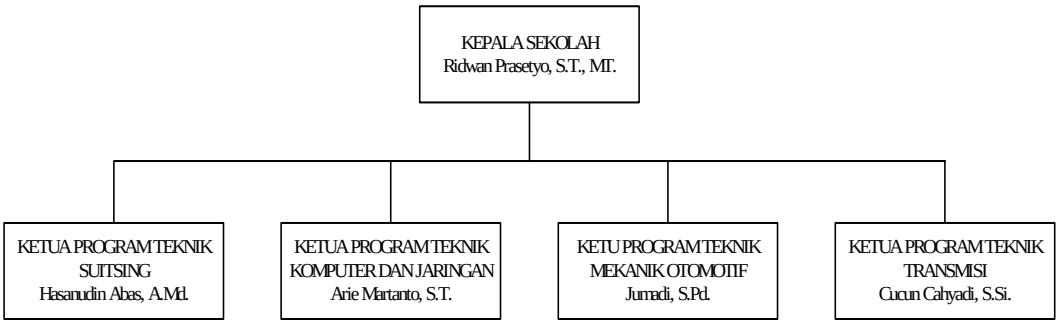
BAB II
TINJAUAN PUSTAKA

2.1 SMK Angkasa 1 Margahayu

Sekolah Menengah Kejuruan Angkasa 1 Margahayu adalah sekolah yang berstatus swasta yang bertempat di jalan Hercules No.4 Lanud Sulaiman Margahayu Bandung, sekolah yang terakreditasi B ini mempunyai program keahlian atau jurusan diantaranya:

1. Teknik Komputer dan Jaringan
2. Teknik Sutsing
3. Teknik Transmisi
4. Teknik Mekanik Otomotif.

Adapun struktur organisasi ketua dari berbagai program yang ada pada sekolah SMK Angkasa 1 Margahayu dapat dilihat pada Gambar 2.1 dibawah ini.



Gambar 2.1 Struktur Organisasi Ketua Program

2.1.1 Visi dan Misi SMK Angkasa 1

Adapun visi dari SMK Angkasa 1 Lanud Sulaiman adalah ”*Menjadikan sekolah bertarap nasional yang memiliki kualitas kompetensi dan marketable*”, sedangkan mempunyai misi yaitu :

1. Beriman dan bertakwa kepada tuhan yang maha esa
2. Menciptakan hasil didik yang siap pakai
3. Kreatif dan inofatif
4. Disiplin
5. Bersih.

2.1.2 Sejarah SMK Angkasa 1

SMK Angkasa 1 berdiri pada tanggal 16 Agustus 1968 dengan nama STM Elektronika Angkasa dengan jurusan pemeliharaan dan pembetulan dengan status swasta dibawah naungan TNI Angkatan Udara. SMK Elektronika sejak tahun 1972 di percaya oleh kanwil P dan K Jawa Barat untuk menyelenggarakan ujian sendiri yang semula berafiliasi ke STM 2 Bandung, hal ini merupakan kepercayaan penuh yang di idam-idamkan oleh sekolah-sekolah swasta.

Pada tahun 1979 STM Elektronika Angkasa mendapat peningkatan status dari swasta penuh ke swasta berbantuan P dan K dengan SK mentri P dan K No.63666/D/I/1979 tanggal 26 Juli dengn nam STM berbantuan Angkasa.

Pada tahun 1986 STM Elektronika Angkasa mendapat pengakuan lagi dengan jenjang akreditas DIAKUI sebagai sekolah menegah kejuruan tingkat tinggi atas (SMKTA) swasta.

Berdasarkan Direktur Jendral Pendidikan Dasar dan Menegah tanggal 6 Januari 1986 dengan No.001/C/Kep/L86 dengn data sekolah B.12284301, berubahlah nama menjadi SMK Angkasa Lanud Sulaiman Bandung. Surat keputusan ditandatangani langsu oleh Bapak Prof.DR.Hasan Walinono di jakarta pada tanggal 10 Februari 1986 sebagai Direktur Jendral Pendidikan Dasar dan Menengah.

