

BAB 2

LANDASAN TEORI

2.1 Inteligencia Semu

2.1.1 Pengertian Inteligencia Semu

Menurut Rich dan Knight (1991, p3), inteligensia semu atau yang sering dikenal dengan kecerdasan buatan adalah ilmu yang mempelajari bagaimana membuat komputer melakukan sesuatu sebaik yang dapat dilakukan manusia.

Menurut Kusumadewi (2003, p1), inteligensia semu merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan manusia. Pengertian inteligensia semu dapat juga dipandang dari berbagai sudut pandang, diantaranya adalah (Kusumadewi, 2003, p2):

a. Sudut pandang kecerdasan

Inteligensia semu akan membuat mesin menjadi 'cerdas' dalam arti komputer mampu berbuat seperti apa yang dilakukan manusia.

b. Sudut pandang penelitian

Inteligensia semu adalah suatu studi bagaimana membuat agar komputer dapat melakukan sesuatu sebaik yang dikerjakan oleh manusia.

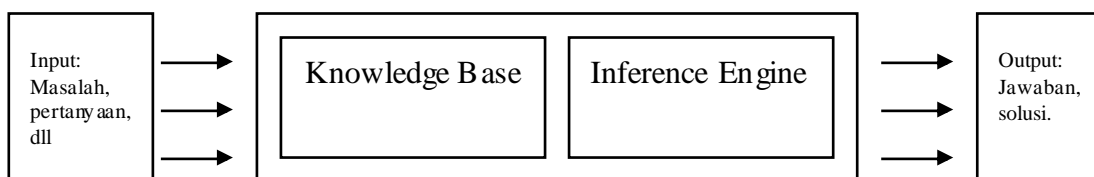
Domain yang sering dibahas oleh para peneliti meliputi:

1) *Mundane task*

- a) Persepsi (*vision* dan *speech*)
- b) Bahasa alami (*understanding*, *generation*, dan *translation*)
- c) Pemikiran yang bersifat *common sense*
- d) *Robot control*

2) *Formal task*

- a) Permainan atau *games*
 - b) Matematika (geometri, logika, kalkulus integral, dan pembuktian)
- 3) *Expert task*
- a) Analisis finansial
 - b) Analisis medis
 - c) Analisis ilmu pengetahuan
 - d) Rekayasa (desain, pencarian kegagalan, perencanaan manufaktur)
- c. Sudut pandang bisnis
- Inteligensia semu adalah kumpulan peralatan yang sangat *powerful* dan metodologis dalam menyelesaikan masalah-masalah bisnis.
- d. Sudut pandang pemrograman
- Inteligensia semu meliputi studi tentang pemrograman simbolik, penyelesaian masalah (*problem solving*) dan pencarian (*searching*).
- Untuk melakukan aplikasi kecerdasan buatan terdapat dua bagian penting yang sangat dibutuhkan, yaitu:
- 1) *Knowledge base* yang berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya
 - 2) *Inference engine*, yaitu kemampuan menarik kesimpulan berdasarkan pengalaman, seperti yang terlihat pada Gambar 2.1.



Gambar 2.1 *Inference Engine*

Ruang lingkup dalam inteligensia semu antara lain :

- a. Sistem Pakar (*Expert System*)
- b. *Natural Language Processing*
- c. *Speech Recognition*
- d. *Robotics and Sensory Systems*
- e. *Computer Vision*
- f. *Intelligent Computer-aided Instruction*
- g. *Game Playing*

2.1.2 Teknik-Teknik dalam Inteligensia Semu

Menurut Rich dan Knight (1991, p8), teknik-teknik inteligensia semu harus dideklarsikan dengan cara:

- a. Pengetahuan yang dapat digeneralisasi
- b. Inteligensia semu harus dapat dimengerti oleh orang yang menggunakannya.
- c. Inteligensia semu mudah untuk diperbaiki dan mudah beradaptasi dengan pengguna.
- d. Inteligensia semu harus dapat digunakan dalam berbagai situasi, walaupun data yang dimiliki tidak lengkap.
- e. Inteligensia semu selalu dapat menyadari seberapa besar kemungkinan sesuatu akan terjadi.

Meskipun teknik inteligensia semu dirancang untuk memecahkan beberapa masalah dalam inteligensia semu, tetapi terdapat jarak antara masalah dan teknik inteligensia semu. Jadi masalah dalam inteligensia semu dapat dipecahkan tanpa harus menggunakan teknik inteligensia semu, dan masalah non-inteligensia semu dapat dipecahkan menggunakan teknik inteligensia semu.

Teknik inteligensia semu memiliki tiga bagian penting, diantaranya adalah:

a. *Search*

Mencari bagaimana cara pemecahan masalah yang ada dengan menggunakan berbagai macam teknik intelegensia yang ada.

b. *Use of Knowledge*

Mencari bagaimana cara pemecahan masalah yang kompleks dengan menyelusuri struktur dari objek yang bersangkutan.

c. *Abstraction*

Mencari bagaimana cara pemecahan masalah dengan menggunakan hal-hal yang tidak penting dari suatu masalah.

2.1.3 Sejarah Inteligencia Semu

Kecerdasan buatan termasuk bidang ilmu yang relatif muda. Pada tahun 1950-an para ilmuwan dan peneliti mulai memikirkan bagaimana caranya agar mesin dapat melakukan pekerjaannya seperti yang dapat dikerjakan oleh manusia. Alan Turing, seorang matematikawan dari Inggris, pertama kali mengusulkan adanya tes untuk melihat mampu tidaknya sebuah mesin dikatakan cerdas. Hasil dari tes tersebut membuktikan bahwa mesin tersebut cerdas karena dapat berkomunikasi dengan orang lain layaknya manusia.

Kecerdasan buatan atau “*artificial intelligence*” atau inteligensia semu itu sendiri dimunculkan oleh seorang profesor dari *Massachusetts Institute of Technology* yang bernama John McCarthy pada tahun 1956 pada Dartmouth Conference yang dihadiri oleh para peneliti AI. Pada konferensi tersebut juga didefinisikan tujuan utama dari kecerdasan buatan, yaitu mengetahui dan memodelkan proses-proses berpikir manusia dan mendesain mesin agar dapat menirukan kelakuan manusia tersebut. Berikut ini adalah beberapa program AI yang mulai dibuat pada tahun 1956-1966, antara lain:

- a. Logic Theorist, diperkenalkan pada Dartmouth Conference. Program ini dapat membuktikan teorema-teorema matematika.
- b. Sad Sam, dibuat oleh Robert K. Lindsay (1960). Program ini dapat mengetahui kalimat-kalimat sederhana yang ditulis dalam bahasa Inggris dan mampu memberikan jawaban dari fakta-fakta yang didengar dalam sebuah percakapan.
- c. ELIZA, dibuat oleh Joseph Weizenbaum (1967). Program ini mampu melakukan terapi terhadap pasien dengan memberikan beberapa pertanyaan.

2.1.4 Inteligencia Semu Banding Inteligencia Alami

Jika dibandingkan dengan inteligensia alami (kecerdasan yang dimiliki manusia), inteligensia semu memiliki beberapa keuntungan secara komersial, di antaranya adalah:

- a. Inteligencia semu lebih bersifat permanen sementara inteligensia alami cepat mengalami perubahan. Hal ini dimungkinkan karena sifat manusia yang pelupa. Inteligencia buatan tidak akan berubah sepanjang sistem komputer dan program tidak mengubahnya.
- b. Inteligencia semu lebih mudah digandakan dan disebar. Memindahkan pengetahuan manusia dari satu orang ke orang lain membutuhkan proses yang sangat lama dan juga suatu keahlian tidak akan dapat digandakan dengan sempurna. Oleh karena itu, jika pengetahuan terletak pada suatu sistem komputer, pengetahuan tersebut dapat disalin dan dipindahkan dengan mudah dari suatu komputer ke komputer lain.
- c. Inteligencia semu lebih murah dibandingkan dengan inteligensia alami. Menyediakan layanan komputer akan lebih mudah dan lebih murah dibandingkan dengan harus mendatangkan seseorang untuk mengerjakan sejumlah pekerjaan dalam jangka waktu yang sangat lama.

