

BAB 2

LANDASAN TEORI

2.1 Multimedia

2.1.1 Pengertian Multimedia

Multimedia diartikan sebagai sebuah kombinasi dari teks, gambar, suara, animasi, dan video yang ditampilkan dalam komputer atau peralatan elektronik lainnya. Jika pemakai dapat mengontrol apa dan kapan elemen yang disajikan maka hal tersebut dikatakan multimedia interaktif. (Vaughan, 1994, p5).

Sedangkan menurut J. Burger (1999, p4) multimedia adalah perpaduan dua media atau lebih dengan suatu personal komputer.

Melalui multimedia, komputer dapat memberikan informasi kepada pemakai dalam bentuk audio dan visual. Multimedia memungkinkan pemakai komputer untuk dapat melihat gambar 3D, foto, video, animasi, dan mendengar suara. Hal ini dapat memberikan *output* dalam bentuk yang jauh lebih bagus daripada tabel dan grafik.

2.1.2 Elemen-elemen Multimedia

Berikut ini merupakan elemen-elemen multimedia yang dapat digunakan untuk membuat aplikasi multimedia :

1. Teks

Merupakan elemen multimedia yang menjadi dasar utama dalam penyampaian informasi. Teks memegang peranan dasar dalam menyusun dokumen karena hampir seluruh aplikasi multimedia menggunakan teks sebagai alat presentasi informasi yang paling sesuai untuk mendeskripsikan suatu nama, definisi serta aturan.

Teks merupakan jenis data yang paling sederhana dan membutuhkan tempat penyimpanan yang paling kecil, biasanya dihasilkan oleh program pengolah kata dan merupakan informasi yang utama pada sebagian besar multimedia.

2. Grafik

Pada multimedia, grafik berfungsi untuk mengilustrasikan informasi yang akan disampaikan secara jelas. Grafik dapat direpresentasikan berupa grafik batang, grafik pie dan lain-lain dalam bentuk dua dimensi maupun tiga dimensi.

3. Animasi

Merupakan gambar gerak. Animasi dapat dihasilkan dengan menayangkan frame- frame gambar secara cepat untuk menghasilkan efek pergerakan. Format file untuk menyimpan tayangan animasi sangat bervariasi, diantaranya adalah *Animated GIF* yang merupakan pengembangan dari format gambar GIF (*Graphics Interchange Format*).

Animasi terdiri dari dua, yaitu :

- Animasi buatan (*Computer Based Animation*)

Animasi yang dihasilkan oleh manusia melalui bantuan program komputer.

- Animasi video (*Full Motion Video*)

Animasi yang dihasilkan melalui rekaman dari kamera video dan kejadian yang sesungguhnya.

4. Suara atau Audio

Merupakan fenomena fisik yang dihasilkan oleh pergetaran materi.

Suara terbagi menjadi tiga kategori, yaitu :

- Ucapan (*Speech*) : suara orang berbicara.
- Musik (*Music*) : suara yang dihasilkan oleh alat musik.
- Efek suara (*Sound effect*) : suara yang dibuat untuk menciptakan kesan suatu kejadian seperti suara tembakan, halilintar, gelas pecah, dan lain-lain.

5. Citra

Merupakan representasi spasial dari objek. Citra dapat dibagi menjadi tiga kategori :

- Gambar dari dunia nyata : lukisan atau *scanning* foto.
- Gambar dari dunia maya : dibuat dengan program editor.
- Gambar gabungan dunia nyata dan dunia maya.

6. Video

Merupakan gabungan elemen multimedia yang lengkap karena menggabungkan semua elemen multimedia yang ada untuk menyajikan informasi. Video menggunakan sistem animasi yang diambil melalui suatu kamera video dan disimpan dalam bentuk file dan format tertentu.

Format data file video yang terkenal diantaranya adalah *Audio Video* (*.Avi), *Movie* (*.Mov) dan *Motion Picture Expert* (*.MPEG). Ukuran format data video relatif besar karena merupakan integrasi dari beberapa format komponen multimedia lain.

2.1.3 Aplikasi Multimedia

Penggunaan multimedia dapat digunakan dalam bidang bisnis, pendidikan maupun hiburan (Hofstetter, 2001, p29). Beberapa aplikasi multimedia yang kini banyak digunakan yaitu :

1. *Bussiness Automation* (otomatisasi bisnis), yaitu aplikasi multimedia untuk keperluan bisnis dimana informasi yang disajikan akan terlihat lebih menarik dan interaktif, misalnya dalam presentasi. Bentuk lain dari aplikasi bisnis yaitu *e-commerce*, *point of sales* serta kios multimedia umum.
2. *Transaction Processing* (proses transaksi), yaitu aplikasi multimedia yang digunakan sebagai pendukung proses transaksi, misalnya kartu kredit, mesin ATM, serta tagihan asuransi medis.
3. *Monitoring dan Alarm* (pengawasan dan peringatan), yaitu penggunaan aplikasi multimedia meningkatkan sistem keamanan, kebakaran dan penanganan medis.
4. *Education* (pendidikan), yaitu penggunaan aplikasi multimedia dalam menyampaikan teori dan praktek agar lebih mudah dan jelas sehingga topik pendidikan menjadi lebih menarik dan interaktif. Banyak

pengajaran yang menggunakan multimedia dengan menggabungkan gambar, suara, dan video, seperti:

- Perangkat ajar bahasa.
- Perangkat ajar berhitung untuk anak-anak.
- Alat peraga biologi.
- Kamus.

5. *Entertainment* (hiburan), yaitu penggunaan aplikasi multimedia untuk bidang hiburan. Merupakan bidang yang cukup banyak menggunakan multimedia untuk menciptakan suatu dimensi baru dan terus berkembang dalam memenuhi kepuasan manusia seperti *computer games*, film animasi dan *virtual reality*.

6. Kedokteran dan pengobatan

Multimedia dapat menciptakan suatu paket informasi kedokteran seperti anatomi tubuh manusia, juga dapat digunakan dalam hal pelatihan kedokteran seperti latihan pembedahan.

7. Sumber Ensiklopedia

Dengan multimedia dapat dibuat suatu ensiklopedia dalam bentuk CD atau DVD. Ensiklopedia ini menyediakan suatu pencarian sehingga mudah menemukan informasi yang diinginkan.

2.2 Rekayasa Piranti Lunak

2.2.1 Pengertian Rekayasa Piranti Lunak

Rekayasa Piranti Lunak menurut Fritz Bauer (Pressman, 1998, p23) adalah penetapan dan penggunaan prinsip-prinsip rekayasa untuk

mendapatkan software yang ekonomis yaitu software yang dapat diandalkan (*reliable*) dan bekerja efisien pada mesin yang *real*.

Sedangkan pengertian dari perangkat lunak / software adalah sebagai berikut :

1. Instruksi-instruksi (program komputer) yang bila dijalankan akan memberikan fungsi dan unjuk kerja yang diinginkan.
2. Struktur data yang memungkinkan program untuk memanipulasi informasi secara adekwat.
3. Dokumen-dokumen yang menjelaskan operasi dan penggunaan program-program.

