BAB 2

TINJAUAN PUSTAKA

2.1 Pembelajaran Berbantuan Komputer

Pembelajaran Berbantuan Komputer merupakan sarana yang baik digunakan dalam proses belajar mengajar karena dapat menjadikan pembelajaran menjadi lebih efektif dan efisien [6]. Perkembangan teknologi khususnya di bidang Teknologi Informasi yang semakin pesat serta semakin ketatnya persaingan mengharuskan para pendidik harus lebih inovatif dalam pengoptimalan proses pembelajaran. Untuk itulah diperlukan Sistem Pembelajaran Berbantuan Komputer.

2.1.1 Pengertian Pembelajaran Berbantuan Komputer

Pembelajaran Berbantuan Komputer (CAI) terkait langsung dengan pemanfaatan komputer dalam (kegiatan) pembelajaran didalam dan diluar kelas, baik secara individu maupun secara kelompok. CAI dapat diartikan sebagai bentuk pembelajaran yang menempatkan komputer dalam peran guru, dimana siswa berinteraksi secara langsung dengan komputer dan kontrol sepenuhnya ditangan siswa sehingga memungkinkan siswa belajar sesuai kemampuan dan memilih materi (pembelajaran) sesuai kebutuhannya [2].

Pembelajaran Berbantuan Komputer diambil dari istilah *Computer Aided Instruction* (CAI). Menurut Herman D. Sujono seperti dikutip oleh Idris istilah CAI (Computer Aided Instruction) menunjuk pada semua *software* pendidikan yang diakses melalui komputer yang dimana anak didik diarahkan untuk dapat berinteraksi dengan dengannya. Ada juga istilah-istilah lain, istilah-istilah itu

antara lain *Computer Based Intruction* (CBI), *Computer Assisted Learning* (CAL), dan *Computer Based Education* (CBE) [3].

2.1.2 Tujuan Pemakaian Komputer dalam Proses Pembelajaran

Menurut Sidik, dkk, tujuan pemakaian komputer dalam proses pembelajaran meliputi [6]:

1. Tujuan Kongitif

Komputer dapat mengajarkan konsep-konsep aturan, prinsip, langkah-langkah, proses, dan kalkulasi yang komplek. Komputer juga dapat menjelaskan konsep tersebut secara sederhana dengan penggabungan *visual* dan *audio* yang dianimasikan. Sehingga cocok untuk pembelajaran secara mandiri.

2. Tujuan Psikomotor

Dengan pembelajaran yang dikemas dalam bentuk *game* dan simulasi sangat bagus digunakan untuk menciptakan kondisi dunia kerja.

3. Tujuan Afektif

Bila program didesain secara tepat dengan memberikan unsur *audio* dan *video* yang isinya menggugah perasaan, pembelajaran sikap/afektif pun dapat dilakukan menggunakan media komputer.

2.1.3 Penerapan Pembelajaran Berbantuan Komputer

Menurut Wihardjo, CAI (*Computer Aided Instruction*) perlu dilakukan pada situasi-situasi sebagai berikut [2]:

1. Biaya dan metode lain (lebih) mahal.
2. Keamanan kurang terjamin.
3. Materi sangat sulit diajarkan dengan metode lain.

4. Praktik siswa secara individual sangat diperlukan.

5. Motivasi siswa kurang.

6. Terdapat kesulitan yang logis dalam pembelajaran konvensional.

2.1.4 Jenis Pembelajaran Berbantuan Komputer

Menurut Strickland Patterson seperti dikutip oleh Eva Handriyantini, S.Kom, M.MT, ada lima tipe Pembelajaran Berbantuan Komputer (*Computer Aided Instruction*) yaitu [6]:

1. *Drill & Practice*

menyajikan materi pelajaran untuk dipelajari secara berulang.

2. *Tutorial*

menyajikan materi yang telah diajarkan atau materi baru yang akan dipelajari.

3. *Simulation*

memberi kesempatan untuk menguji kemampuan pada aplikasi nyata dengan menciptakan situasi yang mengikut-sertakan siswa-siswa untuk bertindak pada situasi tersebut.

4. *Problem Solving*

menyajikan masalah-masalah untuk siswa menyelesaikannya berdasarkan kemampuan yang mereka peroleh.

5. *Education Game*

merupakan paket *software* yang menciptakan kemampuan pada lingkungan permainan yang diberikan sebagai alat bantu untuk memotivasi atau membantu siswa untuk melalui prosedur permainan secara teliti untuk

mengembangkan kemampuannya. Berdasarkan kriteria-kriteria ini, maka aplikasi yang dirancang penulis dapat ditentukan sebagai aplikasi *educational game*.

2.1.5 Tahapan Pengembangan Pembelajaran Berbantuan Komputer

Menurut Luther seperti dikutip oleh Sutopo [6], pengembangan Pembelajaran Berbantuan Komputer berbasis multimedia meliputi tahap-tahap.

1. Konsep (*Concept*)

Dalam tahap ini dilakukan identifikasi perkiraan kebutuhan yang dihasilkan dari tahap pengamatan pada penelitian awal.

2. Rancangan (*Design*)

Dalam tahap ini dibuat desain *visual* tampilan *screen, interface, script* atau cerita, *storyboard* dan struktur navigasi.

3. Pengumpulan Materi (*Collecting Content Material*)

Pada tahap ini dilakukan pengumpulan bahan seperti gambar, animasi, *audio* dan *video* berikut pembuatan gambar, grafik, dan lain-lain yang diperlukan untuk tahap berikutnya

4. Perakitan (*Assembly*)

Dalam tahap ini dilakukan pembuatan ilustrasi, *audio* dan *video*, serta pembuatan aplikasi berdasarkan *story board* dan struktur navigasi yang berasal dari tahap *design*. Dalam tahap ini juga dilakukan pembuatan program.

5. Pengujian (*Testing*)

Dalam pengembangan multimedia perlu dilakukan *testing* (uji coba) setelah produksi.

6. Distribusi (*Distribution*)

Dalam tahap ini dilakukan pembuatan pedoman penggunaan model pembelajaran, kemasan, dan dokumentasi.

2.1.6 Kelebihan Pembelajaran Berbantuan Komputer

Menurut Wiharjo, kelebihan komputer sebagai media pembelajaran adalah [2]:

1. Meningkatkan perhatian dan konsentrasi siswa.
2. Meningkatkan motivasi siswa.
3. Menyesuaikan materi dengan kemampuan siswa.
4. Mereduksi penggunaan waktu penyampaian materi.
5. Dapat mengakomodasi banyak siswa dan menjalankan fungsinya dengan sedikit kesalahan.
6. Bersifat tanggap dan bersahabat sehingga siswa belajar tanpa tekanan psikologis.
7. Materi dapat didesain lebih menarik.
8. Tingkat kemampuan dan kecepatan belajar dapat dikontrol oleh siswa sehingga siswa dapat belajar dan berprestasi sesuai dengan kemampuannya.