- d. Inteligencia semu bersifat konsisten. Hal ini disebabkan karena inteligencia semu adalah bagian dari teknologi komputer sedangkan inteligencia alami akan senantiasa berubah-ubah.
- e. Inteligencia semu dapat didokumentasi. Keputusan yang dibuat oleh komputer dapat didokumentasikan dengan mudah dengan cara melacak setiap aktivitas dari sistem tersebut. Inteligencia alami sangat sulit untuk direproduksi.
- f. Inteligencia semu dapat mengerjakan pekerjaan lebih cepat dibandingkan dengan inteligencia alami.
- g. Inteligencia semu dapat mengerjakan pekerjaan lebih baik dibandingkan dengan inteligencia alami.

Sedangkan keuntungan dari inteligencia alami adalah:

- a. Kreatif. Kemampuan untuk menambahkan atau memenuhi pengetahuan itu sangat melekat pada jiwa manusia. Pada kecerdasan buatan, untuk menambahkan pengetahuan harus dilakukan melalui sistem yang dibangun.
- b. Kecerdasan alami memungkinkan orang untuk menggunakan pengalaman secara langsung sedangkan pada kecerdasan buatan harus bekerja dengan input-input simbolik.
- c. Pemikiran manusia dapat digunakan secara luas, sedangkan kecerdasan buatan sangat terbatas.

2.1.5 Sistem Pakar

2.1.5.1 Pengertian Sistem Pakar

Menurut Turban dan Aronson (2000, p408), sistem pakar adalah salah satu alat yang digunakan untuk meningkatkan produktivitas dan kualitas dari suatu barang. Sistem pakar sudah banyak digunakan oleh perusahaan-perusahaan besar maupun perusahaan tingkat

menengah. Sistem pakar juga merupakan alat penting dalam membantu membuat keputusan bisnis dan pengembangan proses bisnis.

2.1.5.2 Kategori-kategori dari Sistem Pakar

Beberapa kategori dari sistem pakar:

a. *Interpretation systems*

Interpretation systems menerjemahkan situasi deskripsi dari observasi. Kategori ini terdiri dari *surveillance*, *speech understanding*, *image analysis*, *signal interpretation*, dan beberapa *intelligence analyses*.

b. *Prediction systems*

Prediction systems menerjemahkan konsekuensi dari situasi yang ada. Kategori ini terdiri dari ramalan cuaca, *demographic predictions*, *economic forecasting*, *traffic predictions*, *crop estimates*, militer, perdagangan, dan *financial forecasting*.

c. *Diagnosis systems*

Diagnosis systems menerjemahkan *malfunctions systems* pada saat observasi. Kategori ini terdiri dari ilmu kedokteran, elektronik, mekanik, dan *software diagnoses*.

d. *Design systems*

Design systems mengatur objek yang sedang mengalami masalah. Kategori ini terdiri dari *circuit layout*, desain bangunan, dan *plant layout*.

e. *Planning systems*

Planning systems melakukan pengembangan rencana yang sudah ada untuk mencapai target. *Planning systems* umumnya berhubungan dengan masalah perencanaan seperti *automatic programming*. *Planning systems* juga berhubungan dengan rencana jangka panjang dan pendek seperti *project management*, *routing*, komunikasi, pengembangan produk, aplikasi militer, dan perencanaan bisnis.

f. *Monitoring systems*

Monitoring systems melakukan perbandingan antara rencana dan hasil yang sudah dicapai. Kategori ini terdiri dari *air traffic control* sampai *fiscal management tasks*.

g. *Debugging systems*

Debugging systems memberikan solusi pada masalah yang sedang dihadapi. Kategori ini terdiri dari *planning*, *design*, dan *prediction capabilities* untuk menciptakan spesifikasi dan rekomendasi untuk memperbaiki masalah diagnosis.

h. *Repair systems*

Repair systems melaksanakan rencana untuk mengelola solusi sebuah masalah. Kategori ini terdiri dari *systems incorporate debugging*, perencanaan, dan *execution capabilities*.

i. *Instruction systems*

Instruction systems melaksanakan diagnosis, *debugging*, dan memperbaiki kinerja murid. Pada kategori ini, pertama-pertama *subsystems* mendeskripsikan apa saja pengetahuan yang dimiliki oleh murid, menganalisis apa saja kekurangan yang ada pada murid, dan menciptakan tutorial yang dapat membantu murid.

j. *Control systems*

Control systems melaksanakan *interpreting*, memprediksi, memperbaiki, dan memantau sifat sistem. Untuk melakukan hal tersebut, *control systems* harus mengatur situasi yang ada, memprediksikan masa depan, mengantisipasi masalah yang ada, merencanakan perbaikan masalah, dan memantau eksekusi agar berhasil.

2.1.5.3 Keuntungan-keuntungan dari Sistem Pakar

Keuntungan-keuntungan dari sistem pakar adalah sebagai berikut:

a. Meningkatkan *output* dan produktivitas

Sistem pakar dapat bekerja lebih cepat dari pada manusia, oleh karena itu sistem pakar dapat meningkatkan *output* dan produktivitas.

b. Mengurangi waktu pengambilan keputusan

Dengan menggunakan sistem pakar, manusia dapat mengambil keputusan dengan lebih cepat.

c. Meningkatkan proses dan kualitas produk

Dengan menggunakan sistem pakar, kualitas produk dapat ditingkatkan karena nasihat yang konsisten dan mengurangi jumlah kesalahan yang ada.

d. Mengurangi *downtime*

Pada penggunaan sistem pakar, waktu kerja dapat dikurangi karena sistem pakar dapat melakukan diagnosis masalah dan memperbaiki masalah.

e. Menyimpan ilmu pengetahuan yang sudah langka

Sistem pakar dapat menggunakan menyimpan ilmu pengetahuan yang dimiliki oleh seorang pakar yang akan pensiun atau meninggalkan pekerjaan.

f. Fleksibilitas

Sistem pakar dapat memberikan fleksibilitas dalam *service* dan *manufacturing* industri.

2.1.5.4 Perangkat Ajar atau CAI (Computer-Assisted Instruction)

2.1.5.4.1 Pengertian Perangkat Ajar

Menurut Anonimus (2008a dan 2008b), perangkat ajar atau yang sering disebut CAI (*Computer-Aided Instruction*) adalah suatu sistem komputerisasi yang digunakan untuk kegiatan belajar mengajar (2008a, p1). Perangkat ajar adalah salah satu jenis pendidikan yang dijalankan oleh siswa dan dilakukan dimana saja mereka berada dengan menggunakan program pelatihan pada komputer. Perangkat ajar sangat berguna untuk membantu siswa dalam mempelajari aplikasi komputer karena program perangkat ajar dapat berintegrasi

dengan aplikasi komputer lainnya sehingga siswa dapat mempelajari program perangkat ajar seperti mempelajari aplikasi komputer lainnya.

Menurut Kearsley (1983), perangkat ajar adalah semua jenis pelatihan atau pembelajaran yang menggunakan komputer dalam pelatihannya. CAI (*Computer-Aided Instruction*) sering disebut juga CBT (*Computer-Based Training*) atau CAL (*Computer-Assisted Learning*) atau CAT (*Computer-Assisted Testing*).

Menurut Chamber (1983, p5), perangkat ajar adalah suatu fungsi dari komputer yang menyediakan instruksi dalam bentuk latihan, tutorial, dan simulasi. Di Amerika Serikat, CAI (*Computer-Aided Instruction*) atau CAI (*Computer-Assisted Instruction*) atau CBI (*Computer-Based Instruction*) atau CBE (*Computer-Based Education*), sedangkan di Eropa seperti Inggris lebih dikenal dengan nama CAL (*Computer-Assisted Learning*) atau CBT (*Computer-Based Training*).