2.2.2 Karakteristik Peranti Lunak

Peranti lunak mempunyai tiga karakteristik yang membedakannya dengan perangkat keras, yaitu :

1. Sistem elemen software bersifat logika (bukan fisik).
2. Software dikembangkan atau direkayasa.
3. Software tidak rusak.

2.2.3 Elemen-elemen Rekayasa Peranti Lunak

Rekayasa peranti lunak memiliki tiga elemen utama yaitu :

1. Metode-metode

Elemen ini menyediakan kebutuhan bagaimana secara teknis membangun software.

Terdiri dari:

- Perencanaan proyek dan estimasi
- Analisis kebutuhan sistem dan software
- Rancangan struktur data
- Arsitektur program
- Algoritma prosedur
- Pengkodean
- *Testing*
- Pemeliharaan

2. Alat-alat bantu

Elemen ini menyediakan dukungan otomatis dan semi otomatis untuk metode- metode dalam rekayasa software. *Computer Aided Software Engineering* (CASE) adalah salah satu *tools* yang digunakan untuk mendukung pengembangan software. CASE mengkombinasikan hardware, software, dan *software engineering database* untuk menciptakan suatu *software engineering environment*.

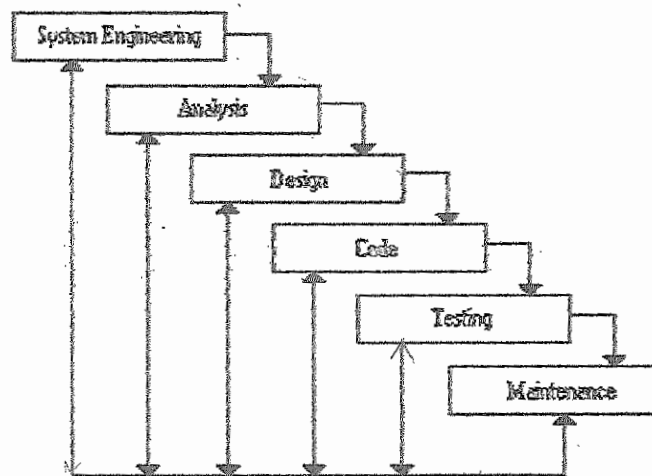
3. Prosedur-prosedur

Elemen ini akan menggabungkan metode dan alat bantu untuk mengembangkan komputer software yang baik. Prosedur-prosedur menentukan urutan metode, mendefinisikan keluaran seperti dokumen-dokumen, laporan-laporan dan formulir-formulir yang dibutuhkan, kontrol yang membantu keyakinan kualitas dan perubahan koordinasi, dan menentukan *milestone* (jarak antara 2 proses) yang memungkinkan manager memperkirakan kemajuan, agar pengembangan komputer software dapat dilakukan tepat waktu.

2.2.4 Paradigma Rekayasa Piranti Lunak

Ada empat paradigma yang digunakan untuk mengembangkan rekayasa perangkat lunak (Pressman, 1997, p29 - p47), yaitu:

1. *The Classic Life Cycle* atau sering juga disebut dengan *Waterfall Model*.



Gambar 2.1 : *Waterfall Model*

- *System engineering*

Pengembangan software dilakukan dimulai dari analisis kebutuhan-kebutuhan software. Analisis kebutuhan ini penting untuk dikerjakan karena akan menentukan konektivitas antara hardware, user, dan database-database .

- *Software requirement analysis*

Pada tahap selanjutnya hasil pengumpulan kebutuhan-kebutuhan software ini akan difokuskan secara khusus pada software. Untuk membangun suatu software yang sesuai dengan permintaan user, *software engineer* harus mengetahui fungsi-fungsi yang dibutuhkan,

user interface, kebutuhan software untuk didokumentasikan dan dibahas bersama dengan *customer*.

- *Design*

Pada tahap ini akan dibuat desain dari software. Pada saat dilakukan desain akan diterjemahkan kebutuhan-kebutuhan dari user dan akan dibuat perancangan software sebelum dilakukan *coding*. Perancangan ini juga akan didokumentasikan dan akan menjadi bagian dari konfigurasi software yang dibuat.

- *Coding*

Setelah dilakukan design software maka tahapan selanjutnya yaitu *coding*. *Coding* dilakukan untuk menterjemahkan perancangan yang telah dibuat ke dalam bentuk yang dapat dibaca oleh mesin.

- *Testing*

Setelah program selesai dibuat maka tahapan selanjutnya akan dilakukan *testing*. Proses *testing* ini difokuskan pada :

- a. *Logical internals of the software*

Pengetesan terhadap *statement-statement*.

- b. *Functional external*

Pengetesan dilakukan untuk menemukan *error* pada program.

Pada pengetesan ini *input* yang diberikan harus mendapatkan *output* yang sesuai dengan yang diharapkan.

- *Maintenance*

Maintenance (pemeliharaan) harus dilakukan untuk mengantisipasi perubahan-perubahan pada sistem. Perubahan ini dapat disebabkan oleh :

- a. *Error* ditemukan.
- b. Perubahan dari lingkungan sistem, misalnya perubahan hardware dan sistem operasi yang baru.
- c. *Customer* memerlukan pengembangan sistem yang baru.

2. *Prototyping*

Prototyping adalah suatu proses untuk membuat suatu model software.

Prototype dapat berbentuk sebagai berikut:

- *Paper prototype*

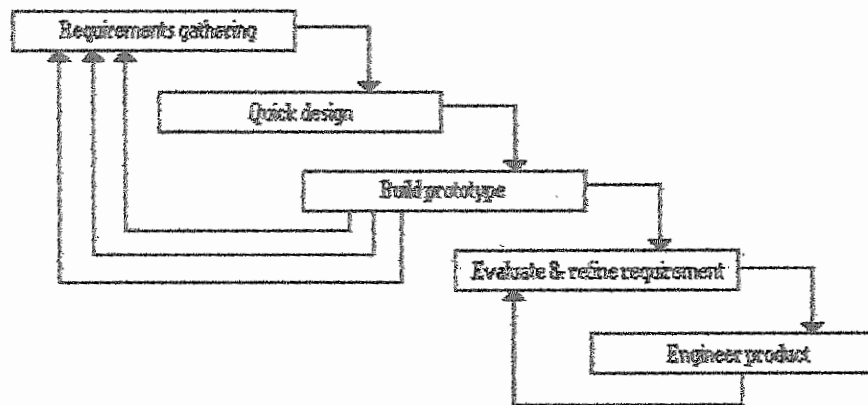
Menggambarakan *human machine interaction* dalam suatu bentuk tertentu sehingga memungkinkan pengguna mengerti bagaimana interaksi tersebut muncul.

- *Working Prototype*

Mengimplementasikan beberapa subset dari fungsi-fungsi dalam software.

- *Existing program*

Mengembangkan sebagian atau semua fungsi yang diinginkan agar mempunyai *features* lain.



Gambar 2.2 : *Prototype Model*

3. *The Evolutionary Paradigm*

Paradigma ini biasa disebut dengan *Spiral Model*. *Spiral model* dikembangkan untuk menggabungkan antara *classic life*, *prototyping* dan *risk analysis*.