2.2 Multimedia

Konsep multimedia telah banyak diterapkan dalam dunia pendidikan. Pembelajaran menggunakan multimedia interaktif berkembang atas dasar pembelajaran konvensional yang tidak bisa memenuhi kebutuhan peserta didik dalam pembelajaran, Sigit, dkk, [6]. Multimedia dapat menjadikan suatu aplikasi menjadi sangat interaktif dan menyajikan *interface* yang menarik.

2.2.1 Pengertian Multimedia.

Multimedia merupakan kombinasi dari teks, gambar, seni grafik, suara, animasi dan elemen-elemen *video* yang dimanipulasi secara *digital*. Tampilan dan cita rasa dari proyek multimedia harus menyenangkan, estetis, mengundang dan mengikat. Proyek harus memuat konsistensi visual, hanya dengan menggunakan elemen-elemen yang mendukung pesan keseluruhan dari program[6].

Menurut Gayestik seperti dikutip oleh Idris, multimedia merupakan suatu sistem komunikai interaktif berbasis komputer yang mampu menciptakan, menyimpan, menyajikan, dan mengakses kembali informasi berupa teks, grafik, suara, *video*, atau animasi[3].

2.2.2 Kriteria Multimedia dalam Pembelajaran

Menurut Sigit dkk, kriteria multimedia adalah sebagai berikut [6]:

1. Memilki lebih dari satu media yang konvergen, misalnya menggabungkan unsur *audio* dan *visual*.
2. Bersifat interaktif, dalam pengertian memiliki kemampuan untuk mengakomodasi respon pengguna.
3. Bersifat mandiri, dalam pengertian memberi kemudahan dan kelengkapan isi sedemikian rupa sehingga pengguna bisa menggunakan tanpa bimbingan orang lain.

2.2.3 Jenis Multimedia

Menurut Sigit dkk, multimedia terbagi menjadi dua jenis, yaitu [6] :

1. Multimedia Linier

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

2. Multimedia Interaktif

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

2.2.4 Kelebihan Pembelajaran Menggunakan Multimedia

Menurut Townsend seperti dikutip oleh Idris, multimedia memliki beberapa keuntungan, yaitu [3]:

1. Multimedia masuk akal sehingga dapat meningkatkan pembelajaran.
2. Multimedia meningkatkan ekpresi diri dengan membiarkan pelajar untuk memutuskan sendiri.
3. Multimedia membuat pelajar menjadi ’pemilik’ sehingga mereka bisa menciptakan apa yang hendak mereka pelajari.
4. Multimedia menciptakan suasana yang aktif, sehingga pelajar dapat terlibat langsung.
5. Multimedia dapat menjembatani komunikasi pelajar dengan instruktur.

6. Pemakaian multimedia sudah tidak asng lagi karena telah digunakan dalam kehidupan sehari-hari seperti *video game* dan *televisi*.

2.3 *Game* Edukasi

Penerapan *education game* bermula dari perkembangan *video game* yang sangat pesat dan menjadikannya sebagai media efektif yang interaktif dan banyak dikembangkan di perindustrian. Melihat kepopuleran *game* tersebut, para pendidik berpikir bahwa mereka mempunyai kesempatan yang baik untuk menggunakan komponen rancangan *game* dan menerapkannya pada kurikulum dengan penggunaan industri berbasis *game*. *Game* harus memiliki desain antarmuka yang interaktif dan mengandung unsur menyenangkan [6].

2.3.1 Pengertian *Game*

Sebuah *game* adalah aktivitas terstuktur atau semi terstuktur, yang biasanya dilakukan untuk bersenang senang dan kadang juga di gunakan sebagai alat pembelajaran. Sebuah *game* bisa di karakteristik dari "apa yang pemain lakukan". Hal tersebut adalah sebagai berikut[6]:

a. Peralatan, seperti bola, kartu, atau sebuah komputer
b. Peraturan, biasa menentukan giliran permainan, hak dan kewajiban pemain dan tujuan permainan
c. Skill strategi dan keburuntungan. *Game* dengan skill misalnya kekuatan fisik dan kekuatan mental.
d. *Single Player Game* (permainan satu orang). Permainan tunggal ini adalah permaian dengan keahlian, berpacu dengan waktu atau keberuntungan.

Pengertian *game* komputer menurut beberapa ahli [6]:

1. Menurut Chris Crawford, seorang *computer game designer* mengemukakan bahwa *game*, pada intinya adalah sebuah interaktif, aktivitas yang berpusat pada sebuah pencapaian, ada pelaku aktif, ada lawan anda.
2. Menurut David Parlett, *Game* adalah sesuatu yang memiliki "akhir dan cara mencapainya", artinya ada tujuan, hasil dan serangkaian peraturan untuk mencapai keduanya.
3. Menurut Roger Caillois, seorang sosiolog Perancis, dalam bukunya yang berjudul *Les jeux et les hommes* menyatakan *game* adalah aktivitas yang mencakup karakteristik berikut: *fun* (bebas bermain adalah pilihan bukan kewajiban), *separate* (terpisah), *uncertain*, *non-productive*, *governed by rules* (ada aturan), *fictitious* (pura-pura).

2.3.2 Pengertian Edukasi

Edukasi adalah proses yang dilakukan oleh seseoarang untuk menemukan jati dirinya, yang dilakukan dengan mengamati dan belajar yang kemudian melahirkan tindakan dan prilaku. Edukasi sebenarnya tidak jauh berbeda dari belajar yang dikembangkan oleh aliran *behaviorisme* dalam psikologi. Hanya istilah ini sering dimaknai dan diinterpretasikan berbeda dari learning yang bermakna belajar. Dan istilah ini seringkali digunakan dalam pendekatan pendidikan yang tentu maknanya lebih dari sekedar belajar.

Dalam teori belajar dan prilaku sebuah tindakan akan didapatkan dari sebuah aktifitas pengamatan yang dikondisikan dan direkayasa dengan tujuan hasil yang sudah ditentukan dan bisa diukur secara logis dan ilmiah. Dalam

belajar ada hubungan erat antara penguat, baik yang positif atau negatif yang terjadi setelah tindakan dilakukan, atau setelah menjadi sebuah perilaku konkrit yang bisa dilihat dengan kasat mata. Stimulan yang terjadi sebelum perilaku muncul sangat mempengaruhi tujuan yang diharapakan. Ini artinya perilaku seseoarang harus selalu bisa diukur dan selalu pula diberikan penguat dan stimulasi yang diciptakan sedimikian rupa sesuai dengan tujuan yang hendak dicapai.