Sistem disekolah kejuruan selalu diwarnai dengan adanya perubahan kurikulum, yang tentunya diselaraskan dengan kemajuan dan perkembangan ilmu pengetahuan dan teknologi. Oleh sebab itu untuk mengikuti perkembangan ilmu pengetahuan dan teknologi, pemerintah sesuai dengan wewenangnya telah mengganti kurikulum 1984 dengan kurikulum yang baru 1994, deng terjadinya kurikulum ini maka nama STM Elektronika Angkasa dirubah menjadi SMK Elektronika Angkasa.

Pada tahun 1996 SMK Elektronika Angkasa, dapat meningkatkan status dari diakui menjadi disamakan. Berdasarkan surat keputusan N0.37/C/Kep/WN/1996 tanggal 26 Maret 1996, ini mengambarkan bahwa kepercayaan masyarakat dan pemerintah terhadap SMK Elektronika Angkasa begitu besar.

Pada tahu 1997 SMK Elektronika Angkasa berubah nama menjadi SMK ANGKASA 1 MARGAHAYU berdasarkan surat edaran Dirjen DISDASMEN tanggal 20 Maret 1997 sebagai tindak lanjut dari keputusan mentri pendidikan dan kebudayaan No.036/0/1997 tentang perubahan SMKTA menjadi STM serta organisai dan tata kerja SMK.

2.2 Pengertian Komputer Sebagai Alat Bantu Pembelajaran

Kemajuan media komputer memberikan beberapa kelebihan untuk kegiatan produksi audio visual. Pada tahun-tahun belakangan komputer mendapat perhatian besar karena kemampuannya yang dapat digunakan dalam bidang kegiatan pembelajaran. Ditambah dengan teknologi jaringan dan internet, komputer seakan menjadi primadona dalam kegiatan pembelajaran.

Pembelajaran yang dibantu komputer dikenal dengan nama CAI yaitu "*Computer Assited Instruction*". Prinsip pembelajaran ini menggunakan komputer sebagai alat bantu menyampaikan pelajaran kepada user secara interaktif. Perubahan metode pembelajaran dan pengajaran telah menyebabkan alat yang digunakan menjadi meluas, misalnya: *video*, *audio*, *slide* dan *film*.

CAI yaitu penggunaan komputer secara langsung dengan siswa untuk menyampaikan isi pelajaran, memberikan latihan dan mengetes kemajuan belajar siswa. CAI juga bermacam-macam bentuknya bergantung kecakapan pendesain dan pengembang pembelajarannya, bisa berbentuk permainan (*games*), mengajarkan konsep-konsep abstrak yang kemudian dikonkritkan dalam bentuk *visual* dan *audio* yang dianimasikan. Jadi CAI adalah penggunaan komputer sebagai alat bantu dalam dunia pendidikan dan pengajaran. CAI membantu siswa memahami suatu materi dan dapat mengulang materi tersebut berulang kali sampai ia menguasai materi itu [6].

2.2.1 Penggunaan Komputer dalam Kegiatan Pembelajaran

Penggunaan komputer dalam kegiatan pembelajaran mempunyai beberapa tujuan diantaranya [6]:

1. Untuk Tujuan *Kognitif.* Komputer dapat mengajarkan konsep-konsep aturan, prinsip, langkah-langkah, proses, dan kalkulasi yang kompleks. Komputer juga dapat menjelaskan konsep tersebut dengan sederhana dengan penggabungan *visual* dan *audio* yang dianimasikan.
2. Untuk Tujuan *Psikomotor*. Dengan bentuk pembelajaran yang dikemas dalam bentuk *games* dan simulasi sangat bagus digunakan untuk menciptakan kondisi dunia kerja. Beberapa contoh program antara lain : simulasi pendaratan pesawat, simulasi perang dalam medan yang paling berat dan sebagainya.
3. Untuk Tujuan *Afektif.* Bila program didesain secara tepat dengan memberikan potongan klip suara atau *video* yang isinya menggugah perasaan, pembelajaran sikap/*afektif* pun dapat dilakukan mengunakan media komputer.

2.2.2 Keistimewaan Komputer dalam Proses Pembelajaran

Keistimewaan pemakaian komputer dalam proses pembelajaran (Muhamad Ikhsan, 2006) [6]:

1. Komputer bisa mengajar secara individual (individualisasi dalam proses pembelajaran) kecepatan bisa sesuaikan dengan kemampuan siswa, metode/strategi belajar yang lebih tepat, penyesuaian isi materi dan tingkat kesukaran.