2.1.5.4.2 Tujuan Perangkat Ajar

Menurut Kearsley (1983, p2), ada delapan tujuan digunakannya CAI (*Computer-Aided Instruction*) dalam bidang pendidikan, yaitu:

a. Biaya yang lebih efektif

Perangkat ajar yang dapat digunakan di rumah atau dimana pun yang memungkinkan untuk mengurangi atau menghilangkan biaya perjalanan pelatihan atau kursus.

b. Bekerja dengan orang banyak terutama orang dewasa

Dengan menggunakan perangkat ajar, jumlah pengguna yang melakukan pelatihan atau kursus dapat ditingkatkan karena banyak pengguna yang melakukan pelatihan atau kursus tidak merasa dikontrol. Selain itu, perangkat ajar dapat melibatkan peserta secara aktif dalam proses pelatihan.

c. Fleksibel dan individualisasi

Perangkat ajar dapat digunakan sesuai dengan keinginan pengguna sehingga pengguna dapat melakukan pelatihan dengan perasaan puas.

d. Perangkat ajar tidak memiliki jadwal tertentu

Perangkat ajar yang tidak memiliki jadwal tertentu sangat berguna untuk pengguna yang memiliki waktu terbatas karena perangkat ajar dapat digunakan pada waktu yang diinginkan pengguna. Selain itu, perangkat ajar dapat melatih lebih banyak pengguna pada waktu yang bersamaan. Hal itu dapat terjadi karena perangkat ajar tidak menggunakan ruangan. Dengan kata lain perangkat ajar dapat digunakan dimana saja, tidak seperti organisasi-organisasi pelatihan yang memerlukan ruangan dalam pengajarannya.

e. Meningkatkan kontrol atas kegiatan pelatihan

Perangkat ajar umumnya dilengkapi dengan sistem yang dapat melacak kemajuan kemampuan dari pengguna. Sistem tersebut umumnya mencatat setiap hasil yang telah dilakukan oleh pengguna sehingga pengguna dapat melihat kemajuan yang mereka lakukan setelah menggunakan perangkat ajar tersebut.

f. Mengurangi penggunaan sumber daya

Dengan menggunakan perangkat ajar, sumber daya yang ingin digunakan dapat dikurangi. Karena dengan menggunakan perangkat ajar, fasilitas-fasilitas organisasi pelatihan dapat berkurang. Organisasi pelatihan dapat menggunakan perangkat ajar dalam pembelajarannya sehingga fasilitas-fasilitas tertentu tidak perlu digunakan.

g. Meningkatkan kinerja pekerjaan

Perangkat ajar dapat meningkatkan kinerja pekerjaan karena perangkat ajar mengajarkan materi khusus untuk spesifik pekerjaan, sehingga pekerja dapat menguasai pekerjaan dengan lebih baik. Selain itu, perangkat ajar dapat mengajarkan materi umum

yang berguna untuk kehidupan sehari-hari dan dapat meningkatkan pengguna dalam mengambil keputusan.

h. Perangkat ajar dapat menggantikan organisasi-organisasi pelatihan

Cara belajar menggunakan perangkat ajar yang fleksibel menyebabkan perubahan cara belajar yang pada umumnya dilakukan di lembaga-lembaga pengajaran dapat dilakukan di rumah secara individu.

2.1.5.4.3 Jenis-Jenis Perangkat Ajar

Menurut Kearsley (1983, p30) terdapat enam bentuk CAI (*Computer-Aided Instruction*), yaitu:

a. Tutorial

Tutorial merupakan salah satu perangkat ajar yang paling umum. Tutorial umumnya digunakan untuk memberikan informasi secara berurutan. Hal ini berguna untuk mengajarkan informasi faktual, aturan, dan aplikasi sederhana dari aturan. Kunci tutorial yang berguna adalah terjadinya interaksi bolak-balik, isi yang jelas, menyediakan sarana untuk latihan dan dapat dipercaya.

b. *Drill and Practice*

Drill and Practice merupakan salah satu perangkat ajar lainnya yang memberikan praktek kepada penggunanya dan memberikan *feedback* yang cepat dan tepat.

c. *Training games*

Training games merupakan salah satu perangkat ajar yang memberikan dorongan motivasi dan kesempatan kepada pengguna untuk berlatih setelah mempelajari informasi baru. *Training games* menambahkan nilai hiburan dan rasa ketertarikan kepada pengguna.

d. Simulasi

Simulasi merupakan salah satu perangkat ajar yang membuat situasi atau keadaan seperti tempat kerja sebenarnya. Selain itu, simulasi dapat mengurangi biaya dan bahaya dari keadaan sebenarnya.

e. *Problem Solving*

Problem Solving merupakan salah satu perangkat ajar yang paling menantang dalam CAI (*Computer-Aided Instruction*). Perangkat ajar ini membantu pengguna mengembangkan ketrampilan logika, memecahkan masalah, dan memberikan pengarahan. Umumnya perangkat ajar ini meningkatkan keahlian berpikir dari pengguna.

f. Demonstrasi atau presentasi

Demonstrasi adalah perangkat ajar yang sangat baik digunakan untuk mendukung pemberian informasi baru. Perangkat ajar ini juga digunakan untuk alat *review*.

2.1.5.4.4 Konsep-Konsep Perangkat Ajar

Menurut Kearsley, proses pembuatan atau pengembangan perangkat ajar atau CAI (*Computer-Aided Instruction*) menggunakan empat konsep dasar, yaitu:

a. Perangkat keras (*Hardware*)

Perangkat keras adalah perangkat fisik yang menjadi penghubung antara komputer dan pengguna, seperti: CPU, monitor, printer, dan lain-lain.

b. Perangkat lunak (*Software*)

Perangkat lunak adalah kumpulan program dalam sistem yang mengoperasikan dan melakukan semua fungsi-fungsi instruksional, seperti: sistem operasi, program-program aplikasi, dan lain-lain.

c. Perangkat ajar (*Courseware*)

Perangkat ajar adalah perangkat lunak yang disertai aturan-aturan khusus yang melengkapi presentasi instruksional.

d. Manusia (*Brainware*)

Manusia yang dimaksud adalah orang-orang yang memiliki keahlian khusus untuk merancang, memelihara, dan mengevaluasi sistem perangkat ajar.

2.1.5.4.5 Fitur-Fitur yang Diberikan Perangkat Ajar

Menurut Kearsley (1983), perangkat ajar memberikan beberapa fitur-fitur diantaranya yaitu:

a. *Feedback* yang cepat

Feedback yang cepat berguna untuk membantu pengguna untuk mengetahui apa yang harus mereka ketahui. Selain itu, *feedback* berguna untuk memantau perkembangan pengguna. *Feedback* dibagi menjadi tiga jenis, di antaranya, yaitu:

- 1) *Right – Wrong Feedback*, yang memberikan respon baik pengguna menjawab benar maupun salah.
- 2) *Right – Blank Feedback*, yang hanya memberikan umpan balik bila pengguna menjawab benar dan tidak memberikan respon jika pengguna salah menjawab.
- 3) *Wrong – Blank Feedback*, yang memberikan umpan balik pada jawaban yang salah saja.

b. Teknologi yang berkembang pesat dewasa ini menciptakan perangkat ajar yang dapat bekerja secara online sehingga perangkat ajar dapat digunakan oleh setiap individu tanpa mengenal tempat dan waktu.

- c. Perangkat ajar dapat berintergrasi dengan berbagai komponen

Komponen-komponen yang dapat berintergrasi dengan perangkat ajar membuat pengguna lebih mudah mempelajari materi yang diberikan. Hal ini dikarenakan manusia dapat mengingat 20% dari apa yang mereka lihat, 40% dari apa yang mereka dengar dan lihat, dan 70% dari apa yang mereka lihat, dengar dan lakukan. Komponen-komponen tersebut adalah *text*, grafik, *video*, dan suara.