Edukasi dalam pendekatan pendidikan lebih kepada pemberian kondisi dan pengaruh kepada seseorang agar bisa menciptakan perilaku spontan dalam jangka waktu yang lama sesuai dengan harapan diri sendiri dan masyarakat. Harapan itu diukur dengan kesepakatan-kesepakatan formal yang disebut hukum, dan kesepakatn-kesepakatan yang tidak formal tetapi menjadi kebiasaan yang dianggap baik berlaku di masyarakat yang disebut adat. Inilah perilaku yang diharapkan dari edukasi yang sepadan dengan kata *learning* dalam pandangan psikologi *behaviorisme*[6].

2.3.3 Pengertian *Game* Edukasi

Education 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 [6].

2.3.4 Kriteria Edukasi *Game*

Menurut Hurd dan Jenuings, perancangan *Education game* yang baik haruslah memenuhi kriteria dari *education game* itu sendiri. Berikut ini adalah beberapa kriteria dari sebuah *education game*, yaitu[6]:

1. Nilai Keseluruhan (*Overall Value*)

Nilai keseluruhan dari suatu *game* terpusat pada desain dan panjang durasi *game*. Aplikasi ini dibangun dengan desain yang menarik dan interaktif. Untuk penentuan panjang durasi, aplikasi ini menggunakan fitur *timer* .

2. Dapat Digunakan(*Usability*)

Mudah digunakan dan diakses adalah poin penting bagi pembuat *game*. Apliksi ini merancang sistem dengan *interface* yang *user friendly* sehingga *user* dengan mudah dapat mengakses aplikasi.

3. Keakuratan (*Accuracy*)

Keakuratan diartikan sebagai bagaimana kesuksesan model/gambaran sebuah *game* dapat dituangkan ke dalam percobaan atau perancangannya. Perancangan aplikasi ini harus sesuai dengan model *game* pada tahap perencanaan.

4. Kesesuaian (*Appropriateness*)

Kesesuaian dapat diartikan bagaimana isi dan desain *game* dapat diadaptasikan terhadap keperluan *user* dengan baik. Aplikasi ini menyediakan menu dan fitur yang diperlukan *user* untuk membantu pemahaman *user* dalam menggunakan aplikasi.

5. Relevan (*Relevance*)

Relevan artinya dapat mengaplikasikan isi *game* ke target *user*. Agar dapat relevan terhadap *user*, sistem harus membimbing mereka dalam pencapaian tujuan pembelajaran. Karena aplikasi ini ditujukan untuk anak-anak maka desain antarmuka harus sesuai dengan nuansa anak-anak, yaitu menampilkan warna-warna yang ceria.

6. Objektifitas (*Objectives*)

Objektifitas menentukan tujuan *user* dan kriteria dari kesuksesan atau kegagalan. Dalam aplikas ini objektivitas adalah usaha untuk mempelajari hasil dari permainan.

7. Umpan Balik (*Feedback*)

Untuk membantu pemahaman *user* bahwa permainan (*performance*) mereka sesuai dengan objek *game* atau tidak, *feedback* harus disediakan. Aplikasi ini menyajikan animasi dan efek suara yang mengindikasikan kesuksesan atau kegagalan permainan.

2.3.5 Hubungan Konsep Multimedia dengan *Education Game*

Konsep multimedia dengan aplikasi *education game* memiliki hubungan yang cukup erat. Menurut Vaughan, multimedia merupakan kombinasi dari teks, gambar, seni grafik, suara, animasi dan elemen-elemen *video* yang dimanipulasi secara digital. Sedangkan menurut Hurd dan Jenuings, *game* harus memiliki desain antarmuka yang interaktif dan mengandung unsur menyenangkan. Desain interaktif sehingga menimbulkan kesenangan bagi *user* dalam aplikasi *education*

game dapat dibangun berdasarkan unsur-unsur multimedia tadi. Oleh karena itu, aplikasi *education game* tidak pernah terlepas dari konsep multmedia.

2.4 Sekolah Menengah Pertama

Sekolah menengah pertama (disingkat SMP, Bahasa Inggris: *junior high school*) adalah jenjang pendidikan dasar pada pendidikan formal di Indonesia setelah lulus sekolah dasar (atau sederajat). Sekolah menengah pertama ditempuh dalam waktu 3 tahun, mulai dari kelas 7 sampai kelas 9. Pada tahun ajaran 1994/1995 hingga 2003/2004, sekolah ini pernah disebut sekolah lanjutan tingkat pertama (SLTP).[10]

Pelajar sekolah menengah pertama umumnya berusia 13-15 tahun. Di Indonesia, setiap warga negara berusia 7-15 tahun tahun wajib mengikuti pendidikan dasar, yakni sekolah dasar (atau sederajat) 6 tahun dan sekolah menengah pertama (atau sederajat) 3 tahun.[10]

Sekolah menengah pertama diselenggarakan oleh pemerintah maupun swasta. Sejak diberlakukannya otonomi daerah pada tahun 2001, pengelolaan sekolah menengah pertama negeri di Indonesia yang sebelumnya berada di bawah Departemen Pendidikan Nasional, kini menjadi tanggung jawab pemerintah daerah kabupaten/kota. Sedangkan Departemen Pendidikan Nasional hanya berperan sebagai regulator dalam bidang standar nasional pendidikan. Secara struktural, sekolah menengah pertama negeri merupakan unit pelaksana teknis dinas pendidikan kabupaten/kota. Adapun Mata Pelajaran yang ada di Sekolah Menengah Pertama diantaranya : [10]

1. Agama
2. Kewarganegaraan
3. Jasmani dan Kesehatan
4. Teknologi Informatika dan Komunikasi
5. Bahasa Indonesia
6. Bahasa Inggris
7. Bahasa Daerah
8. Bahasa Asing
9. Matematika
10. Ilmu Pengetahuan Alam
    1. Fisika
    2. Biologi
    3. Kimia
11. Sejarah
12. Ilmu Pengetahuan Sosial
    1. Geografi
    2. Ekonomi
13. Seni Budaya dan Keterampilan

2.5 Ilmu Pengetahuan Alam

Ilmu alam (bahasa Inggris: *natural science*; atau ilmu pengetahuan alam) adalah istilah yang digunakan yang merujuk pada rumpun ilmu dimana obyeknya adalah benda-benda alam dengan hukum-hukum yang pasti dan umum, berlaku kapan pun dimana pun. [1]

Ilmu alam mempelajari aspek-aspek fisik dan non-manusia tentang Bumi dan alam sekitarnya. Ilmu-ilmu alam membentuk landasan bagi ilmu terapan, yang keduanya dibedakan dari ilmu sosial, humaniora, teologi, dan seni.[1]