2. Bisa digunakan kapan saja (tidak terbatas waktu) dan bisa digunakan dimana saja (tidak terbatas ruang).
3. Hilangkan rasa malu takut.

2.2.3 Bentuk Penerapan Aplikasi CAI dalam Pembelajaran

Adapun bentuk-bentuk penerapan aplikasi CAI (*Computer Assisted Instruction*) dalam pembelajaran sebagai berikut (*Heinich,Etal,* 1996) [6]:

1. *Drill and Practic*a.

a. Tujuan

Setelah menjalankan program *drill and practice* ini siswa akan lebih terampil, cepat, dan tepat dalam melakukan suatu keterampilan. Misalnya keterampilan mengetik, atau menjawab soal hitungan.

b. Isi

Disini siswa dianggap sudah mengetahui teori yg mendasari keterampilan itu serta mengetahui cara/prosedur mengerjakannya. Jadi dalam *drill and practice* tidak ada bagian penjelasan, yang ada hanya sejumlah soal/pertanyaan dan "*feedback*". Soal atau pertanyaan tersebut diberikan dalam suatu urutan/alur ("*sequence*") tertentu : mudah dan sulit, siswa menjawab dinilai dan *feedback* benar dan salah soal/pertanyaan berikut dan seterusnya.

Umumnya kontrol yang dimiliki siswa sangat terbatas. Ia hanya dapat memilih tingkat kesulitan materi, sedangkan alur dari penyajian isi di kontrol oleh sistem. Variabel yang digunakan : tingkat kesulitan isi/materi, kecepatan menjawab, atau waktu menjawab.

2. *Tutorial*

a. Tujuan

Membuat siswa memahami suatu konsep atau materi yang baku.

b. Isi

Sejumlah konsep atau materi yang perlu diajarkan dan difahami siswa. Kemudian diikuti dengan sejumlah pertanyaan, atau latihan soal untuk memeriksa pemahaman siswa terhadap konsep atau materi tersebut. Siswa berinteraksi dengan komputer seperti ia berinteraksi dengan guru: "*one-to-one session*". Bila materi yang akan diberikan cukup banyak, maka penyajiannya akan diberikan secara bertahap, mulai dari materi dasar ke tingkat yg lebih tinggi, dan seterusnya. Selain itu ada pula sejumlah pertanyaan atau soal yg pemunculannya dibuat random. Bila siswa gagal melewati kriteria untuk "lulus", maka ia akan dikembalikan ke bagian penjelasan konsep atau materi yang pertama. Akan tetapi bila sistem ini disertai dengan modul "*remedial*", maka bila gagal, siswa akan diberikan *remedial* terhadap topik yang ia salah saja (tidak mengulang semua).

c. Keuntungan

Lebih *individualized* dari *drill and practice* ada penilaian terhadap respon, serta dapat dibantu bagian yang tidak difahami, mengulang materi, atau ke modul *remedial*.

3. *Games (Edutainment*)

Materi atau konteks dari permainan merupakan hal yang ingin diajarkan, sekaligus ia juga berperan sebagai *motivator*. Pendekatan motivasi, dibedakan diantaranya : Motivasi *intrinsik*, tidak ada reward diluar atau tanpa reward seperti "point" misalnya, anak menyenangi permainan tersebut. Motivasi *ekstrinsik*, ada *reward* dari luar, misalnya uang, atau "*point*". . Menimbulkan motivasi intrinsik harus ada tiga hal :

a. *Challenge* : Goal dari permainan harus jelas. Selain itu hasil/konsekwensi yang dapat dicapai akibat dari aksi/response pemain sulit untuk diterka semacam ada unsur luck. Tidak diketahui cara/strategi yg paling optimal.
b. *Fantasy* : Adanya situasi permainan yang merangsang munculnya imaginasi pemain.
c. *Curiosity* : Ada unsur yang "baru" (*novelty*) bagi pemain, akan jangan terlalu banyak hal "barunya" sebab akan menyebabkan permainan sukar dimengerti.