2.1.5.5 Perangkat Ajar Cerdas atau ICAI (Intelligent Computer-Assisted Instruction)

Menurut Saloky dan Seminsky (2005), perangkat ajar cerdas atau ICAI (*Intelligent Computer-Aided Instruction*) adalah penambahan teknik AI (*Artificial Intelligence*) pada CAI (*Computer-Aided Instruction*) agar dapat berjalan lebih baik dan efisien. Tujuan dari ICAI adalah untuk mengembangkan intruksi material yang mengutamakan perkembangan kemampuan *user*. ICAI terdiri beberapa komponen, di antaranya yaitu:

- a. *Problem-solving expertise*

Komponen ICAI (*Intelligent Computer-Aided Instruction*) ini mengandung ilmu pengetahuan yang ingin diajarkan kepada *user* (*Knowledge based*). Teknik yang digunakan sama seperti teknik yang digunakan dalam sistem pakar.

- b. *Student model*

Student model adalah hal terpenting dalam ICAI (*Intelligent Computer-Aided Instruction*). Komponen ini memberikan analisis di bagian mana *user* mengalami kesulitan, sehingga ICAI (*Intelligent Computer-Aided Instruction*) dapat mengajarkan bagian yang sulit bagi *user*. Jadi komponen ini digunakan untuk mencari bagian yang sulit dipahami oleh *user*.

- c. *Tutoring module*

Tutoring module adalah komponen pada ICAI (*Intelligent Computer-Aided Instruction*) yang memilih bagaimana cara mengajarkan bagian yang sulit dimengerti oleh *user* pada komponen *student model*.

2.2 Game Engine

2.2.1 Pengertian Game Engine

Menurut Anonimus (2008c), *game engine* adalah sebuah sistem perangkat lunak yang dirancang untuk menciptakan dan mengembangkan sebuah permainan game. Fasilitas yang biasa diberikan *game engine* adalah perenderan grafik dalam bentuk 2D atau 3D. Selain itu *game engine* juga memberikan fasilitas lain, seperti: *a physics engine*, suara, *script*, animasi, *artificial intelligence* atau sering disebut juga otak buatan, *networking*, *streaming*, *memory management*, *threading*, dan *a scene graph*.

2.2.2 Jenis-Jenis Game Engine

Menurut Anonimus (2008c), *game engine* yang paling diminati orang pada tahun 1990-an adalah *First-Person Shooter* (FPS), seperti: *Doom*, *Counter Strike*, dan *Quake*. *Game engine* lainnya yang diminati orang, yaitu:

- a. C4 Engine
- b. Panda 3D
- c. Unity
- d. DX Studio
- e. Visual3D.NET Game Engine

Perkembangan pesat yang dilakukan *Frst-Person Shooter* dalam hal kualitas visualitas. Tidak hanya *first-person shooter* yang berkembang, jenis-jenis *game* lainnya juga

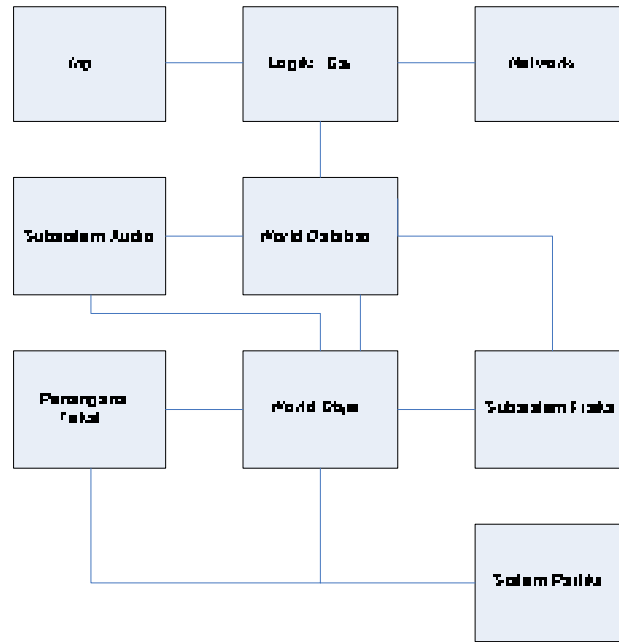
ikut berkembang dalam tingkat realitas, di antaranya: *flight*, *driving*, dan *Real-Time Strategy* (RTS).

2.2.3 Komponen-Komponen *Game Engine*

Menurut Hawkins dan Astle (2001, p708), *game engine* yang sederhana terdiri dari komponen-komponen berikut ini:

- a. Input
- b. Logika Game
- c. *Networking*
- d. Subsistem Audio
- e. *World Database*
- f. Penanganan Tekstur
- g. *World Object*
- h. Subsistem Fisika
- i. Sistem Partikel

Jika digambarkan dalam diagram *flowchart*, akan terlihat seperti Gambar 2.2.



Gambar 2.2 Komponen *Game Engine*

Dari Gambar 2.2, dapat dilihat bahwa pada dasarnya *game engine* menerima masukan dari sub sistem *networking* dan sub sistem input. Hasil dari input tersebut dikelola oleh sub sistem logika *game* agar dapat mengeksekusi *game cycle*. Dalam *game cycle*, sub sistem logika akan menanggapi input dan melakukan perhitungan fisika pada objek *game*, seperti: menangani deteksi dan respon tumbukan, *me-load* dan menghancurkan objek, menggerakkan kamera, dan memainkan musik atau suara yang dibutuhkan selama *game* berjalan.

2.2.4 Engine *RPG Maker VX*

Menurut Anonimus (2008e), *RPG Maker VX* adalah keluaran terbaru dari *RPG Maker Series* yang dikembangkan oleh *Enterbrain*. Sebelum *RPG Maker VX* dikeluarkan, terdapat beberapa *series* dari *RPG Maker* diantaranya: *RPG Maker 95*, *RPG Maker 2000*, *RPG Maker 2003*, dan *RPG Maker XP*. *RPG Maker XP* hanya dapat berjalan pada platform *Windows*. Fasilitas-fasilitas yang diberikan *RPG Maker VX* adalah:

- a. Kemudahan dalam memasukkan perintah *Enter Message*.
- b. *RPG Maker VX* dapat melakukan *generate random dungeon* dengan cepat.
- c. Sistem *battle RPG Maker VX* yang diperbarui dengan tampilan *front-view battle system*.
- d. Fasilitas *RGSS2* yang memudahkan pengguna untuk menulis dan mengubah *script*, dan mengubah kosakata yang terdapat dalam *game*.
- e. Fasilitas *Quick Event Creation* yang memudahkan pengguna untuk membuat *event* tanpa harus membuatnya di *script* di antaranya adalah: *door*, *inn commands*, dan *treasure chests*.

Terdapat beberapa *requirements* pada *RPG Maker VX*, yaitu:

- a. Minimum
 - Microsoft® Windows® 2000 / XP / Vista
 - 1.0GHz Intel® Pentium® III
 - 256MB kapasitas RAM
 - 1024x768 resolusi video
 - 100MB kapasitas *hard-disk* yang tersisa
- b. *Recommended*
 - Microsoft® Windows® 2000 / XP / Vista
 - 2.0GHz Intel® Pentium® 4 atau lebih tinggi
 - 512MB kapasitas RAM
 - 1280x1024 atau resolusi video yang lebih baik
 - 100MB kapasitas *hard-disk* yang tersisa

2.3 Game Design

2.3.1 Jenis-Jenis Game

Menurut Saltzman (2000, p 1), Jenis-jenis *game* terdiri dari:

a. *Action*

Action games lebih membutuhkan koordinasi antara mata dan tangan daripada cerita atau strategi. *Action games* umumnya berorientasi pada kecepatan dan refleksi. Tipe *action games* yang populer adalah tipe *games first-person perspective 3D*. Contoh dari tipe *games* ini adalah *Quake Series*, dan *Unreal Tournament* yang dikeluarkan *Epic Games*.

b. *Strategy*

Strategy games lebih menggunakan logika dan perencanaan. Jenis *game* ini umumnya lebih menggunakan manajemen sumber daya alam dan waktu daripada keterlibatan karakter. Taktik dalam pengorganisasian dan pergerakan sangat penting dalam jenis *game* ini. Umumnya pencipta jenis *game* ini membuat suatu masalah yang menyebabkan pengguna harus berpikir dalam pengambilan keputusan dan pemberian perintah yang tepat. Contoh dari tipe *games* ini adalah *Starcraft* yang dikeluarkan *Blizzard* dan *Age of Empires* yang dikeluarkan *Ensemble Studio*.