Matematika tidak dianggap sebagai ilmu alam, akan tetapi digunakan sebagai penyedia alat/perangkat dan kerangka kerja yang digunakan dalam ilmu-ilmu alam. Istilah ilmu alam juga digunakan untuk mengenali "ilmu" sebagai disiplin yang mengikuti metode ilmiah, berbeda dengan filsafat alam. Di sekolah, ilmu alam dipelajari secara umum di mata pelajaran Ilmu Pengetahuan Alam (biasa disingkat IPA).[1]

2.5.1 Matematika

Carl Friedrich Gauss mengatakan matematika sebagai "Ratunya Ilmu Pengetahuan". Di dalam bahasa aslinya, Latin *Regina Scientiarum*, juga di dalam bahasa Jerman *Königin der Wissenschaften*, kata yang bersesuaian dengan ilmu pengetahuan berarti (lapangan) pengetahuan. Jelas, inipun arti asli di dalam bahasa Inggris, dan tiada keraguan bahwa matematika di dalam konteks ini adalah sebuah ilmu pengetahuan. Pengkhususan yang mempersempit makna menjadi ilmu pengetahuan alam adalah di masa terkemudian. Bila seseorang memandang ilmu pengetahuan hanya terbatas pada dunia fisika, maka matematika, atau sekurang-kurangnya matematika murni, bukanlah ilmu pengetahuan.[11]

Albert Einstein menyatakan bahwa "*sejauh hukum-hukum matematika merujuk kepada kenyataan, maka mereka tidaklah pasti; dan sejauh mereka pasti, mereka tidak merujuk kepada kenyataan*." [11]

2.5.2 Biologi

Biologi atau ilmu hayat adalah ilmu yang mempelajari aspek fisik kehidupan. Istilah "biologi" dipinjam dari bahasa Belanda, *biologie*, yang juga diturunkan dari gabungan kata bahasa Yunani, *βίος*, *bios* ("hidup") dan *λόγος,logos* ("lambang", "ilmu"). Istilah "ilmu hayat" dipinjam dari bahasa Arab, juga berarti "ilmu kehidupan". Obyek kajian biologi pada masa kini sangat luas dan mencakup semua makhluk hidup dalam berbagai aspek kehidupannya. [13]

Berbagai cabang biologi mengkhususkan diri pada setiap kelompok organisme, seperti *botani* (ilmu tentang tumbuhan), *zoologi* (ilmu tentang hewan), dan *mikrobiologi* (ilmu tentang jasad renik). Perbedaan-perbedaan dan pengelompokan berdasarkan ciri-ciri fisik kelompok organisme dipelajari dalam sistematika, yang di dalamnya mencakup pula *taksonomi* dan *paleobiologi*.[13]

2.5.3 Fisika

Fisika (bahasa Yunani: φυσικός (fysikós), "alamiah", dan φύσις (fýsis), "alam") adalah sains atau ilmu tentang alam dalam makna yang terluas. Fisika mempelajari gejala alam yang tidak hidup atau materi dalam lingkup ruang dan waktu. Para fisikawan atau ahli fisika mempelajari perilaku dan sifat materi dalam bidang yang sangat beragam, mulai dari partikel submikroskopis yang membentuk segala materi (fisika partikel) hingga perilaku materi alam semesta sebagai satu kesatuan kosmos.[12]

Beberapa sifat yang dipelajari dalam fisika merupakan sifat yang ada dalam semua sistem materi yang ada, seperti hukum kekekalan energi. Sifat semacam ini sering disebut sebagai hukum fisika. Fisika sering disebut sebagai

"ilmu paling mendasar", karena setiap ilmu alam lainnya (biologi, kimia, geologi, dan lain-lain) mempelajari jenis sistem materi tertentu yang mematuhi hukum fisika. Misalnya, kimia adalah ilmu tentang molekul dan zat kimia yang dibentuknya. Sifat suatu zat kimia ditentukan oleh sifat molekul yang membentuknya, yang dapat dijelaskan oleh ilmu fisika seperti mekanika kuantum, termodinamika, dan elektromagnetika.[12]

2.6 AI (Artificial Intelligence)

Kecerdasan buatan berasal dari bahasa inggris "*artificial intelligence*" atau di singkat AI . *Intelligence* adalah kata sifat yang berarti cerdas , sedangkan *Artificial* artinya buatan. Kecerdasan buatan yang di maksudkan di sini merujuk pada mesin yang mampu berfikir, menimbang tindakan yang akan di ambil dan mampu mengambil keputusan seperti yang telah di definiskan oleh beberapa ahli.[14]

"*Kecerdasan buatan (artificial intelligence) merupakan kawasan penelitian, aplikasi dan instruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia adalah cerdas*"[14].

"*Kecerdasan Buatan (AI) merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia.*" [14].

2.6.1 Macam-Macam Bidang AI (Artificial Intelligence)

Beberapa macam bidang yang menggunakan kecerdasan buatan antara lain:[14]

1. Sistem Pakar (*Expert System*)

   Sistem pakar adalah suatu sistem yang dibuat berdasarkan orang yang benar-benar ahli dalam bidang dan berpengalaman.

2. Pengolahan Bahasa Alami (*Natural Language Processing*)

   Pengolahan Bahasa Alami (*natural language processing*) adalah suatu bidang yang dimana AI tersebut mengolah perintah bahasa sehari-hari ke dalam AI tersebut agar melakukan sesuatu yang sesuai dengan perintah tersebut.

3. Pengenalan Ucapan (*Speech Recognition*)

   Pengenalan ucapan adalah suatu bidang AI yang mengubah bentuk suara dalam suatu format *file* agar dapat dibaca dan sebaliknya mengubah suatu *text file* menjadi suara oleh pengguna misalnya seperti telepon untuk orang bisu dan tuli.

4. Robotika dan Sistem Sensor (*Robotics dan Sensory Systems*)

   Robotika dan sensor adalah bidang AI yang berfungsi sebagai mendeteksi suatu keberadaan benda sehingga AI tersebut dapat bergerak sesuai dengan program yang telah dirancang.

5. *Computer Vision*

Computer Vision adalah bidang AI yang berfungsi untuk mengetahui hasil inputan dan di keluarkan dalam bentuk visual misalnya pengenalan pola sidik jari dan pengenalan tulisan.

6. *Intelligent Computer-aided Instruction*

Intelligent computer-aided instruction adalah bidang AI yang berfungsi sebagai tutor atau membantu dalam pembelajaran di bidang tertentu.

7. *Game Playing* sistem pakar.

Game Playing adalah bidang AI yang dibuat dalam permainan misalnya *Deep blue. Deep blue* adalah program AI yang dibuat untuk permainan catur.