Spiral model mencakup empat komponen atau aktivitas utama, yaitu:

- *Management planning*

Menentukan dan membuat rencana kebutuhan dasar user.

- *Risk analysis*

Merupakan aktivitas-aktivitas statistik yang formal yang digunakan untuk memahami dan menghindari *key project risks*.

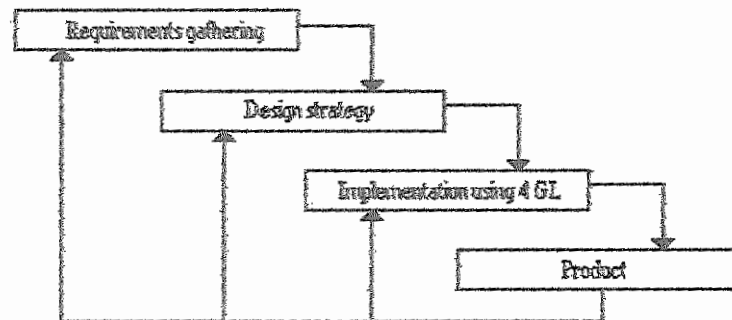
- *Engineering*

Mengaplikasikan metode-metode dalam membuat suatu software.

- *Customer evaluation*

Evaluasi hasil-hasil rekayasa dan memungkinkan untuk adanya perubahan program.

4. Fourth Generation Techniques (4GT)



Gambar 2.3 : The Fourth Generation Techniques

Paradigma 4GT didukung oleh beberapa *tools* berikut ini :

- *Nonprocedural language* untuk *database query*
- *Report generation*
- Manipulasi data
- *High level graphics capability*
- *Screen interaction and definition*
- *Code generation*
- *Spreadsheet capability*

2.2.5 Alat Bantu Perancangan

2.2.5.1 Database

Menurut McLeod (2001, p12), data adalah fakta dan angka-angka yang secara relatif berguna untuk user. Database adalah sekumpulan data yang telah diorganisir untuk melayani beberapa aplikasi pada waktu yang sama, disimpan dan diatur sehingga data diletakkan pada suatu lokasi.

Sedangkan *Database Management System* (DBMS) adalah kumpulan data yang saling berhubungan dimana kita dapat memodifikasi dan mengekstrak informasi dari sebuah database. Tujuan utama sebuah DBMS adalah menyediakan cara pengelompokkan dan penerimaan kembali informasi yang berfungsi untuk memudahkan proses dan efisien.

Langkah-langkah yang dipersiapkan dalam pembuatan database sebagai berikut :

1. Mendisain sebuah database harus sesuai dengan apa yang dibutuhkan dan memenuhi kebutuhan informasi yang dibutuhkan user.
2. Pembuatan langkah dalam pengumpulan data aktual dan cara penyimpanannya sesuai dengan definisi untuk apa database tersebut dibuat.

Ketika sebuah database selesai dibuat, database tersebut harus dapat tersedia untuk user dalam proses pemunculan data kembali dan *update*. User yang melakukan revisi suatu database akan melalui langkah-langkah sebagai berikut :

1. Mendefinisi ulang skema database.
2. Menstruktur dan mengorganisir ulang data yang disimpan.
3. Merubah definisi proses dari database yang direvisi.
4. Merubah cara pandang dan kerja interaksi user dalam database yang baru.

2.2.5.2 STD

State Transition Diagram (STD) adalah model dari tingkah laku sistem yang didasarkan pada definisi satu bagian dari keadaan sistem. Keadaan atau *state* adalah suatu model tingkah laku yang ditemukan Roger S. Pressman (1992, p127).

STD sering digunakan untuk menggambarkan kinerja dari sistem. Melalui STD, tingkah laku sistem dapat dimengerti serta ditunjukkan dan yang lebih penting adalah meyakinkan apakah ada yang kurang atau tertinggal dari tingkah laku yang telah dispesifikasikan. STD merupakan suatu alat untuk merancang suatu system yang digunakan secara *realtime* misalnya proses kontrol, *telephone switching system*, dan *control system*.

Notasi-notasi yang digunakan dalam STD adalah sebagai berikut :

1.  : Menyatakan *state* atau kondisi dari suatu sistem. *State* terdiri dari dua macam yaitu *initial state* atau state awal dan *final state* atau state akhir. *Final state* bisa terdiri atas beberapa state, tetapi *initial state* tidak boleh lebih dari satu.
2.  : Menyatakan perubahan *state* atau kondisi dari suatu sistem.
3. Kondisi : Menyatakan suatu kejadian pada lingkungan eksternal yang dapat dideteksi oleh suatu sistem misalnya suatu signal atau data.

4. Aksi : Sesuatu yang dilakukan oleh sistem terjadi perubahan state atau merupakan reaksi terhadap kondisi. Aksi akan menghasilkan *output*, *message display* pada screen, menghasilkan kalkulasi dan lain-lain.

2.2.5.3 Spesifikasi Proses

Spesifikasi proses merupakan penjelasan dari proses-proses yang terjadi dalam sistem. Spesifikasi proses harus dapat dimengerti baik oleh pemakai maupun pembuat sistem. Spesifikasi proses dapat menjadi pedoman bagi pembuat program dalam membuat kode program (*coding*).

Spesifikasi proses digunakan untuk menggambarkan semua alur proses yang terjadi pada tahapan akhir. Isi spesifikasi proses dapat berupa teks naratif, gambaran proses dari algoritma, persamaan matematika, tabel, diagram atau bagan.

2.3 Interaksi Manusia dan Komputer

2.3.1 Pengertian Interaksi Manusia dan Komputer

Interaksi manusia dan komputer adalah disiplin ilmu yang berhubungan dengan perancangan, evaluasi, serta implementasi sistem komputer yang interaktif untuk digunakan oleh manusia. Interaksi manusia dan komputer berkaitan dengan *user interface* (antarmuka pemakai) yang digunakan oleh pengguna untuk berkomunikasi dan berinteraksi dengan komputer (Shneiderman, 1998, pp4-8).

Hal-hal yang perlu diperhatikan dalam merancang suatu antarmuka yang *user friendly*, antara lain :

1. Waktu untuk belajar

Berapa lama waktu yang diperlukan bagi pengguna pada umumnya untuk mempelajari bagaimana menggunakan perintah yang berhubungan dengan suatu *task*.

2. Kecepatan kinerja

Berapa waktu yang diperlukan untuk menjalankan suatu *task*.

3. Tingkat kesalahan pengguna

Berapa banyak dan apa saja kesalahan yang dilakukan pengguna dalam melaksanakan suatu *task*.

4. Retensi pengguna setelah jangka waktu tertentu

Berapa lama pengguna dapat mempertahankan pengetahuan mereka setelah jangka waktu tertentu. Retensi dapat dihubungkan dengan waktu belajar, dan frekuensi penggunaan memegang peranan penting dalam hal ini.

5. Kepuasan subyektif pemakai

Mencari tahu apakah pengguna sering menggunakan berbagai aspek dari sistem. Jawabannya dapat diperoleh dari wawancara atau survei tertulis yang memuat skala kepuasan dan tempat bagi jawaban bebas.