c. *Adventure*

Adventure games membuat pengguna menjelajahi perjalanan dan memecahkan beberapa masalah-masalah. Umumnya tipe *games* ini memiliki jalan cerita yang membuat pengguna menjadi karakter utama dan untuk mencapai tujuan akhir diperlukan interaksi pengguna dan peralatan-peralatan. Contoh dari tipe *games* ini adalah *Grim Fandango* yang dikeluarkan *LucasArt* dan *Myst or Riven* yang dikeluarkan *Broderbund*.

d. *Role-Playing Games (RPGs)*

Role-Playing games (RPGs) menyerupai jenis *adventure games*. Tetapi pada *Role-playing games*, *game* lebih bergantung pada perkembangan karakter dan strategi dalam bertarung daripada pemecahan masalah-masalah seperti pada *adventure games*. Contoh

dari jenis *game* ini adalah *Diablo* yang dikeluarkan *Blizzard*, *Final Fantasy VII* yang dikeluarkan *Squaresoft* dan *Dungeon Keeper* yang dikeluarkan *Bullfrog*.

e. *Sports*

Sports games adalah salah simulasi pemain tunggal atau tim dari sudut pandang pemain atau tim tersebut. *Reality* merupakan hal penting dalam jenis *game* ini, seperti *fast action* dan taktik strategi. Contoh dari jenis *game* ini adalah *NHL series* dari *Electronic Arts*, *WWF: Warzone* yang dikeluarkan *Acclaim*, *Microsoft Links LS 2000*, dan *NFL Blitz 2000* yang dikeluarkan *Midway*.

f. *Simulations* atau *Sims*

Sims memberikan simulasi animasi atau non-animasi dari objek dan proses. Umumnya jenis *game* ini mengambil *3D first-person prespective*. Contoh dari jenis *game* ini adalah *MiG Alley* yang dikeluarkan *Interplay* dan *Armored Fist* yang dikeluarkan *NovaLogic*.

g. *Puzzle* atau *Classic Games*

Puzzle atau *classic games* merupakan jenis *game* yang paling tua. Jenis-jenis *game* ini antara lain: permainan kartu, *tile games*, *trivia*, *word*, atau *broad games*. Contoh dari jenis *game* ini adalah catur, *checkers*, *backgammon*, mahyong, dan *solitaire*.

2.3.2 Sudut Pandang Dalam Game

Menurut Saltzman (2000,p1), terdapat beberapa sudut pandang dalam *game*, yaitu:

a. *First-Person Prespective*

First-Person Prespective adalah sudut pandang dalam *game* yang membuat seolah-olah pengguna melihat *game* tersebut dari mata mereka sendiri. Contohnya adalah *Duke Nukem 3D* yang dikeluarkan *3D Realms* dan *Turok: Dinoasaur Hunter* yang dikeluarkan *Acclaim*.

b. *Third-Person Prespective*

Third-Person Prespective adalah sudut pandang dalam *game* yang menampilkan layar dimana pengguna sedang bermain. Contohnya adalah *the Tomb Raider series*, *Mario 64*, *Legend of Zelda: Ocarina of Time*, *Crash Bandicoot: Warped*, *Metal Gear Solid*, dan *Resident Evil*.

c. *Top-Down Prespective*

Top-Down Prespective adalah sudut pandang dalam *game* yang melihat *game* dari atas ke bawah. Contohnya adalah *Starcraft* yang dikeluarkan *Blizzard*, *Civilization* yang dikeluarkan *Microprose*, *Steel Panthers* yang dikeluarkan *SSI*, dan *Pokemon Red, Blue, Yellow* (dalam *Color Game Boy*).

d. *Isometric*

Isometric adalah sudut pandang dalam *game* yang memberikan penglihatan *three quarter* sehingga akan menampilkan gambar *3D* yang luar biasa. Contohnya adalah *Diablo* yang dikeluarkan *Blizzard*, *Baldur's Gate* yang dikeluarkan *Interplay*, dan *Twinsen's Odyssey* yang dikeluarkan *Activision*.

e. *Flat, Side-View*

Flat, Side-View adalah sudut pandang dalam *game* memberikan tampilan 2 dimensi *side view* dari aksi. Contohnya adalah *Jazz Jackrabbit 2* yang dikeluarkan *Epic Games*, *Worms 2* yang dikeluarkan *Microprose*, *Claw* yang dikeluarkan *Monolith*, dan *Oddworld: Abe Exoddus* atau *Oddworld: Abe's Oddysee* yang dikeluarkan *GT*.

f. *Text-Based Games*

Umumnya sudut pandang *game* ini tidak memerlukan grafik. Jenis sudut pandang ini juga dapat memberikan jenis *game adventure*. Contohnya adalah *Zork Series* dari *Infocom* dan *Hitchhiker's Guide to Galaxy* yang dikeluarkan *Douglas Adams*.

2.4 Rekayasa Perangkat Lunak

2.4.1 Pengertian Perangkat Lunak

Menurut Pressman (2001, p10), yang dimaksud dengan perangkat lunak adalah kumpulan sesuatu yang membentuk konfigurasi tertentu, yaitu:

- a. Program komputer yang jika dijalankan akan memberikan fungsi tertentu.
- b. Struktur data yang memungkinkan program komputer untuk memanipulasi informasi dengan baik.
- c. Dokumen yang menggambarkan operasi kegunaan suatu program.

Menurut Anonimus, perangkat lunak adalah program komputer yang berfungsi sebagai sarana interaksi antara pengguna dan perangkat keras. Perangkat lunak dapat juga dikatakan sebagai 'penerjemah' perintah-perintah yang dijalankan pengguna komputer untuk diteruskan ke atau diproses oleh perangkat keras. Perangkat lunak ini dibagi menjadi tiga tingkatan:

- a. Tingkatan program aplikasi (*application program*).
- b. Tingkatan sistem operasi (*operating system*).
- c. Tingkatan bahasa pemrograman (bahasa pemrograman tingkat tinggi seperti Pascal, C, dan C++).

2.4.2 Karakteristik Perangkat Lunak

Perangkat lunak memiliki beberapa karakteristik, yaitu:

- a. Perangkat lunak tidak dirancang seperti merancang perangkat keras atau dirancang di pabrik.
- b. Jika digunakan dalam jangka waktu yang lama, masa pakai perangkat keras dapat habis. Sebaliknya, masa pakai perangkat lunak tidak akan habis walaupun digunakan dalam jangka waktu yang lama.

- c. Perangkat lunak yang kompleks dapat diatur sesuai dengan keinginan pengguna. Dengan kata lain, perangkat lunak memiliki fleksibilitas yang tinggi.

2.4.3 Konsep-Konsep Rekayasa Perangkat Lunak

Terdapat empat konsep yang menjadi fokus dari pembuatan perangkat lunak sering disebut dengan 4P, yaitu:

a. *People*

Orang-orang yang berperan dalam perancangan, dan penggunaan perangkat lunak sangat penting dalam menentukan keberhasilan suatu proyek pembuatan perangkat lunak. *People* dibagi menjadi empat, yaitu:

- 1) *Senior managers*
- 2) *Project managers*

Karakteristik yang diperlukan sebagai seorang *project managers*, yaitu:

- (a) Dapat memecahkan masalah yang ada (*problem solving*).
- (b) *Managerial identity*
- (c) Pencapaian (*achievement*) yang telah diraih selama ini, contohnya *reward initiatives*.
- (d) *Influence and team building*, contohnya dapat membaca pikiran orang lain.