2.6.2 Sudut Pandang AI (*Artificial Intelligence*)

AI (*Artificial Intelligence*) dari berbagai sudut pandang antara lain:[14]

1. Sudut pandang kecerdasan : mesin menjadi ‘cerdas’ (mampu berbuat apa yang dilakukan oleh manusia).
2. Sudut pandang penelitian : studi bagaimana membuat agar komputer dapat melakukan sesuatu sebaik yang dilakukan oleh manusia.
    1. Domain penelitian :
        a. *Mundande task*
            1. Persepsi (*vision & speech*)
            2. Bahasa alami (*understanding, generation & translation*)
            3. Pemikiran yang bersifat *commonsense*
            4. Robot control

b. Formal task

1. Permainan/*games*
2. Matematika (geometri, logika, kalkulus, integral, pembuktian)

c. Expert task

1. Analisis *financial*
2. Analisis *medical*
3. Analisis ilmu pengetahuan
4. Rekayasa (desain, pencarian, kegagalan, perencanaan, manufaktur.

2.6.3 Paham Pemikiran AI (*Artificial Intelligence*)

Secara garis besar, AI terbagi ke dalam dua faham pemikiran yaitu AI konvensional dan AI komputasional (CI, *Computational Intelligence*). AI konvensional kebanyakan melibatkan metoda-metoda yang sekarang diklasifiksikan sebagai pembelajaran mesin, yang ditandai dengan formalisme dan analis statistik. Dikenal juga sebagai AI simbolis, AI logis, AI murni dan AI cara lama (GOFAI, *Good Old Fashioned Artificial Intelligence*). Metoda-metodanya meliputi:[14]

1. Sisem pakar, menerapkan kapabilitas pertimbangan untuk mencapai kesimpulan. Sebuah sistem pakar dapat memproses sejumlah besar informasi yang diketahui dan menyediakan kesimpulan-kesimpulan berdasarkan pada informasi-informasi tersebut.
2. Pertimbngan berdasarkan kasus.
3. Jaringan Bayesian.

4. AI berdasar tingkah laku, metoda modular pada pembentukan sistem AI secara manual.

Sedangkan AI komputasional melibatkan pengembangan atau pembelajaran iteratif (misalnya penalaan parameter seperti dalam sistem koneksionis). Pembelajaran ini berdasarkan pada data empiris dan diasosiasikan dengan AI non-simbolis, AI yang tak teratur dan perhitungan lunak. Metoda-metoda pokoknya meliputi:[14]

1. Jarigan Syaraf, sistem dengan kemampuan pengenalan pola yang sangat kuat.
2. Sistem *fuzzy*, teknik-teknik untuk pertimbangan di bawah ketidakpastian, telah digunakan secara meluas dalam industri modern dan sistem kendali produk konsumen.
3. Komputasi *Evolusioner*, menerapkan konsep-konsep yang terinspirasi secara biologis seperti populasi, mutasi dan "*survival of the fittest*" untuk menghasilkan pemecahan masalah yang lebih baik.

2.6.4 Konsep Penyelesaian Masalah dengan AI (*Artificial Intelligence*)

1. *Searching* (teknik pencarian) yaitu teknik penyelesaian masalah yang mempresentasikan masalah ke dalam ruang keadaan (*state*) dan secara sistematis melakukan pembangkitan dan pengujian *state-state* dari *initial state* sampai ditemukan suatu *goal state*.[14]
2. *Reasoning* (teknik penalaran) yaitu teknik penyelesaian masalah yang mempresentasikan masalah masalah ke dalam *logic* (*mathematics tools* yang digunakan untuk mempresentasikan dan memanipulasi fakta dan aturan).[14]

3. *Planning* suatu metode penyelesaian masalah dengan cara memecahkan masalah ke dalam sub-sub masalah yang lebih kecil, menyelesaikan masalah satu demi satu, kemudian menggabungkan soluis-solusi.[14]
4. *Learning* Secara otomatis menemkan aturan yang di harapkan bisa berlaku umum untuk data-data yang belum pernah kita ketahui.[14]

2.6.5 AI (*Artificial Intelligence*) Pada Game

1. *Behavior Trees* (Pohon Perilaku)

Behavior trees adalah kompromi kognitif antara gambar dan kata-kata. Mereka mengambil apa yang dijelaskan dalam kata-kata dan memberikan struktur yang dirancang untuk membantu efisiensi pemahaman dan memperjelas maksud asli. *Behavior trees* tidak selalu mengatakan lebih dari bahasa alami, atau matematika atau bentuk diagram lainnya. Sebaliknya, mereka adalah alternatif yang secara akurat dan jelas menggambarkan apa yang kita katakan dalam bahasa alami berurutan. *Behavior trees* menggunakan struktur (*non-sekuensial*) terintegrasi yang dirancang khusus untuk mengurangi beban ingatan jangka pendek kita dalam pemahaman informasi. *Behavior trees* biasanya dibangun dengan ketat oleh kata-kata terjemahan dari bahasa alami.[14]

Sebuah *Behavior trees* adalah, formal seperti pohon bentuk grafik yang mewakili perilaku individu atau jaringan entitas yang menyadari atau mengubah negara, membuat keputusan, peristiwa *respond-to/cause*, dan berinteraksi dengan bertukar informasi dan atau lewat kontrol.[14]

2. Collision detection

Collision detection adalah komponen penting pada game. Disadari atau tidak, komponen ini sering kali membuat game menjadi realistik, dan terkadang membuat pemain marah jika *collision detection* tidak berjalan dengan sesungguhnya.[14]

Collision detection terdiri dari beberapa teknik mulai dari *bounding box*, *reduce bounding box*, *multi bounding box*, *circle*, sampai penggunaan sudut dalam pendeteksian tabrakan.

Bounding box, meski memiliki algoritma yang super cepat, dan metode penyimpanan data yang tidak terlalu susah, namun *bounding box* memiliki kelemahan yang pokok, yakni menganggap area kosong sebagai solid area. Hal ini memberikan efek kesalahan deteksi *collision detection*. Kesalahan ini dikurangi dengan menggunakan *reduce bounding box*, namun sayang penggunaan *reduce bounding box* juga masih menyebabkan permasalahan yang hampir sama.