2.3.2 Prinsip-prinsip Perancangan Interface

Menurut Shneiderman (1992, pp74-75), ada delapan aturan emas (*eight golden rules*) yang harus diperhatikan dalam merancang suatu sistem interaksi manusia dan komputer yang baik :

1. Berusaha keras untuk konsisten, dalam penggunaan warna, huruf harus seragam.
2. Memungkinkan pemakai rutin untuk menggunakan *shortcuts*.
3. Memberikan umpan balik yang informatif.
4. Merancang dialog yang dapat menghasilkan akhir dari suatu aksi.
5. Pencegahan dan penanganan kesalahan yang sederhana.
6. Memperbolehkan pengguna untuk membatalkan suatu aksi.
7. Mendukung *internal locus of control*, artinya manusia yang memegang kendali atas sistem, bukan sebaliknya.
8. Mengurangi beban ingatan jangka pendek.

2.4 Perangkat Ajar

2.4.1 Sejarah Perkembangan Perangkat Ajar

Di era globalisasi ini, perkembangan teknologi telah semakin pesat, khususnya teknologi komputer yang pada mulanya komputer diciptakan hanya untuk keperluan proses hitung-menghitung. Namun, sekarang ini komputer telah dapat digunakan untuk berbagai keperluan baik dalam bidang pengetahuan maupun pendidikan ataupun untuk penelitian. Dalam bidang pendidikan, komputer diperlukan sebagai alat bantu dalam proses

pembelajaran pelajar. CAI (*Computer Aided Instructions*) adalah salah satu bentuk aplikasi penerapan kegunaan komputer dibidang pendidikan.

CAI masuk pada tahun 1950-an di Sekolah Dasar dan awal tahun 1960-an di AS. Sistem untuk mata pelajaran yang lebih tinggi dimulai pada awal tahun 1960-an. CAI mulai diperkenalkan oleh Harvard University sekitar tahun 1965. Dasar pemikiran yang melatarbelakangi istilah tersebut adalah kemungkinan digunakan komputer sebagai alat bantu dalam dunia pendidikan. Untuk pengembangan dan implementasinya Harvard University bekerja sama dengan pihak IBM (*International Business Machine*) dengan melakukan penelitian lebih lanjut.

Menurut Chambers (1989, p5) istilah perangkat ajar di AS dikenal dengan CAI (*Computer Aided Instructions*), CBI (*Computer Based Instructions*) dan CBE (*Computer Based Education*). Sedangkan di Eropa lebih dikenal dengan CAL (*Computer Assisted Learning*) dan di Inggris dikenal dengan CBT (*Computer Based Training*).

2.4.2 Pengertian Perangkat Ajar

Computer Aided Instructions (CAI) adalah suatu cara belajar yang efektif, orang belajar harus secara berkesinambungan mengerjakan sesuatu seperti menjawab pertanyaan, memilih topik, bertanya dan sebagainya. Dengan konsep ini diharapkan cara belajar pasif dapat diubah menjadi cara belajar yang lebih aktif. (Kuntjahjani, 2000, p87).

Menurut Steward, *Computer Assisted Instructions* (CAI) menunjukkan kepada penggunaan komputer sebagai alat bantu dalam pengajaran atau proses belajar (Steward, 1987, p190).

Jadi *Computer Aided Instructions* (CAI) atau perangkat ajar adalah suatu cara belajar yang berbasiskan komputer yang berisi program yang interaktif sehingga membuat proses belajar-mengajar menjadi lebih efektif.

2.4.3 Tujuan Perangkat Ajar

Pada dasarnya tujuan pengajaran berbasiskan komputer adalah mencapai cara belajar yang aktif bagi pemakainya seperti peningkatan hasil belajar dan mengajar, dan efisiensi pengguna sumber daya yang terbatas yaitu sumber daya waktu dan peralatan.

Menurut Kearsley (1988, p2), terdapat 10 tujuan yang ingin dicapai melalui perangkat ajar, yaitu:

1. Peningkatan pengawasan.
2. Pengurangan kebutuhan sumber daya, terutama manusia.
3. Individualisasi.
4. Ketepatan waktu dan tingginya tingkat ketersediaan.
5. Pengurangan waktu pelatihan.
6. Perbaikan kinerja kerja.

7. Kenyamanan penggunaan.
8. Alat pengubah cara belajar.
9. Peningkatan kepuasan belajar.
10. Pengurangan waktu pengembangan.

2.4.4 Komponen-komponen Dasar Perangkat Ajar

Dalam CAI atau perangkat ajar ada empat komponen dasar yang terlibat dalam pembuatan dan pemakaian CAI yaitu:

1. Perangkat Keras (Hardware)

Komponen ini merupakan alat fisik yang berhubungan langsung antara pengguna dengan sistem pengajaran berbasis komputer seperti CPU, monitor, mouse, keyboard, printer dan sebagainya.

2. Perangkat Lunak (Software)

Komponen ini meliputi semua program dalam sistem untuk mengoperasikan dan melakukan fungsi-fungsi instruksional.

3. Perangkat Ajar (Courseware)

Komponen ini merupakan bagian dari piranti lunak yang merupakan program yang melengkapi persentasi instruksional.

4. Manusia (Brainware)

Komponen ini meliputi orang-orang yang memiliki keahlian khusus dalam mengembangkan, memelihara dan mengevaluasi suatu sistem pengajaran berbasis komputer.

2.4.5 Jenis-jenis Perangkat Ajar

Menurut Kearsley (1983, p30-36), ada 3 jenis perangkat ajar (CAI), yaitu:

1. *Drill and Practice*

Jenis CAI ini merupakan jenis yang termudah dalam penggunaan komputer untuk instruksi. Cara ini terdiri dari menampilkan sebuah pertanyaan atau masalah kemudian terima masukan dari pengguna. Lalu program akan memeriksa jawaban atau tanggapan.

Drill and Practice membantu siswa praktek beberapa ketrampilan atau konsep yang telah dipelajari sebelumnya. Setiap program menyimpan materi yang serupa yang ditemukan dibuku. Keuntungan CAI jenis ini adalah komputer dapat menyediakan *feedback* secara cepat dengan respon yang benar (Steward, 1987, p191).

Dari landasan diatas dapat diketahui bahwa perangkat ajar *Drill and Practice* dapat membantu pelajar berlatih dengan beberapa keahlian atau konsep-konsep yang telah diajarkan sebelumnya.

2. Tutorial

Berbeda dengan jenis *Drill and Practice*, jenis Tutorial ini melibatkan representasi informasi yang berhubungan. Pada tutorial akan menampilkan informasi-informasi sebagai konsep prosedur-prosedur tertentu diikuti dengan sejenis pertanyaan atau masalah. Tutorial merupakan perangkat ajar dengan menggunakan komputer sebagai pengajar materi pelajaran yang diprogramkan secara lengkap untuk mensimulasikan interaksi antara pengajar dan pelajar di dalam ruang kelas.

3. Socratic

Jenis ini berisi percakapan antara komputer dengan peserta pelatihan dalam *natural language* dan banyak digunakan di bidang *artificial intelegence* (Kuntjahjani, 2000, p90). Socratic cenderung berasal dari penelitian bidang inteligensi semu (*Artificial Intelegence*), yakni berupa sistem perangkat ajar yang dapat mendiagnosa input yang diterimanya.