- 3) *Practitioners*
- 4) *Customers*
- 5) *End-Users*

b. *Product*

Product adalah perangkat lunak yang sedang dirancang oleh *people*. Ruang lingkup produk perangkat lunak terdiri dari:

- 1) Konteks yaitu bagaimana piranti lunak dapat bekerja dalam sistem yang besar, produk, atau konteks bisnis.
- 2) Informasi objek yaitu apa saja output dan input dari perangkat lunak.
- 3) Fungsi dan kinerja dari perangkat lunak.

c. *Process*

Process adalah suatu rangkaian aktivitas dan tugas-tugas dari *software engineering* untuk menyelesaikan pekerjaannya. Pembuatan perangkat lunak terdiri berbagai aktivitas, di antaranya adalah:

- 1) Berkomunikasi dengan pemesan atau *customer*
- 2) Perencanaan
- 3) Analisis risiko
- 4) *Engineering*
- 5) Konstruksi dan pelepasan perangkat lunak
- 6) Evaluasi perangkat lunak yang dilakukan oleh pemesan atau *customer*

d. *Project*

Project adalah semua tugas yang diperlukan untuk menciptakan perangkat lunak menjadi suatu kenyataan. Pendekatan-pendekatan untuk pembuatan perangkat lunak antara lain:

- 1) Mencoba untuk mengerti permasalahan-permasalahan dan menetapkan tujuan secara realistis.
- 2) Memberikan insentif, menekankan pada kualitas, dan meminimalisasi birokrasi.
- 3) Melacak kemajuan.
- 4) Melakukan analisis *postmortem*.
- 5) Membuat keputusan yang bijak dan tetap membuat itu sederhana.

Dalam pembuatan perangkat lunak, tidak semua pembuatan perangkat lunak dilakukan secara individual tetapi pembuatan perangkat lunak juga dapat dikerjakan secara tim.

Beberapa manfaat dari pembuatan perangkat lunak secara tim adalah:

- a. Beberapa permasalahan yang sulit dapat dipecahkan lebih mudah
- b. Tim yang sudah bekerja bersama-sama dalam waktu yang lama akan memiliki kerja sama yang baik.
- c. Pengerjaan perangkat lunak dapat sesuai dengan jangka waktu yang sudah ditetapkan
- d. Tingkat keberhasilan dalam pemecahan masalah akan semakin tinggi
- e. Sistem yang dibangun oleh tim memiliki kualitas yang lebih tinggi
- f. Tingkat komunikasi sangat penting dalam pembuatan perangkat lunak. Oleh karena itu, pembuatan perangkat lunak secara tim lebih baik daripada pembuatan perangkat lunak secara individual.

Pembuatan perangkat lunak secara tim dibagi menjadi tiga jenis, yaitu:

a. *Democratic Decentralized (DD)*

Democratic decentralized memiliki beberapa karakteristik, di antaranya ialah:

- 1) Pemimpin yang tidak tetap
- 2) Keputusan diambil dengan cara pengambilan suara terbanyak
- 3) Jalur komunikasi berjalan secara horizontal

b. *Controlled Decentralized (CD)*

Controlled decentralized memiliki beberapa karakteristik, di antaranya ialah:

- 1) Pemimpin yang sudah ditetapkan
- 2) Pemecahan masalah dilakukan secara tim, tetapi dalam pengimplementasiannya dilakukan oleh sub tim
- 3) Jalur komunikasi berjalan secara horizontal dan vertikal

c. *Controlled Centralized (CC)*

Controlled centralized memiliki beberapa karakteristik, diantaranya ialah:

- 1) Pemimpin yang sudah ditetapkan
- 2) Pemecahan masalah dilakukan oleh pemimpin tim
- 3) Jalur komunikasi berjalan secara vertikal

Perbedaan antara *centralized* dan *decentralized* yaitu:

- a. Struktur *centralized* dapat menyelesaikan tugas-tugas lebih cepat dan sangat cocok untuk masalah-masalah kecil sedangkan *decentralized* dapat menyelesaikan masalah lebih baik sebagai tim daripada individual dan sangat cocok untuk masalah-masalah yang sulit.
- b. Proyek-proyek besar lebih baik dilakukan secara tim dengan menggunakan *centralized* karena lebih mudah mengatur sub tim sedangkan *decentralized* baik digunakan untuk proyek-proyek jangka panjang karena *decentralized* memiliki moral yang lebih tinggi untuk proyek jangka panjang.

Kesimpulannya tipe *centralized* lebih banyak menghasilkan kesalahan daripada tipe *decentralized* sedangkan tipe *decentralized* lebih baik digunakan untuk tugas jangka panjang daripada tipe *centralized*.

Dalam pembuatan perangkat lunak, terdapat beberapa model atau tahapan-tahapan yang digunakan untuk pembuatan perangkat lunak. Menurut Maciaszek dan Liong (2005, p16-23), salah satu dari metode itu adalah *waterfall model*. Tahapan-tahapan dalam *waterfall model*, yaitu:

- a. *Requirements Analysis*

Requirements Analysis atau *requirements engineering* adalah aktivitas memutuskan dan membuat spesifikasi kebutuhan. Ada beberapa metode yang digunakan dalam mengambil kebutuhan, di antaranya:

- 1) Mewawacarai pengguna dan ahli terkait
- 2) Kuesioner
- 3) Observasi pengguna dalam melakukan tugas mereka
- 4) Mempelajari dokumen sistem berjalan
- 5) Mempelajari sistem aplikasi sejenis untuk mengetahui pengetahuan dasar
- 6) Membuat *prototype* model solusi untuk menemukan dan mengkonfirmasi kebutuhan
- 7) Pengadaan sesi dimana pengembang dan pengguna merumuskan kebutuhan

b. *Systems Design*

Systems Design adalah implementasi deskripsi dari struktur perangkat lunak. Struktur perangkat lunak tersebut terdiri dari *data*, *interface*, komponen sistem dan algoritma yang digunakan.

c. Implementasi

Pada implementasi, pembuatan perangkat lunak lebih mengarah pada *programming*. Dalam hal ini, *programming* bukan hanya terbatas pada pembuatan *coding* tetapi *programming* juga melakukan *design* dimana *coding* tersebut akan ditempatkan.

d. *Integration* dan *Deployment*

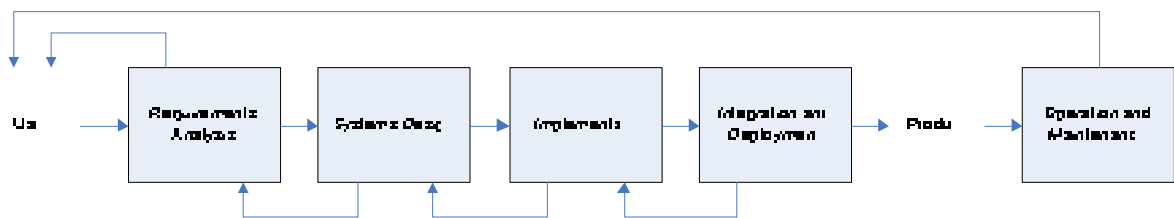
Integrasi (*Integration*) adalah mengumpulkan aplikasi dari himpunan komponen yang sebelumnya telah diimplementasikan dan diujikan. Penyebaran (*deployment*) adalah pemberian atau penyerahan sistem kepada pengguna untuk kebutuhan produksi.

e. *Operation* dan *Maintenance*

Operasi (*Operation*) menekankan pada tahapan daur hidup saat produk perangkat lunak digunakan pada operasi sehari-hari dan sistem sebelumnya (manual ataupun otomatis)

dihapus secara bertahap. Operasi dimulai bersamaan dengan mulainya pemeliharaan (*maintenance*) produk. Pada rekayasa perangkat lunak, pemeliharaan memiliki arti yang sedikit berbeda dengan arti pada penggunaan yang biasanya. Pertama, pemeliharaan bukan hanya sebuah perbaikan masalah yang muncul secara tidak terencana. Pemeliharaan direncanakan pada tahapan awal daur hidup. Kedua, pemeliharaan termasuk evolusi produk. Pada beberapa model daur hidup iteratif, sulit untuk membedakan antara pengembangan dan pemeliharaan.

Untuk selengkapnya, dapat dilihat pada Gambar 2.3.