4. *Mindtools*

Mindtools alat bantu belajar yang menyediakan sejumlah fasilitas atau fungsi yang dapat dipakai untuk digunakan siswa dalam memfungsikan cara berpikirnya sehingga dapat optimal. Lingkungan pembelajaran yang disajikan pada siswa bukan berpatokan pada membuat siswa menurut saja pada struktur materi yang sudah dirancang alurnya oleh programmer, akan tetapi justru hanya memberikan sejumlah fasilitas atau alat (*tools*) untuk digunakan siswa dalam ia mengambil dan merancang alur belajarnya sendiri.

Kontrol penuh ada di tangan siswa (*learner control*) dalam ia menentukan baik tujuan yang ingin dicapai, materi yang dipelajari, maupun tingkat kedalaman pemahaman yang ingin diraih. dan akan lebih memotivasi siswa untuk belajar karena ia dapat sesuaikan dengan kebutuhannya. Guru disini berperan sebagai fasilitator, model, dan pelatih (*coach*).

Berangkat dari asumsi dasar bahwa siswa itu mempunyai perbedaan dalam daya tangkap, lingkup pengetahuan yang sudah dimiliki (*prior knowledge*), keterampilan belajar, minat, maupun motivasi untuk belajar.

Belajar yang dalam (*deep learning*) menuntut siswa menggunakan teknik/strategi berpikir yang sistematis dan terencana, tajam daya analisanya, kritis, kreatif, dan memiliki ketrampilan memecahkan masalah (*problem solving*) yang baik.

Keterampilan berfikir (ketrampilan belajar) adalah ketrampilan yang harus dengan sengaja dipelajari, bukan bersifat bawaan lahir (seperti halnya inteligensi). Keterampilan belajar inilah yang menjadikannya *self-regulated* (*directed*) *learner*.

5. *Simulation*

a. Tujuan

Proses simulasi biasanya digunakan untuk mengajarkan proses atau konsep yang tidak secara mudah dapat dilihat (*abstrak*), seperti bagaimana bekerjanya proses ekonomi, atau bagaimana hubungan antara *supply and demand* terhadap harga dan seterusnya. Simulasi juga dilakukan untuk memunculkan suatu keadaan yang berbahaya dan dicobakan di dunia riel. Misalnya percobaan percampuran berbagai zat kimia, atau perputaran planet. Umumnya

setelah siswa mencoba sendiri, atau menjalankan simulasi ini, guru harus memeriksa kesimpulan (*discovery*) yang dibuat siswa.

b. Isi

Suatu konsep atau keadaan yang akan dieksplorasi proses perubahan atau terjadinya. Siswa akan diberikan sejumlah variabel (beserta parameternya) yang dapat di mainkan atau manipulasi untuk menimbulkan keadaan tertentu. Asumsi dasar dari proses belajar disini adalah melalui percobaannya siswa akan mengerti prinsip dari terjadinya proses tersebut (*discovery learning*).

c. Keuntungan

Berlangsungnya proses dapat diatur kecepatannya; dapat dipercepat (untuk proses yang perubahannya lama), atau diperlambat (untuk proses yang perubahannya terjadi cepat). Variabel-variabel yang berpengaruh terhadap proses perubahan dapat secara langsung dimainkan, atau di manipulasi siswa sehingga ia dapat melihat langsung bagaimana pengaruh setiap variabel itu. Umumnya siswa akan lebih termotivasi menjalankan simulasi dibanding *drill and practice* atau *tutorial*, karena siswa punya kontrol terhadap variabel yang dipilihnya rasa ingin tahunya terpenuhi.

2.3 Pengertian Kamus

Kamus adalah sejenis buku rujukan yang menerangkan makna kata-kata. Ia berfungsi untuk membantu seseorang mengenal perkataan baru. Selain menerangkan maksud kata, kamus juga mungkin mempunyai pedoman sebutan, asal-usul (etimologi) sesuatu perkataan dan juga contoh pengunaan bagi sesuatu perkataan. Untuk memperjelas kadang kala terdapat juga ilustrasi di dalam kamus. Kata kamus diserap dari bahasa Arab *qamus*, dengan bentuk jamaknya *qawamis*. Kata Arab itu sendiri berasal dari kata Yunani *okeanos* yang berarti 'lautan'. Sejarah kata itu jelas memperlihatkan makna dasar yang terkandung dalam kata kamus, yaitu wadah pengetahuan, khususnya pengetahuan bahasa, yang tidak terhingga dalam dan luasnya. Dewasa ini kamus merupakan khazanah yang memuat perbendaharaan kata suatu bahasa, yang secara ideal tidak terbatas jumlahnya. Setiap kebudayaan besar di dunia bangga akan kamus bahasanya. Dalam kenyataannya kamus itu tidak hanya menjadi lambang kebanggaan suatu bangsa, tetapi juga mempunyai fungsi dan manfaat praktis [10].

