

BAB II

LANDASAN TEORI

2.1 Artificial Intelligence

2.1.1 Sejarah Artificial Intelligence

Artificial Intelligence (AI) atau Kecerdasan buatan merupakan salah satu bidang ilmu komputer yang didefinisikan sebagai kecerdasan yang dibuat untuk suatu sistem dengan menggunakan algoritma-algoritma tertentu sehingga sistem tersebut seolah-olah dapat berpikir seperti manusia. Berbagai literatur mengenai kecerdasan buatan menyebutkan bahwa ide mengenai kecerdasan buatan diawali pada awal abad 17 ketika Rene Descartes mengemukakan bahwa tubuh hewan bukanlah apa-apa melainkan hanya mesin-mesin yang rumit.

Kemudian Blaise Pascal yang menciptakan mesin penghitung digital mekanis pertama pada 1642. Selanjutnya pada abad 19, Charles Babbage dan Ada Lovelace bekerja pada mesin penghitung mekanis yang dapat diprogram.

Perkembangan terus berlanjut, Bertrand Russell dan Alfred North Whitehead menerbitkan *Principia Mathematica*, yang merombak logika formal. Warren McCulloch dan Walter Pitts menerbitkan “Kalkulus Logis Gagasan yang tetap ada dalam Aktivitas” pada 1943 yang meletakkan pondasi awal untuk jaringan syaraf.

Tahun 1950-an adalah periode usaha aktif dalam AI. Program AI pertama yang bekerja ditulis pada 1951 untuk menjalankan mesin Ferranti Mark I di University of Manchester (UK): sebuah program permainan naskah yang ditulis oleh Christopher Strachey dan program permainan catur yang ditulis oleh Dietrich Prinz.

John McCarthy membuat istilah “Kecerdasan Buatan” pada konferensi pertama pada tahun 1956, selain itu dia juga menemukan bahasa pemrograman Lisp. Alan Turing memperkenalkan “Turing test” sebagai sebuah cara untuk mengoperasionalkan test perilaku cerdas. Joseph Weizenbaum membangun ELIZA, sebuah chatterbot yang menerapkan psikoterapi Rogerian.

Selama tahun 1960-an dan 1970-an, Joel Moses mendemonstrasikan kekuatan pertimbangan simbolis untuk mengintegrasikan masalah di dalam program Macsyma, program berbasis pengetahuan yang sukses pertama kali dalam bidang matematika. Marvin Minsky dan Seymour Papert menerbitkan *Perceptrons*, yang mendemostrasikan batas jaringan syaraf sederhana dan Alain Colmerauer mengembangkan bahasa komputer Prolog. Ted Shortliffe mendemonstrasikan kekuatan sistem berbasis aturan untuk representasi pengetahuan dan inferensi dalam diagnosa dan terapi medis yang diyakini sebagai sistem pakar pertama. Hans Moravec mengembangkan kendaraan terkendali komputer pertama untuk mengatasi jalan yang mempunyai rintangan secara mandiri.

Program kecerdasan buatan lebih sederhana dalam pengoperasiannya, sehingga banyak membantu pemakai. Program konvensional dijalankan secara procedural dan kaku, rangkaian tahap solusinya sudah didefinisikan secara tepat oleh programnya. Sebaliknya, pada program kecerdasan buatan untuk mendapatkan solusi yang memuaskan dilakukan pendekatan trial and error, mirip seperti apa yang dilakukan oleh manusia.

2.1.2 Pengertian AI

Kecerdasan Buatan atau *Artificial Intelligence (AI)* adalah salah satu cabang Ilmu pengetahuan berhubungan dengan pemanfaatan mesin untuk memecahkan persoalan yang rumit dengan cara yang lebih manusiawi. Hal ini biasanya dilakukan dengan mengikuti / mencontoh karakteristik dan analogi berpikir dari kecerdasan / Intelligensia manusia, dan menerapkannya sebagai algoritma yang dikenal oleh komputer.

Dengan suatu pendekatan yang kurang lebih fleksibel dan efisien dapat diambil tergantung dari keperluan, yang mempengaruhi bagaimana wujud dari perilaku kecerdasan buatan. AI biasanya dihubungkan dengan Ilmu Komputer.

Adapun beberapa definisi *Artificial Intelligence* menurut para sumber adalah sebagai berikut :

1. Schalkoff (1990): "Artificial Intelligence adalah bidang studi yang berusaha menerangkan dan meniru perilaku cerdas dalam bentuk proses komputasi".