Gambar 2.3 *Waterfall Model*

2.5 Interaksi Komputer dan Manusia

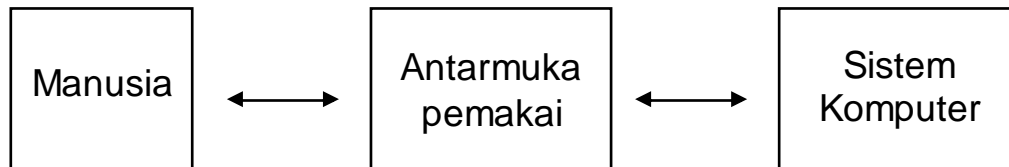
2.5.1 Definisi Interaksi Manusia dan Komputer

Menurut Shneiderman (1998, p10), interaksi manusia dan komputer adalah ilmu yang mempelajari bagaimana manusia berinteraksi dengan komputer dan pengaruh dari interaksi antara manusia dengan komputer.

2.5.2 Fokus Perancangan Antarmuka Pemakai

Fokus dari interaksi manusia dan komputer adalah perancangan dan evaluasi antarmuka pemakai (*user interface*). Antarmuka pemakai atau *user interface* adalah bagian

sistem komputer yang memungkinkan manusia berinteraksi dengan komputer, seperti terlihat pada Gambar 2.4.



Gambar 2.4 *User Interface*

Menurut Shneiderman (1998, p.74), dalam merancang sistem antarmuka pemakai (*user interface*) yang baik terdapat 8 aturan emas yang harus diperhatikan, yaitu:

a. Konsisten

Konsisten dalam hal ini artinya tetap sama dalam penggunaan istilah seperti warna, *layout*, *fonts*, dan sebagainya. Hal ini dimaksudkan agar pengguna tidak bingung ketika menggunakan aplikasi.

b. Tombol-tombol pembantu (*shortcuts*)

Tombol-tombol ini dimaksudkan untuk membantu pengguna berinteraksi dengan komputer dan dapat menghasilkan tampilan layar lebih cepat. Umumnya pengguna dengan kemahiran yang lebih tinggi (*frequent knowledgeable user*) dapat menggunakan perintah tersembunyi.

c. Memberikan umpan balik yang informatif

Sistem yang baik harus memiliki umpan balik yang sering dari system walaupun sekecil apa pun.

d. Merancang dialog untuk menampilkan keadaan terakhir

e. Memberikan penanganan yang sederhana

Sistem yang baik dapat memperbaiki kesalahan yang dilakukan oleh pengguna walaupun kesalahan yang ditimbulkan oleh pengguna sangat kecil.

- f. Sistem mengizinkan pembalikan aksi (*undo*) dengan mudah

Sistem memperbolehkan pengguna untuk membalikkan apa yang sudah pengguna lakukan (*undo*).

- g. Mendukung kontrol *internal locus* (pemakai yang menguasai sistem)

Pengguna dapat menguasai sistem dan tidak terbatas pada merespon sistem saja.

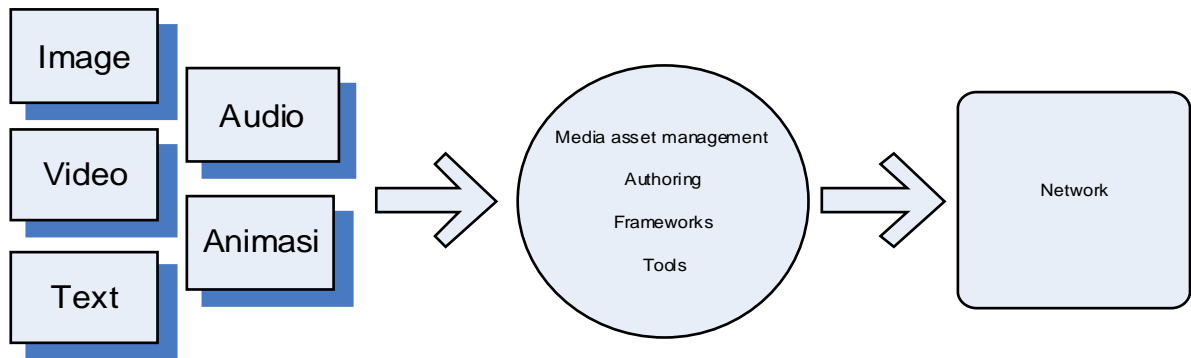
- h. Mengurangi beban ingatan jangka pendek

Dalam hal ini, yang dimaksud adalah mengurangi informasi yang berlebihan dan tidak perlu agar pengguna tidak terlalu banyak menghafal.

2.6 Multimedia

2.6.1 Definisi Multimedia

Menurut Vaughnan (2004), multimedia berasal dari kata multi dan media yang dalam bahasa Yunani disebut multi dengan “*multus*” yang artinya *multiple* atau banyak dan media dengan “*medium*” yang artinya tengah atau dapat disebut juga penengah (*intermediary*). Jadi multimedia adalah kombinasi tersusun dari elemen-elemen yang dimanipulasi secara digital. Elemen-elemen tersebut adalah *image*, *video*, *text*, *audio*, dan *speech*. Proses penggabungan elemen-elemen multimedia dapat dilihat pada Gambar 2.5:



Gambar 2.5 Proses Penggabungan Elemen-elemen Multimedia

Menurut wikipedia, multimedia adalah media atau medium yang mengkombinasikan beberapa bentuk-bentuk aplikasi menjadi satu. Bentuk-bentuk aplikasi yang termasuk dalam kombinasi multimedia adalah teks, audio, *image*, animasi, dan *video*.

2.6.2 Manfaat Multimedia

Menurut Hofstetter (2001), manfaat dalam menggunakan multimedia, yaitu:

- a. Lebih mudah digunakan oleh banyak orang
- b. Pengalaman baru yang didapat pada saat menggunakan multimedia
- c. Antarmuka (*interface*) yang intuitif
- d. Interaksi yang baik
- e. Lebih mudah mengerti konten-konten yang diberikan
- f. Biaya yang efisien
- g. Kesenangan yang lebih dari pada aplikasi lain.

Manfaat multimedia tidak hanya sebatas pada beberapa bidang seperti: iklan, seni, pendidikan, hiburan, rekayasa, matematika, bisnis, penelitian ilmiah, dan tata ruang tetapi juga terdapat dalam berbagai bidang, contohnya:

a. Industri

Dalam bidang industri, multimedia digunakan untuk menyajikan informasi kepada pemegang saham, atasan atau *partner*. Selain itu, multimedia juga digunakan sebagai alat pelatihan bagi karyawan, periklanan, dan penjualan produk melalui website ke seluruh dunia.

b. Simulasi

Pada bidang simulasi, multimedia digunakan untuk mengurangi bahaya dan biaya yang tinggi pada keadaan nyata, contohnya yaitu: dalam pelatihan militer yang sangat berbahaya jika dilakukan secara nyata dan simulasi pada pelatihan penerbangan yang memiliki biaya yang tinggi dan membahayakan jika terjadi kecelakaan.

c. Pengobatan

Pada bidang pengobatan, dokter dapat melakukan pelatihan melalui *virtual* operasi. Selain itu, dokter juga dapat melakukan simulasi penyebaran penyakit pada tubuh manusia sehingga dapat mencegah penyebaran penyakit tersebut.

2.6.3 Elemen-Elemen Multimedia

Multimedia terdiri dari berbagai elemen yang menjadi satu. Elemen-elemen yang menyusun multimedia adalah:

a. Teks

Teks adalah media awal untuk berbagai jenis multimedia lainnya. Teks umumnya terdapat dalam bentuk kata-kata, kalimat-kalimat, dan paragraf. Teks umumnya digunakan untuk menyampaikan pikiran, ide, dan kenyataan yang sering terjadi.

Multimedia yang menggunakan teks umumnya digunakan untuk:

- 1) Menjelaskan bagaimana multimedia itu dijalankan
- 2) Menuntun pengguna dalam menggunakan aplikasi
- 3) Mengantarkan informasi sesuai dengan aplikasi

Teks terdiri dari berbagai elemen-elemen, yaitu:

- 1) Karakter alfabet yaitu A sampai dengan Z
- 2) Angka yaitu 0 sampai dengan 9
- 3) Karakter khusus seperti tanda baca (. , ; ' ...), simbol (* & ^ % \$! @ ~ #) dan lain-lain
- 4) *Character sets* yaitu kumpulan karakter yang tidak berhubungan dengan nilai numerik.

b. *Image*

Image adalah representasi grafik dan visual dari informasi yang dapat ditampilkan pada layar komputer atau media cetak. Gambar terbagi menjadi beberapa bagian, yaitu:

- 1) Foto (*photographs*)
- 2) Gambar (*drawing*)
- 3) Lukisan (*painting*)
- 4) Televisi dan film (*motion pictures*)
- 5) *Semantics*
- 6) Peta (*maps*), dll.