2.3.1 Sejarah Kamus

Leksikografi adalah bidang ilmu bahasa yang mengkaji cara pembuatan kamus. Sebagian besar (atau bahkan semua) sarjana memiliki kamus, namun mereka belum tentu tahu bahwa penulisan kamus yang baik harus melalui berbagai proses.

Dua nama besar yang mengawali penyusunan kamus adalah Samuel Johnson (1709-1784) dan Noah Webster (1758-1843). Johnson, ahli bahasa dari Inggris, membuat *Dictionary of the English Language* pada tahun 1755, yang terdiri atas

dua volume. Di Amerika, *Webster* pertama kali membuat kamus *An American Dictionary of the English Language* pada tahun 1828, yang juga terdiri atas dua volume. Selanjutnya, pada tahun 1884 diterbitkan *Oxford English Dictionary* yang terdiri atas 12 volume.

Saat ini, kamus umum yang cukup luas digunakan adalah *Oxford Advanced Learner's Dictionary*. Mengapa kamus *Oxford?*. Beberapa orang mungkin secara sederhana akan menjawab karena kamus tersebut lengkap dan cukup mudah dimengerti. Tidak banyak yang tahu bahwa (setelah tahun 1995) kamus tersebut ditulis berdasarkan hasil analisis *British National Corpus* yang melibatkan cukup banyak ahli bahasa dan menghabiskan dana universitas dan dana negara yang jumlahnya cukup besar. Secara umum, definisi yang diberikan dalam kamus tersebut seharusnya dapat mudah dipahami oleh pelajar karena semua entri dalam kamus tersebut hanya didefinisikan oleh sekelompok kosa kata inti. Bagaimana kosa-kata inti tersebut disusun? Tentu hanya ahli bahasa yang dapat menjelaskannya, sedangkan para sarjana dan pelajar dapat langsung saja menikmati dan menggunakan berbagai kamus *Oxford* yang ada dipasaran [10].

2.3.2 Jenis-jenis Kamus

Kamus dapat dibedakan dan dapat dilihat dari jenisnya berdasarkan diantaranya [10]:

1. Berdasarkan penggunaan bahasa

Kamus boleh ditulis dalam satu atau lebih daripada satu bahasa. Dengan itu kamus boleh dibagi menjadi beberapa jenis yaitu:

a. Kamus Ekabahasa

Kamus ini hanya menggunakan satu bahasa. Kata-kata (entri) yang dijelaskan dan penjelasannya adalah terdiri daripada bahasa yang sama. Kamus ini mempunyai perbedaan yang jelas dengan kamus dwibahasa kerana penyusunan dibuat berasaskan pembuktian data korpus. Ini bermaksud definisi makna ke atas kata-kata adalah berdasarkan makna yang diberikan dalam contoh kalimat yang mengandung kata-kata berhubungan. Contoh bagi kamus ekabahasa ialah *Kamus Besar Bahasa Indonesia* (di Indonesia) dan *Kamus Dewan* di (Malaysia).

b. Kamus Dwibahasa

Kamus ini menggunakan dua bahasa, yakni kata masukan daripada bahasa yang dikamuskan diberi padanan atau pemerian takrifnya dengan menggunakan bahasa yang lain. Contohnya: *Kamus Inggris-Indonesia, Kamus Dwibahasa Oxford Fajar (Inggeris-Melayu;Melayu-Inggeris)*

c. Kamus Aneka Bahasa

Kamus ini sekurang-kurangnya menggunakan tiga bahasa atau lebih. Misalnya, kata Bahasa Melayu Bahasa Inggris dan Bahasa Mandarin secara serentak. Contoh bagi kamus aneka bahasa ialah *Kamus Melayu-Cina-Inggeris Pelangi* susunan Yuen Boon Chan pada tahun 2004.