2. Rich dan Knight (1991): “Artificial Intelligence adalah studi tentang cara membuat komputer melakukan sesuatu yang sampai saat ini orang dapat melakukannya lebih baik”.
3. Luger dan Stubblefield (1993): “Artificial Intelligence adalah cabang ilmu komputer yang berhubungan dengan otomasi perilaku yang cerdas”.
4. Haag dan Keen (1996): “Kecerdasan Buatan adalah bidang studi yang berhubungan dengan penangkapan, pemodelan dan penyimpanan kecerdasan manusia dalam sebuah sistem teknologi informasi sehingga sistem tersebut dapat memfasilitasi proses pengambilan keputusan yang biasanya dilakukan oleh manusia”.

Walaupun AI memiliki konotasi fiksi ilmiah yang kuat, AI membentuk cabang yang sangat penting pada ilmu komputer, berhubungan dengan perilaku, pembelajaran dan adaptasi yang cerdas dalam sebuah mesin. Penelitian dalam AI menyangkut pembuatan mesin untuk mengotomatisasikan tugas-tugas yang membutuhkan perilaku cerdas. Termasuk contohnya adalah pengendalian, perencanaan dan penjadwalan, kemampuan untuk menjawab diagnosa dan pertanyaan pelanggan, serta pengenalan tulisan tangan, suara dan wajah.

Hal-hal seperti itu telah menjadi disiplin ilmu tersendiri, yang memusatkan perhatian pada penyediaan solusi masalah kehidupan yang nyata. Sistem AI sekarang ini sering digunakan dalam bidang

ekonomi, obat-obatan, teknik dan militer, seperti yang telah dibangun dalam beberapa aplikasi perangkat lunak komputer rumah.

Ada banyak jenis kecerdasan buatan, setidaknya ada lima jenis kecerdasan buatan yang sering kita temui, yaitu :

1. Jaringan Syaraf Buatan (Artificial Neural Networks)

Dalam industri minyak bumi AI ini dapat digunakan untuk membuat pola waktu, misal produksi suatu sumur minyak pada waktu tertentu.

2. Logika Fuzzy (Fuzzy Logics)

Logika ini untuk menentukan nilai diantara dua keadaan biner (1 dan 0). Logika inilah yang saya pakai untuk identifikasi suatu minyak bumi sama (1) atau tidak (0) yang akan dibahas lebih lanjut.

3. Algoritma Genetik (Genetic Algorithms)

Algoritma Genetik biasanya digunakan dibidang kedokteran, misal untuk menganalisis DNA.

4. Robotika (Robotics)

AI ini banyak digunakan di pabrik. Biasanya dibuat untuk melakukan kegiatan otomatisasi, misal dalam PLC (Programmable Logic Control).

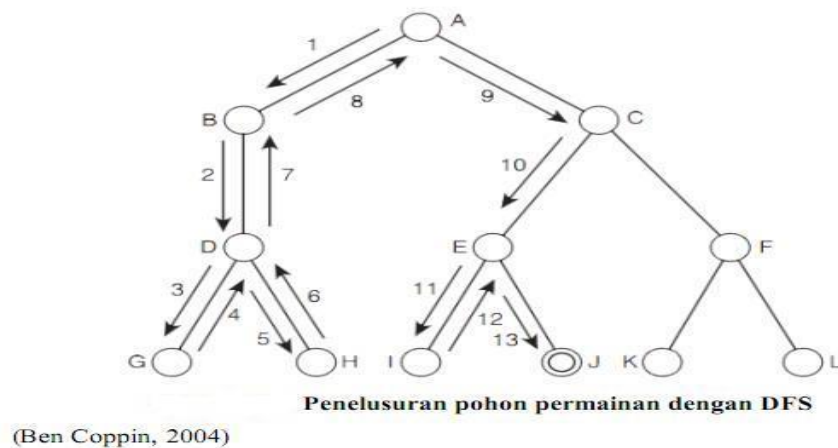
5. Permainan Komputer (Games)

AI jenis ini yang paling disukai oleh anak-anak saya, misal untuk memainkan game Age of Mythology atau Counter Strike

Didalam AI ada beberapa teknik pencarian, salah satunya yang digunakan dalam aplikasi ini yaitu pencarian *Depth-First Search* yang menggunakan pohon pencarian dalam melakukan penelusuran dari setiap langkah yang dijalankan.

