda dari *class*. *Activation* baru menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah messages.

Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan icon khusus untuk *objek boundary*, *controller* dan *persistent entity*.

Contoh sequence diagram dapat dilihat pada gambar berikut :



Gambar II-12 *Sequence* Diagram [16]

II.7 Skala *Likert*

Skala *Likert* digunakan untuk mengukur sikap dalam suatu penelitian. Yang dimaksud dengan sikap menurut Thurstone ialah [17]:

1. Pengaruh atau penolakan
2. Penilaian
3. Suka atau tidak suka
4. Kepsitifan atau kenegatifan terhadap suatu obyek

Biasanya sikap dalam skala *Likert* diekspresikan mulai dari yang paling negatif, netral sampai ke yang paling positif dalam bentuk sebagai berikut [17]:

1. Sangat tidak setuju
2. Tidak setuju
3. Tidak tahu (netral)
4. Setuju
5. Sangat setuju

Untuk melakukan kuantifikasi maka skala tersebut kemudian diberi angka-angka sebagai simbol agar dapat dilakukan perhitungan, dapat dilihat pada tabel berikut [17]:

Tabel II-1 Skor Skala Likert

Skala	Skor
Sangat Tidak Setuju	1
Tidak Setuju	2
Tidak Tahu (netral)	3
Setuju	4
Sangat Sertuju	5

II.8 Uji T Dependen

Pada dasarnya uji T digunakan untuk membandingkan rata-rata dua populasi dengan sekala interval [17]. Pada uji T dependen di fokuskan pada situasi penelitian yang menuntut pengujian perbedaan dua mean populasi yang saling berkorelasi [18]. Contoh yang umum ditemui adalah desain pra uji–pasca uji

(pretest–posttest), dimana untuk mengkaji perubahan yang terjadi akibat suatu perlakuan, kita dapat membandingkan perilaku atas kemampuan subjek penelitian sebelum dan sesudah perlakuan diberikan.

Adapun langkah – langkah pengerjaan teknik statistik uji t dependen adalah sebagai berikut [18]:

1. Buatlah H_1 dan H_0 dalam bentuk hipotesis statistic
2. Buatlah H_1 dan H_0 dalam bentuk kalimat
3. Masukkan angka-angka statistik dari tabel distribusi.
4. Tentukan besarnya D dan D^2 (dalam kolom tabel distribusi)
5. Hitung besarnya SD (standar deviasi) dengan rumus :

   $$SD = \sqrt{\frac{\sum D^2 - [(\sum D)^2 np]}{np-1}}$$

6. Hitung besarnya kesalahan baku distribusi sampling (Standard error of the sampling distribution of differences), dengan rumus berikut :

   $$SD = \sqrt{\frac{SD}{np}}$$

7. Uji perbedaan dengan menggunakan rumus uji t dependen berikut ini :

   $$\text{Uji } t = \frac{X1 - X2}{SD}$$

8. Menguji tingkat kesalahan (alpha) = 5% atau 1% dengan rumus berikut ini :
   db = n-1
9. Bandingkan hasil t hitung dengan t tabel, dengan ketentuan jika t hitung > t tabel maka Hasil signifikan dan H_0 diterima, sebaliknya H_0 ditolak jika t hitung < t tabel maka hasil tidak signifikan, H_1 ditolak dan H_0 diterima.
10. Berikan kesimpulan dalam bentuk kalimat.