2 Berdasarkan isi

Kamus boleh muncul dalam perbagai isi. Ini adalah karena kamus diterbitkan dengan tujuan memenuhi keperluan gologan tertentu. Contohnya, golongan pelajar sekolah memerlukan kamus berukuran kecil untuk memudahkan

mereka membawa kamus ke sekolah. Secara umumnya kamus boleh dibagi kepada 3 jenis ukuran:

a. Kamus Mini

   Pada zaman sekarang sebenarnya susah untuk menjumpai kamus ini.Ia juga dikenali sebagai kamus saku karena ia dapat disimpan dalam saku. Tebalnya kurang daripada 2 cm.

b. Kamus Kecil

   Kamus berukuran kecil yang biasa dijumpai. Ia merupakan kamus yang mudah dibawa.*Kamus Dwibahasa Oxford Fajar (Inggeris-Melayu;Melayu-Inggeris)*

c. Kamus Besar

   Kamus ini memuatkan segala leksikal yang terdapat dalam satu bahsaa. Setiap perkataannya dijelaskan maksud secara lengkap.Biasanya ukurannya besar dan tidak sesuai untuk dibawa ke sana sini.Contohnya *Kamus Besar Bahasa Indonesia*.

3. Berdasarkan keistimewaan dari kamus

Kamus istimewa merujuk kepada kamus yang mempunyai fungsi yang khusus. Contohnya :

a. Kamus Istilah

   Kamus ini berisi istilah-istilah khusus dalam bidang tertentu. Fungsinya adalah untuk kegunaan ilmiah. Contohnya ialah *Kamus Istilah Fiqh.*

b. Kamus Etimologi

   Kamus yang menerangkan asal usul sesuatu perkataan dan maksud asalnya.

c. Kamus Tesaurus (perkataan searti)

   Kamus yang menerangkan maksud sesuatu perkataan dengan memberikan kata-kata searti (sinonim) dan dapat juga kata-kata yang berlawanan arti (antonim). Kamus ini adalah untuk membantu para penulis untuk meragamkan penggunaan diksi. Contohnya, *Tesaurus Bahasa Indonesia.*

d. Kamus Peribahasa / Simpulan Bahasa

   Kamus yang menerangkan maksud sesuatu peribahasa/simpulan bahasa. Selain daripada digunakan sebagai rujukan, kamus ini juga sesuai untuk dibaca dengan tujuan keindahan.

e. Kamus Kata Nama Khas

   Kamus yang hanya menyimpan kata nama khas seperti nama tempat, nama tokoh, dan juga nama bagi institusi. Tujuannya adalah untuk menyediakan rujukan bagi nama-nama ini.

f. Kamus Terjemahan

   Kamus yang menyedia kata searti bahasa asing untuk satu bahasa sasaran. Kegunaannya adalah untuk membantu para penerjemah.

g. Kamus Kolokasi

   Kamus yang menerangkan tentang padanan kata, contohnya kata 'terdiri' yang selalu berpadanan dengan 'dari' atau 'atas'.

2.4 Pengetian Komputer

Kata komputer berasal dari bahasa Latin yaitu *Computare* yang artinya *menghitung*. Dalam bahasa Inggris disebut *to compute.* Secara definisi komputer diterjemahkan sebagai sekumpulan alat elektronik yang saling bekerja sama, dapat menerima data (*input*), mengolah data (*prosses)* dan memberikan informasi (output) serta terkoordinasi dibawah kontrol program yang tersimpan di memori [10].

2.5 Pengertian Informatika

Informatika (Inggris:*Informatics*) mencakup beberapa macam bidang, termasuk di dalamnya : ilmu komputer, ilmu informasi , sistem informasi, teknik komputer dan aplikasi informasi dalam sistem informasi manajemen. Secara umum informatika mempelajari struktur, sifat, dan interaksi dari beberapa sistem yang dipakai untuk mengumpulkan data, memproses dan menyimpan hasil pemrosesan data, serta menampilkannya dalam bentuk informasi. Aspek dari informatika lebih luas dari sekedar sistem informasi berbasis komputer saja, tetapi masih banyak informasi yang tidak dan belum diproses dengan komputer [10].