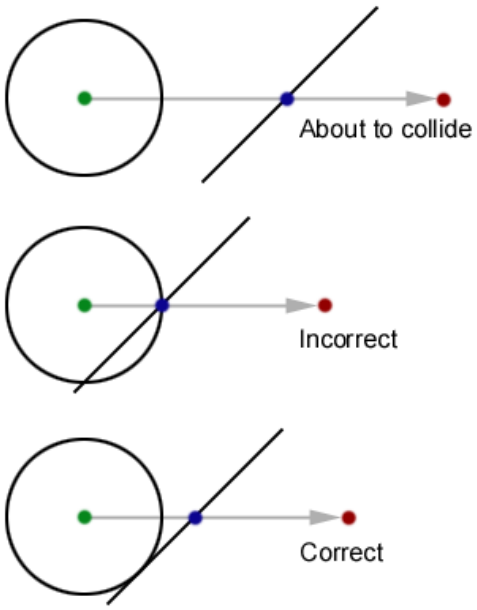
Penggunaan *multi bounding box* dirasa cukup bagus, di mana setiap area tubuh diberikan *bounding box*. Seperti tangan, kaki, kepala, dan badan diberikan *bounding box*-nya masing-masing. Cara ini pasti lebih lama dari *bounding box*, namun dirasa cukup mengurangi area kosong yang dianggap objek.

Cara lain selain *bounding box* adalah penggunaan lingkaran. Penggunaan lingkaran memiliki kelebihan bound memiliki jarak yang sama dengan pusat, sehingga dengan memanfaatkan sifat ini dapat dibuat algoritma yang lebih cepat dari deteksi bounding box. Namun sayang algoritma ini masih kurang bagus dalam mengatasi area kosong.

Penggunaan cara lain adalah dengan pendeteksian garis yang berpotongan, dan penggunaan sudut. Untuk cara yang ini, objek akan pertama kali dilist area edge terluar, untuk membatasi area badan, dan dunia luar.

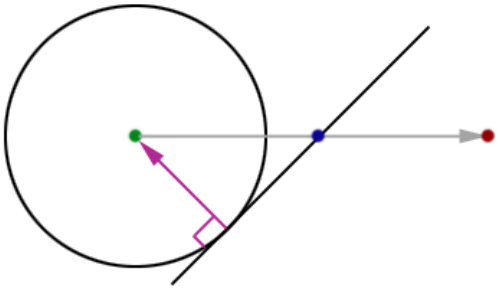
2.6.6 Teknik *Collision Detection* yang digunakan

Teknik *Collision detection* yang digunakan adalah *bounding box*, dibawah ini akan dijelaskan bagaimana *collision* terjadi antara lingkaran dan garis atau sebuah permukaan. [16]



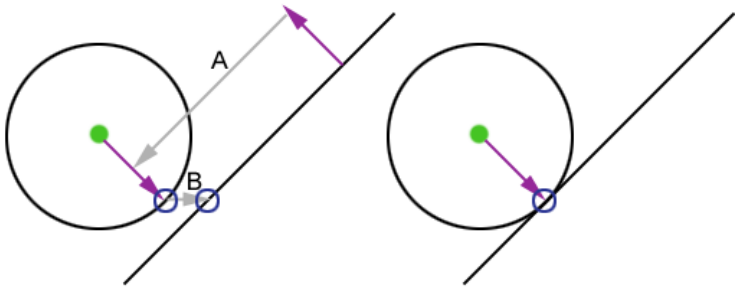
Gambar 2. 1 Sebuah lingkaran menabrak sebuah permukaan (garis)

Pada gambar 2.1 kita bisa melihat bahwa lingkaran bisa menabrak permukaan. Namun tubrukan lingkaran dengan permukaan tidak semudah yang kita bayangkan. Tubrukan tersebut bisa saja terjadi tidak benar yaitu ketika lingkaran sudah melewati garis.



Gambar 2. 2 Tubrukan yang benar antara garis dan lingkaran

Pada gambar 2.2, kita bisa melihat bahwa vector dibentuk dari titik perpotongan dan pusat dari lingkaran adalah terjadi bersamaan dengan garis normal dari garis. Kita bisa membalik proses ini untuk memutuskan titik pada garis yang terdapat pada lingkaran yang akan berpotongan dengan garis sebelum kita melakukan tubrukan. Pada gambar 2.3 akan diperlihatkan lebih jelas lagi.

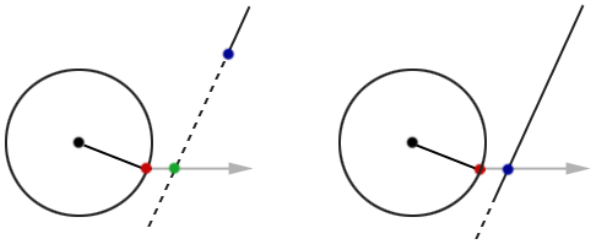


Gambar 2. 3 Menggunakan garis normal dari garis untuk menemukan titik potong lingkaran

Gambar 2.3 menunjukkan bahwa garis normal dari garis bisa digunakan untuk mendapatkan titik pada permukaan lingkaran yang akan beririsan dengan garis. Kita sebut dengan titik potong lingkaran.

Untuk mendapatkan titik potong lingkaran kita harus menentukan garis normal dari garis ke jari-jari lingkaran lingkaran. Lalu tarik garis abu-abu dari garis normal menuju garis jari-jari yang disebut sebagai garis A dan kita akan melihat bahwa hasilnya adalah vector titik potong lingkaran (yang berupa garis yang berlawanan arah dari garis normal dan diambil dari titik tengah lingkaran).

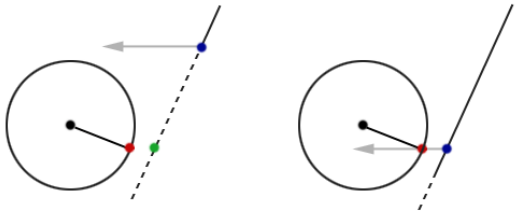
Dari titik yang didapatkan, kita pindahkan velocity vector (garis B berwarna abu-abu), hingga kita memotong garis. Sehingga lingkaran kita akan menuju garis dengan baik. [16]



Gambar 2. 4 Tabrakan dan bukan tabrakan dengan sebuah lingkaran

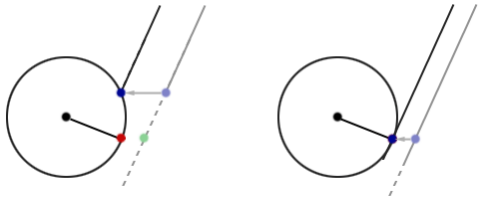
Gambar 2.4 bagian kiri menunjukkan kalau kita menggunakan titik potong lingkaran untuk menentukan perpotongan dengan garis kita akan mendapatkan titik yang terletak pada garis polygon.

Gambar 2.4 bagian kanan menunjukkan kalau perpotongan dengan garis *polygon* menghasilkan sebuah titik pada *polygon*, sehingga tidak perlu menentukkan titik terdekat dengan parimeter *polygon*. Pada kasus ini, titik potong garis dan titik potong pada *polygon* adalah sama.