2.2 Algoritma Depth First Search (DFS)

Depth First Search (DFS) adalah sebuah algoritma untuk mencari sebuah elemen di dalam sebuah pohon, struktur pohon, atau graf dimana proses yang terjadi merupakan proses pencarian secara mendalam. Proses pencarian dilakukan pada satu simpul dalam setiap level dari yang paling kiri. Jika pada level yang paling dalam, solusi belum ditemukan, maka akan dilakukan proses backtracking (penelusuran balik untuk mendapatkan jalur yang diinginkan) dan pencarian dilanjutkan pada simpul sebelah kanan. Simpul yang kiri dapat dihapus dari memori. Jika pada level yang paling dalam tidak ditemukan solusi, maka dilakukan proses backtracking kembali dan pencarian dilanjutkan pada level sebelumnya. Demikian seterusnya sampai ditemukan solusi. Jika solusi ditemukan, maka tidak diperlukan proses backtracking.



Gambar 2.1: Pohon Pencarian DFS

Pada gambar tersebut dapat diketahui bahwa proses penelusuran *DFS* dimulai dari A-B-D-G-D-H-D-B-A-C-E-I-J. A merupakan *root* yang menandakan keadaan belum diambil, lalu ditelusuri hingga kedalaman yang paling dalam sebelah kiri yaitu G, lalu melakukan proses *backtrack* ke D lalu lanjut lagi ke H.

Proses tersebut akan berhenti karena telah mencapai *goal state* yaitu *leaf node* G. Langkah-langkah cara kerja algoritma *DFS* adalah sebagai berikut :

1. Masukkan *root* ke dalam struktur data tumpukan (*stack*).
2. Ambil simpul dari tumpukan teratas, dan diperiksa apakah simpul merupakan solusi.
3. Jika simpul merupakan solusi, maka pencarian selesai dan hasil dikembalikan.
4. Jika simpul bukan solusi, masukkan seluruh simpul yang bertetangga dengan simpul tersebut kedalam tumpukan.

5. Jika tumpukan kosong dan setiap simpul sudah ditelusuri, pencarian selesai dan solusi tidak ditemukan.
6. Ulangi pencarian dari poin kedua.

Untuk pencarian pada pohon yang memiliki kedalaman yang sangat dalam ataupun hampir tak terhingga, algoritma *DFS*, bisa lebih cepat menemukan jalur menuju solusi pada pohon tersebut. Adapun contoh *pseudocode* untuk algoritma *Depth First Search* dalam notasi algoritmik adalah sebagai berikut :

```

PROCEDURE DFS (INPUT v: INTEGER)
{Mengunjungi seluruh simpul graf dengan algoritma
pencarian DFS
Masukan: v adalah simpul awal kunjungan
Keluaran: semua simpul yang dikunjungi ditulis ke
layar}
Deklarasi
    w : INTEGER
Algoritma:
    WRITE (v)
    dikunjungi[v] ← TRUE
    FOR w ← 1 TO n DO
        IF A[v,w] = 1 THEN
            {simpul v dan simpul w bertetangga}
            IF NOT dikunjungi[w] THEN
                DFS (w)
            END IF
        END IF
    END FOR

```

Runut-balik adalah algoritma yang berbasis pada algoritma DFS untuk mencari solusi persoalan secara lebih optimal. Algoritma ini secara sistematis mencari solusi persoalan di antara semua kemungkinan solusi yang ada. Prinsip kerja dari algoritma ini pemangkasan simpul-simpul yang

tidak mengarah ke solusi. Sehingga, setiap simpul yang tidak memenuhi suatu fungsi pembatas, tidak akan diproses di algoritma ini.

Di dalam algoritma ini, terdapat beberapa fungsi dan variabel yang digunakan di dalam pemrosesan solusi, yaitu ruang solusi, fungsi pembangkit, serta fungsi pembatas. Ruang solusi persoalan dinyatakan sebagai vektor dengan n -tuple. Fungsi pembangkit untuk membangkitkan nilai untuk nilai elemen dengan indeks tertentu yang merupakan komponen vektor solusi. Fungsi pembatas digunakan untuk menentukan apakah suatu simpul dapat mengarahkan ke solusi atau tidak.