Terdapat 3 jenis *feedback* yang digunakan perancangan perangkat ajar berbasis komputer guna memberikan hasil interaksi yang maksimal, antara lain:

1. Umpan balik *right-wrong*

Merupakan jenis umpan balik yang muncul baik ketika pengguna perangkat ajar memberikan respon yang benar maupun yang salah.

2. Umpan balik *right-blank*

Merupakan jenis umpan balik yang hanya muncul ketika pengguna perangkat ajar memberikan respon yang benar saja.

3. Umpan balik *wrong-blank*

Merupakan jenis umpan balik yang hanya muncul ketika pengguna perangkat ajar memberikan respon yang salah saja.

2.5 Microsoft DirectX

Microsoft DirectX adalah suatu set program aplikasi *low level* yang untuk menciptakan *game* dan aplikasi multimedia berperforma tinggi. DirectX

mendukung fasilitas grafik dua dimensi (2-D) dan tiga dimensi (3-D), efek suara dan musik, *input device*, dan aplikasi jaringan seperti *multiplayer games*.

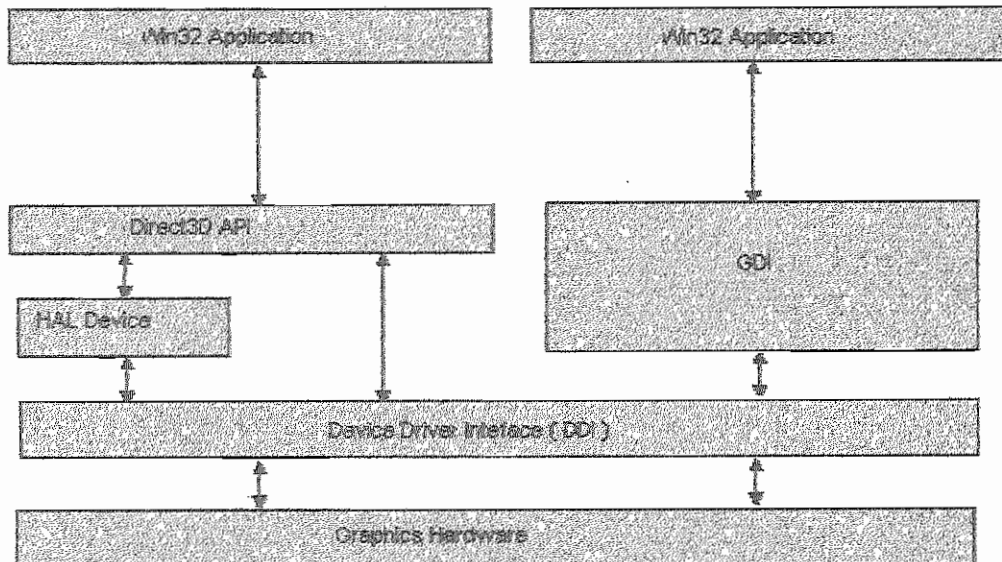
2.5.1 Komponen Microsoft DirectX 8.0

Komponen yang terdapat pada Microsoft DirectX 8.0 adalah sebagai berikut :

1. Komponen grafik DirectX yang menggabungkan Microsoft DirectDraw dan Microsoft Direct3D untuk dipergunakan ke dalam satu API (*Application Programming Interface*) yang dapat dipergunakan untuk semua program grafik.
2. Komponen audio DirectX yang menggabungkan Microsoft DirectSound dan Microsoft DirectMusic yang dipergunakan untuk program audio.
3. Komponen DirectInput yang menyediakan fasilitas masukan seperti keyboard, mouse, joystick, dsb.
4. Komponen DirectPlay yang menyediakan fasilitas *multiplayer games* pada jaringan.
5. Komponen DirectShow yang menyediakan fasilitas capture dan playback dari multimedia streams.
6. Komponen DirectSetup yaitu program API sederhana yang menyediakan fasilitas instalasi untuk semua komponen DirectX.

2.5.2 Penggabungan sistem

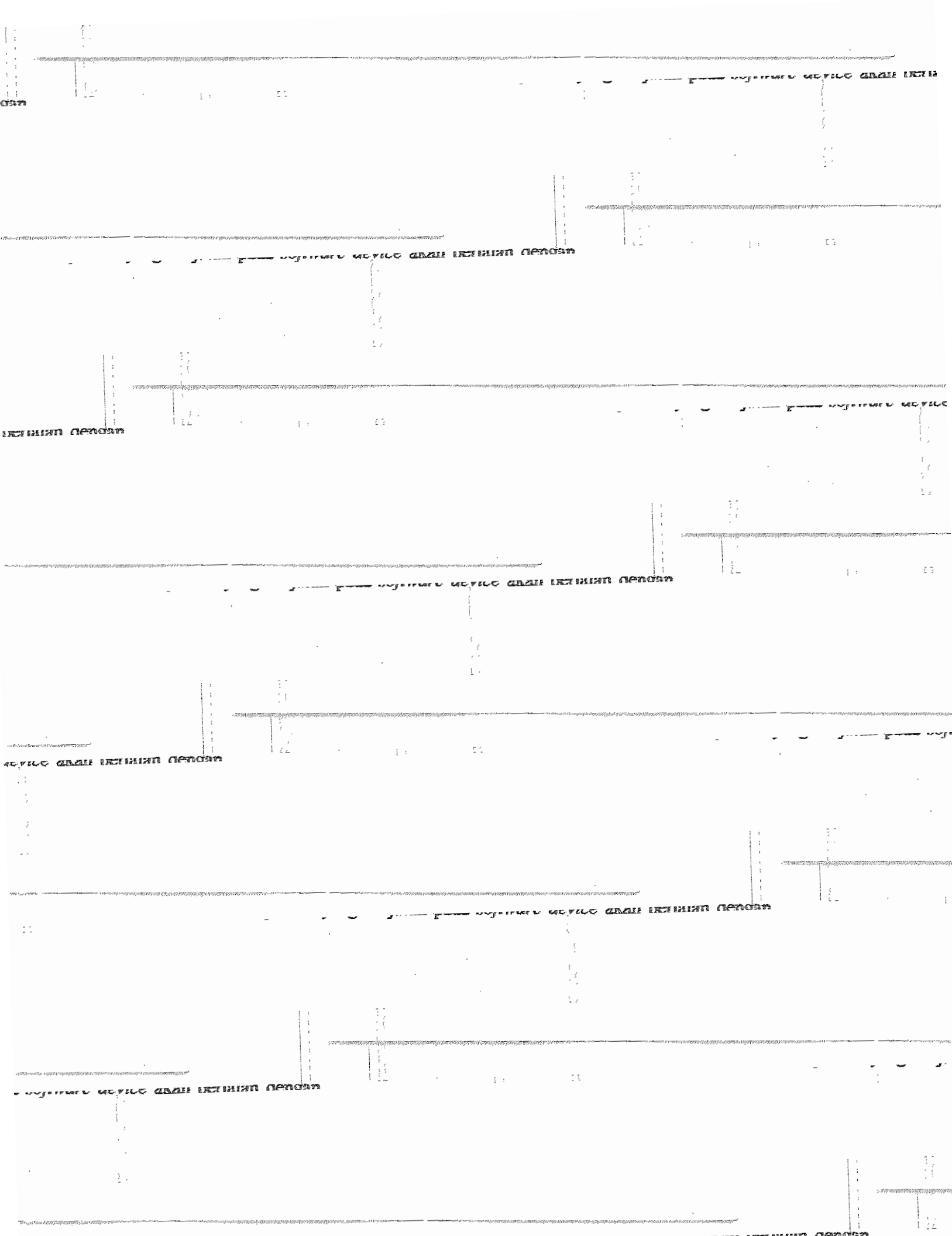
Berikut adalah diagram yang menunjukkan hubungan antara Microsoft Direct3D, *Graphics Device Interface* (GDI), *Hardware Abstraction Layer* (HAL), dan *Hardware*:



Gambar 2.4 Hubungan Direct3D, GDI, HAL, dan *Hardware*

Seperti pada gambar diatas, aplikasi Direct3D berada diluar aplikasi GDI, dan keduanya mempunyai akses terhadap perangkat grafik melalui *device driver* dari kartu grafis. Tidak seperti GDI, Direct3D dapat mengambil keuntungan dari fitur hardware bila HAL diaktifkan. HAL menyediakan akselerasi hardware berdasarkan fitur yang didukung oleh kartu grafis yang akan dapat mempercepat proses perhitungan grafik.

Software device, meskipun tidak mendukung seluruh fitur dari DirectX 8.0, dimaksudkan agar aplikasi akan selalu mencari *device* yang aktif pada semua sistem, sehingga sistem dapat berjalan. Pada beberapa

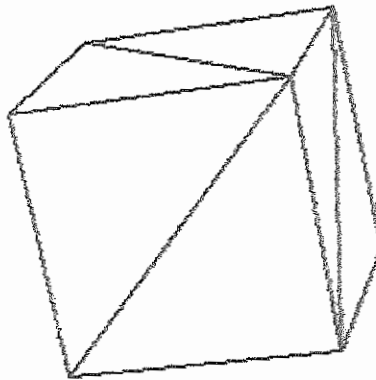


positif yang mengarah ke atas atau ke bawah dan sumbu X positif yang mengarah ke kiri atau ke kanan.

2.5.4 Koleksi-Koleksi 3-D

Objek merupakan koleksi titik-titik koordinat dari suatu sistem koordinat 3-D. Suatu cara yang paling mudah untuk membuat koleksi-koleksi titik-titik koordinat tersebut adalah dengan menggunakan *point list*. *Point list* memuat titik-titik koordinat 3-D yang dapat dihubungkan dalam membentuk sebuah objek.

Kondisi 3-D sering disebut dengan *polygon*. *Polygon* adalah kumpulan titik-titik koordinat 3-D yang mempunyai anggota paling sedikit 3 titik. *Polygon* yang paling sederhana adalah *triangle* (segitiga). Microsoft Direct3D menggunakan sistem *triangle* untuk menggambarkan *polygon*.



Gambar 2.6 Contoh *Polygon-Polygon* Pada Kubus

Gambar diatas menunjukkan sebuah contoh sistem koordinat yang membentuk sebuah kubus. Kubus tersebut digambarkan dengan 2 *triangle* pada setiap sisinya. Gambar tersebut merupakan rangka dari suatu gambar

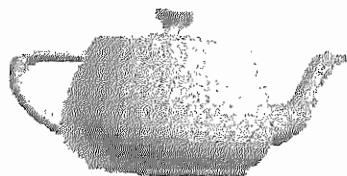
3-D. Kita dapat menambahkan tekstur dan material pada sisi kubus tersebut agar terlihat padat.

2.5.5 Metode *Shading*

Shading digunakan untuk melakukan *render* terhadap *polygon* sehingga akan tampak efek kedalaman dari suatu objek. Proses *shading* akan menentukan intensitas warna dan pencahayaan terhadap sembarang titik pada sisi *polygon*. Microsoft Direct3D mendukung 2 teknik *shading*, yaitu:

1. *Flat Shading*

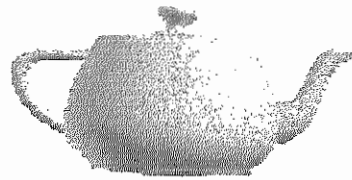
Pada *flat shading*, Direct3D akan me-render sebuah *polygon* menggunakan warna material pada titik pertama sebuah *polygon* untuk mewarnai seluruh titik-titik pada *polygon* itu. Objek 3-D yang di-render dengan menggunakan *flat shading* akan mempunyai tepi yang tajam diantara *polygon-polygon*-nya. *Flat Shading* merupakan teknik *shading* yang paling sedikit dalam melakukan perhitungan, sehingga proses akan menjadi lebih cepat.



Gambar 2.7 Contoh Hasil *Shading* Dengan Metode *Flat Shading*

2. Gouraud Shading

Pada *Gouraud shading*, Direct3D akan menghitung sebuah warna pada setiap titik dengan menggunakan titik normal dan efek pencahayaan. Setelah itu akan menyisipkan warna pada titik-titik disampingnya sepanjang permukaan *polygon*. Misalnya, komponen warna merah pada titik 1 adalah 0.8 dan komponen merah pada titik 2 adalah 0.4, menggunakan *gouraud shading* dan model pewarnaan RGB, modul pencahayaan akan memberikan nilai 0.6 kepada pixel tengah terhadap garis diantara kedua titik itu. Metode *Gouraud shading* akan memberikan permukaan objek menjadi lebih halus.



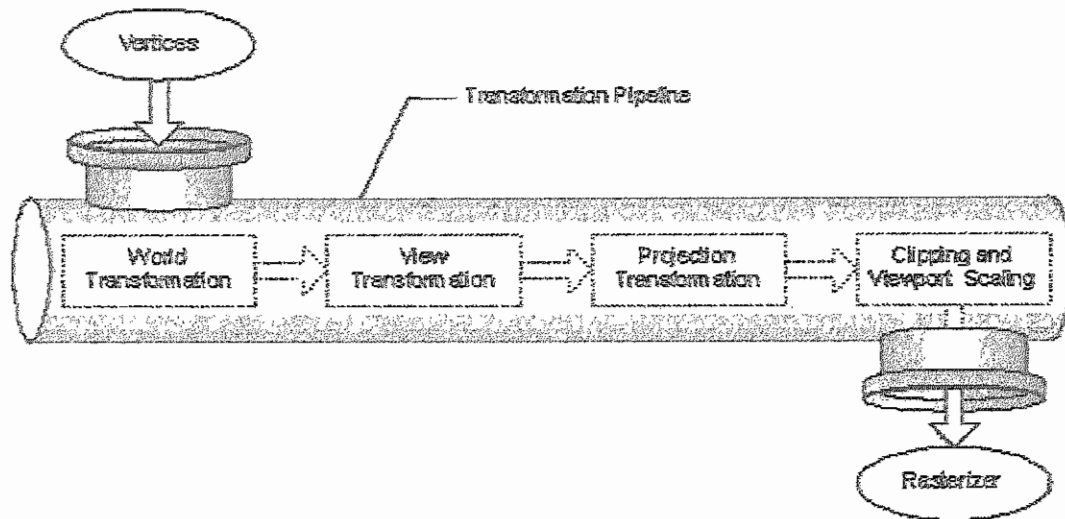
Gambar 2.8 Contoh Hasil *Shading* Dengan Metode *Gouraud Shading*

2.5.6 Transformasi Grafik

Transformasi grafik akan menentukan model dan pandangan terhadap semesta, proyeksi titik-titik untuk ditampilkan ke layar, dan *clipping* pada *viewport*. Transformasi juga menentukan perhitungan pencahayaan pada setiap titik.