Gambar 2. 5 Menggunakan titik potong polygon untuk bertabrakan dengan lingkaran

Gambar 2.5 menunjukkan kalau kita mengambil garis yang berlawanan arah dengan *vector* kecepatan, kita akan mengambil titik tubrukan dengan lingkaran, kita sebut dengan perpotongan berlawanan. Kita akan membutuhkan titik potong *polygon.*



Gambar 2. 6 Perpotongan berlawanan arah dengan lingkaran

Gambar 2.6 menunjukkan di mana titik potong *polygon* digunakan untuk perpotongan berlawanan untuk menempatkan titik di mana sebuah tubrukan dengan lingkaran terjadi. Berikut ini adalah *pseudo code* dari perpotongan lingkaran (inputan adalah asal cahaya dan *vector* arah normal) :

```
double intersectSphere(Point rO, Vector rV, Point sO, double sR)
{
   Vector Q = sO - rO;
   double c = length of Q;
   double v = Q * rV;
   double d = sR*sR - (c*c – v*v);
   if (d < 0.0) return -1.0;
   return v - sqrt(d);
}
```

2.6.7 Algoritma yang digunakan

AI yang yang digunakan pada *game* ini menggunakan metode *Greedy Best First Search*. Metode atau algoritma *Greedy* merupakan algoritma yang membentuk solusi langkah per langkah. Pada setiap langkah tersebut akan dipilih keputusan yang paling optimal. Keputusan tersebut tidak perlu memperhatikan keputusan selanjutnya yang akan diambil, dan keputusan tersebut tidak dapat diubah lagi pada langkah selanjutnya.[2]

a. Prinsip Utama Algoritma Greedy

Prinsip utama algoritma greedy adalah "*take what you can get now!?"*. Maksud dari prinsip tersebut adalah sebagai berikut. Pada setiap langkah dalam algoritma *greedy*, kita ambil keputusan yang paling optimal untuk langkah tersebut tanpa memperhatikan konsekuensi pada langkah selanjutnya. Kita namakan solusi tersebut dengan optimum lokal. Kemudian saat pengambilan nilai optimum lokal pada setiap langkah, diharapkan tercapai optimum global, yaitu tercapainya solusi optimum yang melibatkan keseluruhan langkah dari awal sampai akhir.

b. Elemen Algoritma *Greedy*

Elemen-elemen yang digunakan dalam penerapan algoritma *greedy* antara lain:

1. Himpunan Kandidat
2. Himpunan Solusi
3. Fungsi Seleksi
4. Fungsi Kelayakan
5. Fungsi Solusi
6. Fungsi Objektif

c. Skema Umum Algoritma *Greedy*

Misal kita mengasumsikan elemen algoritma *greedy* sebagai berikut:

Himpunan Kandidat = C,

Himpunan Solusi = S,

Fungsi Seleksi = select(),

Fungsi Kelayakan = feasible(),

Fungsi Solusi = solution(), dan

Fungsi Obyektif = objective().

Skema umum dari algoritma *greedy* dapat kita tuliskan:

1. Inisialisasi S dengan kosong.
2. Pilih sebuah kandidat dari C (dengan select()).
3. Kurangi C dengan kandidat yang telah terpilih di atas.

4. Periksa apakah kandidat yang dipilih tersebut bersama, sama dengan S membentuk solusi yang layak (dengan feasible()). Jika ya, masukkan kandidat ke S; jika tidak buang kandidat tersebut dan tidak perlu ditelaah lagi.
5. Periksa apakah S yang sudah terbentuk telah memberikan solusi yang lengkap (dengan solution()). Jika ya, berhenti; jika tidak, ulangi dari langkah 2.

Berikut contoh penerapan algoritma *Greedy* dalam masalah penukaran uang : [16]

Persoalan : Diberikan uang senilai *A*. Tukar *A* dengan koin-koin uang yang ada. Berapa jumlah minimum koin yang diperlukan untuk penukaran tersebut ?

Contoh: tersedia koin-koin 1, 5, 10, dan 25

Uang senilai 32 dapat ditukar dengan cara berikut:

32 = 1 + 1 + … + 1 (32 koin)

32 = 5 + 5 + 5 + 5 + 10 + 1 + 1 (7 koin)

32 = 10 + 10 + 10 + 1 + 1 (5 koin)

… dan seterusnya

Minimum: 32 = 25 + 5 + 1 + 1 (hanya 4 koin)

Elemen-elemen algoritma *greedy*-nya adalah:

1. Himpunan kandidat: himpunan koin yang merepresentasikan nilai 1, 5, 10, 25, paling sedikit mengandung satu koin untuk setiap nilai.
2. Himpunan solusi: total nilai koin yang dipilih tepat sama jumlahnya dengan nilai uang yang ditukarkan.

3. Fungsi seleksi: pilihlah koin yang bernilai tertinggi dari himpunan kandidat yang tersisa.
4. Fungsi layak: memeriksa apakah nilai total dari himpunan koin yang dipilih tidak melebihi jumlah uang yang harus dibayar.
5. Fungsi obyektif: jumlah koin yang digunakan minimum.

Strategi *greedy* yang digunakan adalah:

Pada setiap langkah, pilihlah koin dengan nilai sebesar mungkin dari himpunan koin yang tersisa dengan syarat (kendala) tidak melebihi nilai uang yang ditukarkan.

Tinjau masalah menukarkan uang 32 dengan koin 1, 5, 10, dan 25:

Langkah 1: pilih 1 buah koin 25 (Total = 25)

Langkah 2: pilih 1 buah koin 5 (Total = 25 + 5 = 30)

Langkah 3: pilih 2 buah koin 1 (Total = 25+5+1+1= 32)

Solusi: Jumlah koin minimum = 4 (solusi optimal!)

Pada setiap langkah di atas kita memperoleh optimum lokal, dan pada akhir algoritma kita memperoleh optimum global (yang pada contoh ini merupakan solusi optimum).

2.7 *Tools* yang digunakan

2.7.1 Adobe Flash

Flash adalah aplikasi yang sangat kaya sehingga menarik bagi *designer* maupun *developer* multimedia. Para *designer* sangat menyukai aplikasi ini karena kemudahannya dalam menghasilkan animasi, sedangkan para *developer* menikmati perlengkapannya untuk membuat aplikasi-aplikasi hebat yang ada pada *Flash. Flash* merupakan standar professional untuk menghasilkan *web experience* yang bercita-rasa tinggi [6].

2.7.1.1 Konsep Dasar *Flash*

Dalam membuat sebuah aplikasi menggunakan *Macromedia Flash,* terdapat beberapa konsep dasar yang perlu untuk dipahami terlebih dahulu. Konsep dasar tersebut meliputi animasi, *movie*, objek, teks, sound dan simbol.

1. Objek

Flash menyediakan tool untuk membuat objek sederhana seperti garis, lingkaran, persegi empat.