Ruang solusi diatur sehingga menjadi sebuah struktur pohon seperti pada algoritma DFS. Setiap simpul dari pohon tersebut menyatakan status persoalan, sedangkan sisi / cabang dilabeli dengan nilai-nilai komponen persoalan. Lintasan dari akar pohon hingga daun menyatakan solusi yang mungkin. Pengaturan pohon ruang solusi ini disebut juga sebagai pohon ruang status. Solusi dicari dengan cara membentuk lintasan dari akar hingga ke daun. Aturan pembentukan yang digunakan mengikuti aturan DFS. Simpul-simpul yang sudah dibentuk dinamakan simpul hidup. Simpul hidup yang sedang dikembangkan dinamakan simpul-E. Setiap kali simpul-E dikembangkan, lintasan yang dibangun pun bertambah panjang. Jika lintasan tersebut tidak mengarah ke solusi, maka simpul-E tersebut ditinggalkan sehingga menjadi simpul mati. Simpul yang sudah mati tidak akan pernah diperluas lagi.

Apabila pembentukan lintasan berakhir dengan simpul mati, maka proses pencarian pun diteruskan dengan membangkitkan simpul anak yang

lainnya. Bila tidak ada lagi simpul anak yang dapat dibangkitkan, maka dapat dilakukan runut-balik ke simpul orangtua yang terdekat dan masih merupakan simpul hidup. Simpul orangtua ini pun yang akan menjadi simpul-E yang baru. Proses berhenti apabila telah ditemukan solusi atau tidak ada lagi simpul hidup untuk runut-balik.

2.3 Teori Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah pengembangan dan penggunaan prinsip pengembangan untuk memperoleh perangkat lunak secara ekonomis yang reliable dan bekerja secara efisien pada mesin nyata. Pada definisi tersebut Bauer memberi kita suatu dasar melalui berbagai pertanyaan seperti berikut ini:

- Bagaimana kita secara ekonomis membangun perangkat lunak sehingga dapat diandalkan reliable-nya?
- Apakah yang dibutuhkan untuk menciptakan program komputer yang bekerja secara efisien pada lebih dari satu mesin riil yang berbeda?

IEEE telah mengembangkan definisi yang lebih komprehensif, yaitu sebagai berikut:

Rekayasa perangkat lunak:

- (1) Aplikasi dari sebuah pendekatan kuantifiabel, disiplin dan sistematis kepada pengembangan, operasi dan pemeliharaan perangkat lunak; yaitu aplikasi dari Rekayasa perangkat lunak.
- (2) Studi tentang pendekatan-pendekatan seperti pada (1).

2.3.1 Metode Rekayasa Perangkat Lunak

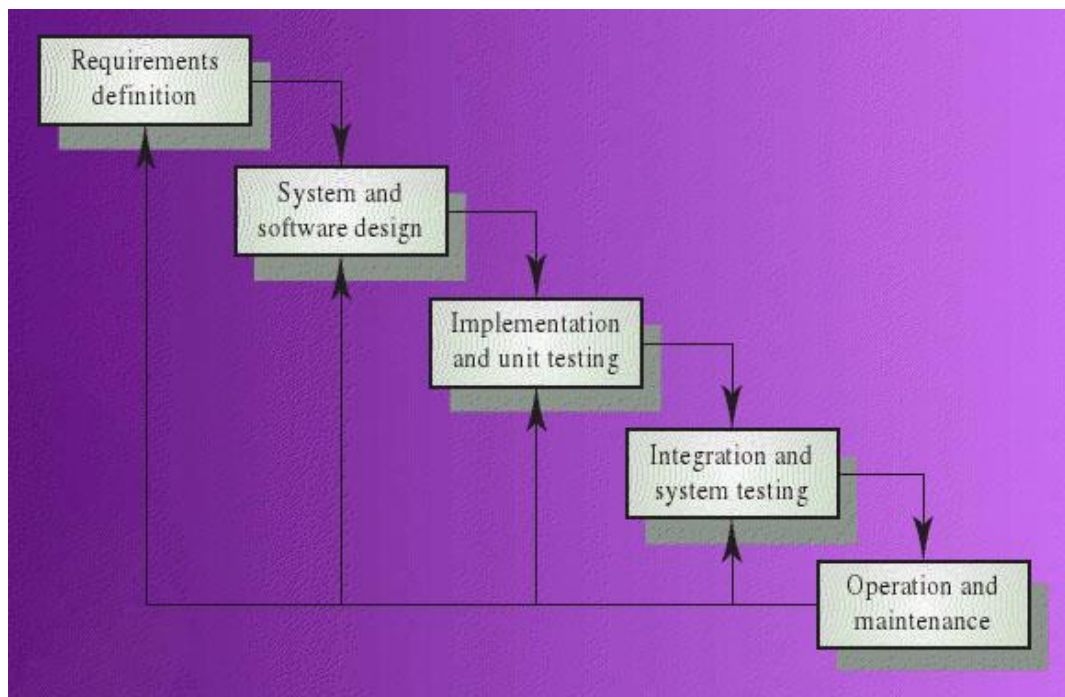
Pendekatan-pendekatan terhadap pengembangan perangkat lunak mencakup model, notasi, aturan, saran pengembangan sistem (rekomendasi), dan panduan proses.