Perhitungan geometri pada transformasi dilakukan dari pengambilan inputan titik-titik. Proses transformasi mencakup 3 tipe

transformasi yaitu: *world transformation*, *view transformation*, dan *projection transformation*. Proses selanjutnya adalah proses *clipping* dan meneruskan semuanya ke *rasterizer*. Untuk lebih jelasnya diilustrasikan dengan gambar berikut:



Gambar 2.9 Proses Transformasi Pada Microsoft Direct3D 8.0

Pada tahapan awal transformasi, tidak ada transformasi yang dilakukan, semua model titik-titik dideklarasikan sebagai koordinat lokal yang merupakan koordinat asli dari objek 3D.

Pada tahapan selanjutnya akan dilakukan proses transformasi terhadap titik-titik dari koordinat awalnya terhadap sistem koordinat semesta. Proses transformasi yang menggabungkan semua koordinat objek terhadap koordinat semesta disebut dengan *world transformation*.

Pada tahapan selanjutnya titik-titik yang ada pada koordinat semesta akan diorientasikan terhadap sistem kamera. Pada tahapan inilah, aplikasi akan menentukan *point-of-view* terhadap layar, koordinat semesta, yang dapat dilokasikan dan diputar keliling dari pandangan kamera.

Tahapan selanjutnya adalah *projection transformation*. Pada tahapan ini, objek biasanya akan diperbesar mengikuti jarak objek tersebut dari pengamat untuk memberikan kesan kedalaman pandangan pada layar. Objek yang lebih dekat akan dibuat terlihat menjadi lebih besar daripada objek yang lebih jauh.

Pada tahap akhir, semua titik-titik yang berada dibelakang objek dan tidak akan terlihat oleh pengamat akan dihilangkan, sehingga proses perhitungan tidak akan menghabiskan waktu untuk menghitung warna dan *shading* terhadap titik tersebut. Proses ini disebut dengan *clipping*. Setelah dilakukan proses *clipping*, akan tersisa titik-titik yang kemudian akan diperbesar dan akan dikonversikan menjadi koordinat layar yang merupakan koordinat 2D. Tahapan ini biasa disebut dengan *rasterization*.

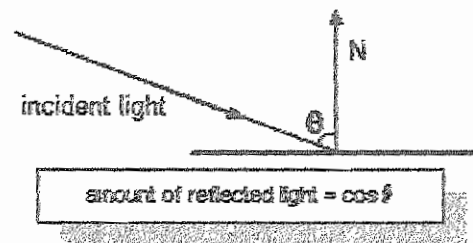
2.5.7 *Material dan Lighting*

Material menjelaskan bagaimana *polygon* memantulkan cahaya atau bagaimana suatu permukaan *polygon* bila memancarkan cahaya. Secara umum, *material* merupakan kumpulan *properties* yang menjelaskan kepada Direct3D bagaimana suatu *polygon* itu di-render.

Material properties mencakup pemantulan cahaya *diffuse*, cahaya *ambient*, *light emission*, dan *specular highlight*. Properti *material* ini mempengaruhi warna yang digunakan oleh Direct3D untuk melakukan rasterisasi *polygon*.

Pemantulan *diffuse* paling berpengaruh dalam menentukan warna. Pencahayaan *diffuse* adalah terarah, sudut datang cahaya untuk

pencahayaannya ini mempengaruhi intensitas pemantulan secara keseluruhan. Bila sudut datang cahaya bertambah, efek pemantulan cahaya *diffuse* menjadi berkurang. Besarnya cahaya yang dipantulkan adalah cosinus antara sudut datang dan normal, seperti ditunjukkan pada gambar dibawah ini:



Gambar 2.10 Pencahayaannya *Diffuse*

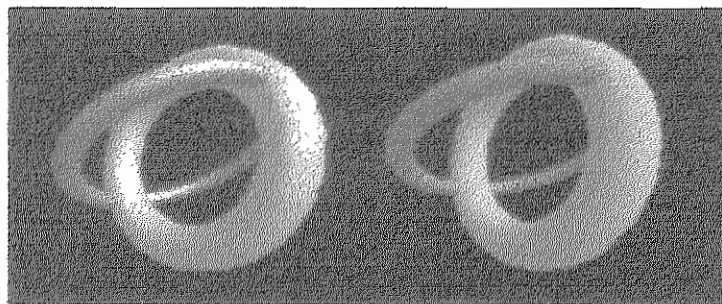
Pencahayaannya *ambient* tidak terarah. Pemantulan *ambient* kurang berpengaruh terhadap warna yang ditampilkan pada objek yang telah di-*render*, akan tetapi akan berpengaruh terhadap warna secara keseluruhan, dan akan sangat jelas terlihat ketika sedikit atau tidak ada sama sekali cahaya *diffuse* yang memantul pada *material*.

Pencahayaannya *diffuse* dan *ambient* bersama-sama menentukan warna yang spesifik terhadap suatu objek, dan biasanya keduanya merupakan warna yang identik. Sebagai contoh, untuk me-*render* objek yang berwarna biru, diciptakan suatu *material* yang memantulkan hanya komponen cahaya berwarna biru terhadap pencahayaannya *diffuse* atau *ambient*. Ketika objek tersebut diletakkan pada suatu ruangan dengan cahaya berwarna putih, objek tersebut akan memantulkan komponen cahaya berwarna biru, sehingga objek tersebut akan terlihat berwarna biru. Akan tetapi, di dalam suatu ruangan yang hanya mempunyai cahaya berwarna merah, objek

tersebut akan terlihat berwarna hitam, karena objek tersebut tidak dapat memantulkan cahaya berwarna merah.

Emission digunakan untuk menjadikan suatu objek yang di-render menjadi berkilau. Cahaya *emission* ini berpengaruh terhadap warna objek, misalnya, menjadikan objek yang gelap menjadi terang. Material dengan properti *emissive* tidak memancarkan cahaya yang dapat dipantulkan oleh objek lain di layar.

Pencahayaannya *specular* menciptakan cahaya sorot pada objek, menjadikan objek tampak berkilau. Intensitas yang tinggi dari *specular* membuat objek memiliki efek cahaya yang lebih tajam, sebaliknya intensitas yang lebih rendah akan membuat objek menjadi lebih gelap. Gambar berikut akan menunjukkan suatu objek yang di-render dengan menggunakan pencahayaannya *specular* dan yang tidak menggunakan pencahayaannya *specular*.