2. Teks

Pada *toolbox* disediakan fasilitas untuk menulis teks. *Flash* teks dikategorikan dalam tiga jenis yaitu teks statis label, teks dinamis, dan teks input.

3. Simbol

Dalam *Macromedia Flash* ada beberapa simbol yang harus diketahui yaitu *movie clip, button,* dan *graphic.* Masing-masing simbol mempunyai fungsi tersendiri.

4. Sound

Format *sound* yang dapat pergunakan dalam *Flash* dapat bermacam macam seperti *WAV, MP3*. *Sound* dapat mengimpor dari luar tetapi untuk *sound-sound* tertentu telah disediakan di dalam program *Flash.*

5. Animasi

Animasi adalah proses penciptaan efek gerak atau efek perubahan bentuk yang terjadi selama beberapa waktu. Animasi bisa berupa gerak sebuah objek dari tempat yang satu ke tempat yang lain, perubahaan warna, atau perubahan bentuk.

6. *Movie*

Dalam membuat animasi, maka seseorang akan mengatur jalan cerita dari animasi tersebut, membuat beberapa objek dan merangkainya menjadi suatu bagian yang bermakna tertentu. Suatu *movie* terkadang terdiri dari beberapa animasi yang biasa disebut *movie clip.* Bahasa yang digunakan dalam perancangan aplikasi ini adalah *Action Script 2.0.[6]*

2.7.2 UML (Unified Modeling Language)

UML (*Unified Modeling Language*) adalah sebuah bahasa untuk menetukan, visualisasi, kontruksi, dan mendokumentasikan *artifact* (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. *Artifact* dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya.[15]

UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan system yang besar dan kompleks. UML tidak hanya

digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

2.7.2.1 Bagian – bagian UML

Bagian-bagian utama dari UML adalah *view*, *diagram*, *model element,* dan *general mechanism*.

a. *View*

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. View bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah diagram.

Beberapa jenis view dalam UML antara lain: *use case view*, *logical view, component view*, *concurrency view,* dan *deployment view*.

b. *Use case view*

Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan *external actors*. *Actor* yang berinteraksi dengan sistem dapat berupa *user* atau sistem lainnya.

View ini digambarkan dalam *use case* diagram dan kadang-kadang dengan *activity diagrams*. *View* ini digunakan terutama untuk pelanggan, perancang (*designer*), pengembang (*developer*), dan penguji sistem (*tester*).

c. *Logical view*

Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (*class, object,*dan *relationship*) dan kolaborasi dinamis yang terjadi ketika object mengirim pesan ke object lain dalam suatu fungsi tertentu.

View ini digambarkan dalam *class diagrams* untuk struktur statis dan dalam *state, sequence, collaboration*, dan *activity diagram* untuk model dinamisnya. *View* ini digunakan untuk perancang (*designer*) dan pengembang (*developer*).

d. *Component view*

Mendeskripsikan implementasi dan ketergantungan modul. Komponen yang merupakan tipe lainnya dari *code module* diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumber daya komponen dan informasi administrasi lainnya. *View* ini digambarkan dalam *component view* dan digunakan untuk pengembang (*developer*).

e. *Concurrency view*

Membagi sistem ke dalam proses dan prosesor. *View* ini digambarkan dalam diagram dinamis (*state, sequence, collaboration,* dan *activity diagrams*) dan diagram implementasi (*component* dan *deployment diagrams*) serta digunakan untuk pengembang (*developer*), pengintegrasi (*integrator*), dan penguji (*tester*).

f. *Deployment view*

Mendeskripsikan fisik dari sistem seperti komputer dan perangkat (*nodes*) dan bagaimana hubungannya dengan lainnya. *View* ini digambarkan dalam *deployment diagrams* dan digunakan untuk pengembang (*developer*), pengintegrasi (*integrator*), dan penguji (*tester*).

g. *Diagram*

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah

diagram merupakan bagian dari suatu view tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu. Adapun jenis diagram antara lain :[15]

1. *Use Case Diagram*

Use case adalah abstraksi dari interaksi antara system dan actor. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana system akan terlihat di mata *user*. Sedangkan *use case diagram* memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan client.

2. ***Class Diagram***

Class adalah dekripsi kelompok obyek-obyek dengan property, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya *class diagram* dapat memberikan pandangan global atas sebuah sistem. Hal tersebut tercermin dari *class- class* yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa *class* diagram. *Class* diagram sangat membantu dalam visualisasi struktur kelas dari suatu sistem.

3. *Component Diagram*

Component software merupakan bagian fisik dari sebuah system, karena menetap di komputer tidak berada di benak para analis. Komponen merupakan implementasi *software* dari sebuah atau lebih *class*. Komponen dapat berupa *source code*, komponent biner, atau *executable component*.

Sebuah komponent berisi informasi tentang *logic class* atau *class* yang diimplementasikan sehingga membuat pemetaan dari *logical view* ke *component view*. Sehingga component diagram merepresentasikan dunia riil yaitu *component software* yang mengandung *component, interface* dan *relationship.*

4. *Deployment Diagram*

Menggambarkan tata letak sebuah system secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware*, menunjukkan hubungan komputer dengan perangkat (nodes) satu sama lain dan jenis hubungannya. Di dalam *nodes, executeable component* dan *object* yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh *node* tertentu dan ketergantungan komponen.

5. *State Diagram*

Menggambarkan semua state (kondisi) yang dimiliki oleh suatu object dari suatu class dan keadaan yang menyebabkan state berubah. Kejadian dapat berupa object lain yang mengirim pesan. State class tidak digambarkan untuk semua class, hanya yang mempunyai sejumlah state yang terdefinisi dengan baik dan kondisi class berubah oleh stateyang berbeda.

6. *Sequence Diagram*

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antaraobject, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

7. *Collaboration Diagram*

Menggambarkan kolaborasi dinamis sepertisequence diagrams. Dalam menunjukkan pertukaran pesan, *collaboration diagrams* menggambarkan *object* dan hubungannya (mengacu ke konteks). Jika penekannya pada waktu atau urutan gunakan *sequence diagrams*, tapi jika penekanannya pada konteks gunakan *collaboration diagram*.

8. *Activity Diagram*

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

2.7.2.2 Tujuan Penggunaan UML

1. Memberikan bahasa pemodelan yang bebas dari berbagai bahas pemrograman dan proses rekayasa.
2. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.
3. Memberikan model yang siap pakai, bahsa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.

4. UML bisa juga berfungsi sebagai sebuah (*blue print*) cetak biru karena sangat lengkap dan detail. Dengan cetak biru ini maka akan bias diketahui informasi secara detail tentang coding program atau bahkan membaca program dan menginterpretasikan kembali ke dalam bentuk diagram (*reserve enginering*).