Beberapa metode Rekayasa Perangkat Lunak:

- Metode Structur Analysis (DeMarco 1978) yang mendefinisikan komponen fungsional dasar
- Sistem Berorientasi Obyek (Booch 1994 dan Rambaugh et al 1991)
- UML-Unified Modelling Language (Fowler and Scoot 1997, Booch et al 1999, Rambaugh et al 1999)

2.3.2 Model Waterfall

Model Waterfall merupakan model yang menggunakan milestone sebagai titik transisi dan pengujian, artinya setiap aktivitas pada tahap pengembangan harus diselesaikan sebelum menuju tahap pengembangan berikutnya. Kondisi semacam ini akan sangat berpengaruh pada perangkat lunak dan menimbulkan masalah terhadap kebutuhan iterasi dimana aplikasi akan terus berkembang dengan penyesuaian-penyesuaian terhadap kebutuhan, proses bisnis dan lingkungan aplikasi yang terus berubah dari waktu ke waktu.



Gambar 2.2 Model Proses Waterfall

Adapun tahap-tahap pengembangan perangkat lunak menurut model waterfall yaitu:

- **Requirements analysis and definition:**

Pada tahap ini tim pengembang mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan mendefinisikan kebutuhan-kebutuhan proses bisnis yang harus dipenuhi oleh perangkat lunak yang akan dibangun.

- **System and software design:**

Pada tahap ini tim pengembang mendesain sistem dan aplikasi meliputi desain konseptual, logikal dan fisik.

- **Implementation and unit testing:**

Pada tahap ini desain yang telah di rancang diimplementasikan dengan menerjemahkan ke dalam kode-kode program menggunakan sebuah bahasa pemrograman, sekaligus melakukan pengujian terhadap unit-unit program yang telah dibuat.

- **Integration and system testing:**

Tim pengembang menyatukan unit-unit program kemudian melakukan pengujian sistem perangkat lunak secara keseluruhan.

- **Operation and maintenance:**

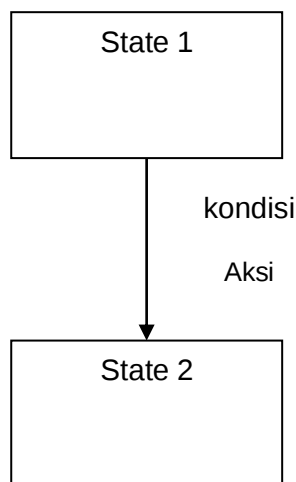
Tim pengembang melakukan pengoperasian program dan melakukan pemeliharaan terhadap perangkat lunak dengan penyesuaian atau perubahan terhadap situasi sebenarnya.

2.4 State Transition Diagram

Pengertian *State Transition Diagram* adalah diagram yang menggambarkan hasil perubahan dari *state* selama dilakukannya proses dalam proses *state* yang terbatas.

State Transition Diagram merupakan diagram yang menggambarkan bagaimana *state* yang satu dihubungkan dengan *state* yang lain pada satu waktu. STD menggambarkan suatu *state* yang mempunyai kondisi dimana dapat menyebabkan perubahan satu *state* ke *state* yang lain.

Pada dasarnya merupakan sebuah diagram yang terdiri dari *state* dan transisi atau perpindahan *state*. Transisi atau perpindahan *state* terdiri dari kondisi dan aksi. Transisi diantara kedua keadaan pada umumnya disebabkan oleh suatu kondisi. Kondisi adalah suatu kejadian yang dapat diketahui oleh sistem. Sedangkan aksi adalah tindakan yang dilakukan oleh sistem apabila terjadi perubahan *state* atau merupakan reaksi dari sistem.