Image memiliki beberapa peran pada aplikasi multimedia, di antaranya adalah:

- 1) Alat navigasi
- 2) Komponen antarmuka pengguna (*user interface*)
- 3) Sistem *help*

4) *Clip art*

Secara umum *image* dibagi menjadi dua bentuk format:

1) *Bitmapped* dan *raster images*

Bitmapped images umumnya disimpan dalam bentuk *array of pixels*.

2) *Draw-type* atau *Vector graphics* atau *metafile Images*

Vector graphics disimpan sebagai bentuk geometri sebagai representasi *image*.

c. Animasi

Animasi adalah gambar yang mengalami pergerakan dari suatu titik menuju titik yang lain. Animasi dibagi menjadi animasi 2D dan animasi 3D.

Animasi 2D terbagi menjadi dua tipe, yaitu:

1) Animasi Sel (*Cel animation*)

Animasi sel adalah animasi yang perubahannya berdasarkan pada pergerakan dari satu *frame* ke *frame* selanjutnya.

2) Animasi Jalur (*Path animation*)

Animasi jalur adalah animasi yang bergerak sesuai dengan jalur yang ditentukan oleh layar.

Pembuatan animasi 3D dibagi menjadi tiga tahap, yaitu:

1) *Modelling*

Pada tahap ini, *broad contours* (kerangka) dan struktur dari obyek 3D dan latar belakang dihasilkan.

2) *Animation*

Tahap ini menentukan pergerakan obyek 3D.

3) *Rendering*

Pada tahap ini terjadi pemberian atribut pada obyek seperti warna, tekstur permukaan, dan transparansi.

Selain animasi 2D dan animasi 3D, animasi juga memiliki *special effect* seperti:

1) *Morphing*

Morphing adalah efek yang menyatukan 2 *image* menjadi 1 *image*.

2) *Warping*

Warping adalah efek yang memberikan pergerakan perubahan pada suatu *image*.

3) *Virtual Reality*

Virtual reality membuat sebuah lingkungan maya dimana pengguna menjadi salah satu bagian dari lingkungan tersebut.

d) Audio atau suara

Suara adalah vibrasi (getaran) cepat yang ditransmisikan sebagai variasi dalam tekanan udara. Pada saat suara bervibrasi, suara akan membenturkan molekul dari media yang ada di sekitarnya yang menyebabkan tekanan gelombang berpencar ke semua arah. Suara terdiri dari kata-kata yang diucapkan, suara manusia, musik, dan suara gangguan yang sering disebut *noise*.

Tipe-tipe suara dalam multimedia adalah:

- 1) Ucapan
- 2) Musik
- 3) Efek suara

Gembang suara memiliki dua jenis karakteristik, yaitu frekuensi dan amplitudo.

e) Video

Video memiliki beberapa jenis yaitu:

1) *Movie* dari film

Movie dari film adalah *frame image* yang direkam secara berurutan yang menggunakan *light sensitive technology* yang mirip dengan kamera pengambilan foto.

2) Video Analog

Video analog adalah informasi video yang disimpan menggunakan sinyal video televisi, film, video *tape* atau media non-komputer lainnya. Setiap *frame* pada analog video dipresentasikan sebagai fluktuasi sinyal voltase yang dikenal dengan gelombang analog.

3) Video digital

Video digital adalah digitasi dari sinyal video analog yang diformat menjadi numerik. Video digital dapat menciptakan ilusi gerakan dengan menampilkan perubahan sekuensial *image* secara cepat pada alat tampil. Konversi dari format analog ke digital membutuhkan sebuah ADC (*Analog to Digital Converter*) dan sebaliknya untuk melakukan konversi dari format digital ke format analog, diperlukan DAC (*Digital to Analog Converter*). Selain itu, DAC dapat digunakan untuk menampilkan video digital pada peralatan analog.

2.7 Bahasa Pemrograman C

2.7.1 Sejarah Bahasa Pemrograman C

Menurut Deitel (2004, p8), bahasa C dirancang oleh Dennis M. Ritchie di Bell Laboratories pada tahun 1972. Bahasa C dikembangkan dari bahasa BCPL dan bahasa B. Bahasa BCPL dikembangkan Martin Richard pada tahun 1967 sebagai bahasa untuk menulis sistem operasi dan *compiler*. Pada tahun 1970, Ken Thompson merancang bahasa B dengan

memasukkan *feature* BCPL. Bahasa B dirancang dengan tujuan untuk digunakan membuat sistem operasi UNIX untuk komputer DEC PDP-7 pada Bell Laboratories.

Pada tahun 1978, Dennis M. Ritchie dan Brian W. Kernighan mempublikasikan bahasa C melalui buku *The C Programming Language*. Pada tahun 1983 komite teknis X3J11 dibentuk di bawah *American National Standards Committee on Computers and Information Processing (X3)* untuk menyusun standar bahasa C, Standar ini terbentuk pada tahun 1989. Pembakuan ini diperbarui pada tahun 1999 dan tercantum pada dokumen ISO/IEC 9899:1999.

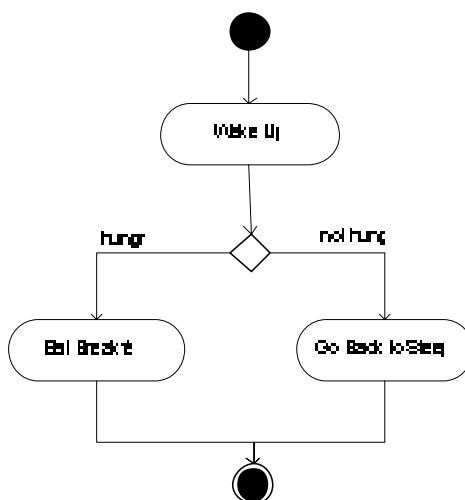
2.7.2 Pengertian Bahasa Pemrograman C

Menurut Deitel, bahasa pemrograman C merupakan bahasa pemrograman yang bersifat umum, tidak dikhususkan untuk bidang aplikasi tertentu. Bahasa pemrograman ini digolongkan sebagai bahasa pemrograman tingkat tinggi (*high-level language*). Hal ini dikarenakan perintah-perintah dalam bahasa pemrograman bahasa C mendekati bahasa manusia yaitu Bahasa Inggris. Selain itu, bahasa pemrograman bahasa C memiliki karakteristik lain seperti hemat ekspresi, alur kontrol, menggunakan struktur data modern, dan kaya dengan operator.

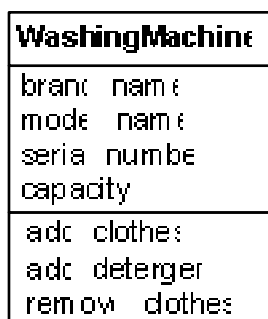
2.8 UML (*Unified Modeling Language*)

Menurut Joseph Schmuller (1999, p8), *Unified Modeling Language (UML)* adalah salah satu dari alat yang paling menarik di antara sistem pengembangan sekarang ini. UML dapat membuat pengembang sistem menciptakan cetak biru yang menangkap visi mereka dalam cara standar yang mudah dimengerti dan mengkomunikasikannya dengan pengembang lain.

Ada beberapa diagram yang digunakan di antaranya *ativity diagram* yang menggambarkan alur proses kerja yang dapat dilihat pada Gambar 2.6 dan *class diagram* yang menyediakan representasi *class-class* yang ada dan interaksinya dan dapat dilihat pada Gambar 2.7.



Gambar 2.6 Activity Diagram



Gambar 2.7 Class Diagram