Gambar 2.11 Perbandingan Objek Dengan Pencahayaannya *Specular* dan Tanpa Pencahayaannya *Specular*

2.6 Taksonomi Tujuan Kognitif Menurut Bloom

Taksonomi Bloom mengelompokkan tujuan kognitif ke dalam enam kategori. Keenam kategori ini mencakup kompetensi keterampilan intelektual dari yang sederhana (tingkat pengetahuan) sampai dengan yang paling kompleks (tingkat evaluasi).

Keenam kategori ini diasumsikan bersifat hierarkis, yang berarti tujuan pada level yang tinggi dapat dicapai hanya apabila tujuan pada level yang lebih rendah telah dikuasai.

1. Pengetahuan/pengenalan

Tujuan instruksional pada level ini menuntut pelajar untuk mampu mengingat (*recall*) informasi yang telah diterima sebelumnya, seperti misalnya: fakta, *terminology*, rumus, strategi pemecahan masalah, dan sebagainya. Beberapa contoh kata kerja yang mewakili kelompok ini misalnya mengidentifikasi, memilih, menyebutkan nama, membuat daftar, dan lain-lain.

Contoh:

Pelajar dapat menyebutkan langkah-langkah prosedur pengukuran tingkat kontaminasi air (Lingkungan Hidup)

2. Pemahaman

Tujuan pada kategori ini berhubungan dengan kemampuan untuk menjelaskan pengetahuan/informasi yang telah diketahui dengan kata-kata sendiri. Dalam hal ini pelajar diharapkan untuk menerjemahkan atau menyebutkan kembali yang telah didengar dengan kata-kata sendiri. Kata kerja

dalam kelompok ini misalnya membedakan, menjelaskan, menyimpulkan, merangkumkan, memperkirakan, dan lain-lain.

Contoh:

Pelajar dapat membuat ringkasan sejarah pergerakan nasional untuk mencapai kemerdekaan (Sejarah Indonesia)

Kata kerja yang diperoleh harus operasional, dengan pengertian bahwa kompetensi dan perilaku tersebut dapat diukur unjuk kerjanya. Hal ini penting untuk menunjukkan apakah tujuan instruksional yang ditetapkan dapat tercapai atau tidak pada akhir pembelajaran.

Pemilihan kata kerja untuk dua tingkat kemampuan kognitif yang paling dasar, yaitu pengetahuan dan pemahaman sering terjebak pada kata-kata yang tidak operasional, misalnya “mengetahui” atau “memahami”. Kedua kata tersebut tidak dapat digunakan sebab dapat bermakna banyak.

Sebagai contoh seseorang yang “memahami” prinsip manajemen, dapat diartikan “mampu menjelaskan” (tingkat pemahaman) tentang manajemen atau bahkan “mampu menggunakan” (tingkat penerapan) prinsip-prinsip manajemen dalam berbagai kasus. Dua kemampuan tersebut levelnya jelas berbeda, hal ini selanjutnya akan mempengaruhi pemilihan strategi dan materi pembelajaran. Oleh sebab itu pilih kata kerja yang operasional yang lebih akurat merefleksikan tingkat kemampuan kognitif yang dimaksud.

3. Penerapan

Penerapan merupakan kemampuan untuk menggunakan atau menerapkan informasi yang telah dipelajari ke dalam situasi atau konteks yang lain atau yang baru. Sebagai contoh, menyusun kuesioner penelitian untuk penulisan skripsi merupakan penerapan prinsip-prinsip penyusunan instrument penelitian yang sebelumnya telah dipelajari pelajar dalam mata kuliah metode penelitian. Kata kerja yang dapat digunakan untuk tingkat penerapan umpamanya: menghitung, mengembangkan, menggunakan, memodifikasi, mentransfer, dan lain-lain.

Contoh:

Pelajar akan dapat menentukan apakah rata-rata (mean) 2 set data berbeda secara signifikan atau tidak (Statistik).

4. Analisis

Analisis merupakan kemampuan untuk mengidentifikasi, memisahkan dan membedakan komponen-komponen atau elemen suatu fakta, konsep, pendapat, asumsi, hipotesa atau kesimpulan dan memeriksa setiap komponen tersebut untuk melihat ada tidaknya kontradiksi. Dalam hal ini pelajar diharapkan untuk menunjukkan hubungan di antara berbagai gagasan dengan cara membandingkan gagasan tersebut dengan standar, prinsip atau prosedur yang telah dipelajari. Sebagai contoh, pembuatan kritik suatu karya literatur atau seni merupakan analisis. Tugas seperti ini memerlukan kemampuan analisis sebab menuntut pelajar untuk membuat tanggapan terhadap berbagai

aspek, seperti tema, plot, derajat realisme, dan sebagainya, serta melihat hubungan di antara aspek-aspek tersebut.

Contoh kata kerja tingkat analisis adalah membuat diagram, membedakan, menghubungkan, menjabarkan ke dalam bagian-bagian, dan lain-lain.

Contoh:

Pelajar akan dapat menjelaskan perbedaan demokrasi liberal dan demokrasi perwakilan, dengan melihat implikasi dalam sistem pemerintahan (Ilmu Politik).

5. Sintesis

Tujuan instruksional level ini menuntut pelajar untuk mampu mengkombinasikan bagian atau elemen ke dalam satu kesatuan atau struktur yang lebih besar. Menulis essay tentang "Perwujudan Bhinneka Tunggal Ika dalam masyarakat Indonesia" merupakan contoh sintesis. Dalam hal ini pelajar harus melihat berbagai aspek sosial, budaya dan ekonomi dalam kelompok etnik, misalnya sistem kekerabatan, sistem keagamaan dan sebagainya, dan kemudian membandingkan perwujudan berbagai aspek tersebut dan membuat kesimpulan. Contoh kata kerja operasional adalah menciptakan, mendesain, memformulasikan, membuat prediksi, dan lain-lain.

Contoh:

Pelajar mampu menciptakan tarian baru yang mengandung unsur-unsur gerak daerah Nusantara (Koreografi).

6. Evaluasi

Tujuan ini merupakan tujuan yang paling tinggi tingkatnya, yang mengharapkan pelajar mampu membuat penilaian dan keputusan tentang nilai suatu gagasan, metode, produk atau benda dengan menggunakan kriteria tertentu. Sebagai contoh, kemampuan mengevaluasi suatu program video apakah memenuhi syarat sebagai program instruksional yang baik atau tidak, merupakan tujuan tingkat evaluasi. Dalam hal ini pelajar harus mempertimbangkan dari segi isi, strategi presentasi, budaya, karakteristik pengguna, dan sebagainya. Di samping itu, kriteria program yang baik harus terlebih dahulu jelas bagi pelajar. Kata kerja operasional yaitu membuat kritik, membuat penilaian, membandingkan, membuat evaluasi, dan lain-lain.

Contoh:

Dengan menggunakan kriteria yang ditetapkan pelajar dapat membuat penilaian tentang efisiensi dan efektifitas suatu program Pendidikan Kesejahteraan Keluarga (Pendidikan Masyarakat).