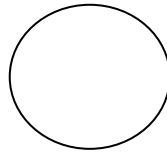


Gambar 2.3 Contoh Perubahan State

Adapun komponen atau symbol yang digunakan dalam diagram STD adalah sebagai berikut:

a. Modul

Menggunakan simbol lingkaran kecil (Gambar 2.3) yang mewakili modul yang dipanggil apabila terjadi suatu tindakan.



Gambar 2.4 Notasi Modul

b. Tampilan kondisi (*state*)

Merupakan layer yang ditampilkan menurut keadaan atau atribut, untuk memenuhi suatu tindakan pada waktu tertentu yang mewakili suatu bentuk keberadaan atau kondisi tertentu.



Gambar 2.5 Notasi Tampilan

c. Tindakan (*state transition*)

Menggunakan simbol anak panah (Gambar 2.5) disertai keterangan tindakan yang dilakukan.



Gambar 2.6 Notasi Transisi

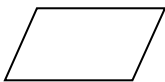
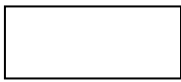
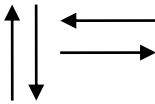
2.5 Flowchart

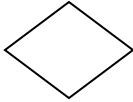
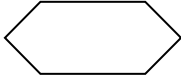
Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. Flowchart menolong analisis dan

programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.

Flowchart Program dihasilkan dari Flowchart Sistem. Flowchart Program merupakan keterangan yang lebih rinci tentang bagaimana setiap langkah program atau prosedur sesungguhnya dilaksanakan. Flowchart ini menunjukkan setiap langkah program atau prosedur dalam urutan yang tepat saat terjadi. Programmer menggunakan flowchart program untuk menggambarkan urutan instruksi dari program komputer. Analis Sistem menggunakan flowchart program untuk menggambarkan urutan tugas-tugas pekerjaan dalam suatu prosedur atau operasi.

Tabel 2.1 Simbol-simbol Dalam Flowchart

No	Gambar	Nama	Keterangan
1.		Simbol input/output (<i>input/output symbol</i>)	Digunakan untuk mewakili data input/output
2.		Simbol proses	Digunakan untuk mewakili suatu proses
3.		Simbol garis alir (<i>flow line symbol</i>)	Digunakan untuk menunjukkan arus dari proses

4.		Simbol penghubung (<i>connector symbol</i>)	Digunakan untuk menunjukkan sambungan dari bagan alir yang terputus di halaman yang masih sama atau di halaman lainnya
5.		Simbol keputusan (<i>decision symbol</i>)	Digunakan untuk suatu penyeleksian kondisi di dalam program
6.		Simbol proses terdefinisi (<i>predifined process symbol</i>)	Digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain
7.		Simbol persiapan (<i>preparation symbol</i>)	Digunakan untuk memberi nilai awal suatu besaran
8.		Simbol terminal (<i>terminal point symbol</i>)	Digunakan untuk menunjukkan awal dan akhir dari suatu proses

2.6 Analisa Interaksi Manusia dan Komputer

Multimedia merupakan perkembangan teknologi komputer yang saat ini menjadi perhatian luas. Istilah multimedia digunakan untuk menjelaskan suatu sistem yang terdiri dari perangkat keras (*hardware*), perangkat lunak (*software*), serta alat bantu lain seperti televisi, monitor video dan sistem pirngan optik atau sistem stereo yang semuanya dikombinasikan untuk menghasilkan suatu penyajian *audio visual* penuh. (Mc Leod Raymond, 1996)

Definisi lain yang menyatakan bahwa multimedia adalah kombinasi dari komputer dan video atau multimedia secara umum merupakan kombinasi dari tiga elemen yaitu suara, gambar dan teks atau multimedia merupakan kombinasi dari teks, grafik, animasi dan video pada komputer. Multimedia juga merupakan gabungan dari dua atau lebih media yang berbeda pada suatu *personal computer* (PC).

Penggunaan multimedia sangat penting sekali dan lebih cenderung digunakan pada aplikasi hiburan, seperti game dan pembuatan animasi bergerak yang dapat menarik minat para pengguna.