Informatika mempunyai konsep dasar, teori, dan perkembangan aplikasi tersendiri. Informatika dapat mendukung dan berkaitan dengan aspek kognitif dan sosial, termasuk tentang pengaruh serta akibat sosial dari teknologi informasi pada umumnya. Penggunaan informasi dalam beberapa macam bidang, seperti bioinformatika, informatika medis, dan informasi yang mendukung ilmu perpustakaan, merupakan beberapa contoh yang lain dari bidang informatika.

Dalam ruang lingkup yang lebih luas, informatika meliputi beberapa aspek diantaranya [10] :

a. Teori informasi yang mempelajari konsep matematis dari suatu informasi
b. Ilmu informasi yang mempelajari tentang cara pengumpulan, klasifikasi, manipulasi penyimpanan, pengaksesan, dan penyebarluasan informasi untuk keperluan sosial dan kemasyarakatan secara menyeluruh
c. Ilmu komputer dan teknik komputer yang mempelajari tentang pemrosesan, pengarsipan, dan penyebaran informasi dengan menggunakan teknologi informasi dan alat lain yang berbasis komputer.

2.6 Pengertian PHP

PHP adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memrogram situs web dinamis, walaupun tidak tertutup kemungkinam untuk digunakan untuk pemakaian lain. PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP bernama FI (Form Interpreted). Pada saat tersebut PHP adalah sekumpulan script yang digunakan untuk mengolah data form dari web.Perkembangan selanjutnya adalah Rasmus melepaskan kode sumber tersebut dan menamakannya PHP/FI, pada saat tersebut kepanjangan dari PHP/FI adalah *Personal Home Page/Form Interpreter*. Dengan pelepasan kode sumber ini menjadi *open source*, maka banya programmer yang tertarik untuk ikut mengembangkan PHP.Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini interpreter sudah diimplementasikan dalam C.

Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan [6].

Pada tahun 1997, sebuah perusahaan bernama *Zend*, menulis ulang interpreter PHP menjadi lebih bersih, lebih baik dan lebih cepat. Kemudian pada Juni 1998 perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan nama rilis tersebut menjadi PHP 3.0.Pada pertengahan tahun 1999, *Zend* merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai. Versi ini banyak dipakai sebab versi ini mampu dipakai untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan proses dan stabilitas yang tinggi.Pata Juni 2004, *Zend* merilis PHP 5.0. Versi ini adalah versi mutakhir dari PHP. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Dalam versi ini juga dikenalkan model pemrograman berorientasi objek baru untuk menjawab perkembangan bahasa pemrograman kearah pemrograman berorientasi objek.

PHP merupakan bahasa pemrograman web, penulisan script PHP dapat dijadikan satu dengan kode html dan disimpan dalam satu *file* yang sama ataupun berdiri sendiri dengan nama *file* berekstensi *.php. Ciri khas PHP diawali dengan tanda “<?php“ dan diakhiri dengan “?>”. PHP memiliki fasilitas untuk melakukan koneksi ke *database*. Script program PHP disimpan dan dijalankan di sisi *server* sehingga keamanan data lebih terjamin.

Contoh penggunaan script php sebagai berikut:

```
<? php
phpinfo ( );
?>
```

2.7 Pengertian MySQL

MySQL merupakan sebuah sistem manajemen *database open source* yang populer dan gratis untuk *platform UNIX*. Sistem manajemen *database MySQL* menggunakan kumpulan perintah sederhana untuk memasukkan, memanggil, menghapus, dan memperbarui data, dengan ini kita dapat mengembangkan *database* yang kompleks [13].

Beberapa kemampuan *MySQL* adalah sebagai berikut:

a. *MySQL* bisa diakses dan dimanipulasi dari sejumlah bahasa pemrograman terkenal, diantaranya adalah *C, C++, Java, Perl, Phyton,* dan *PHP*
b. *MySQL* mendukung tipe data yang umum digunakan, termasuk *FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, SET,* dan *ENUM*
c. *MySQL* memungkinkan alokasi *password* pada tiap *server* yang melalui *MySQL* untuk melakukan autentifikasi terenkripsi
d. *MySQL* mendukung berbagai macam metode koneksi, seperti TCP/IP, *socket UNIX*, dan koneksi untuk *Windows* NT/2000
e. *MySQL* bisa diperoleh secara gratis termasuk aplikasi-aplikasi lain yang diperlukan dalam memakai *MySQL*.

MySQL juga merupakan salah satu sistem manajemen *database* yang stabil di pasaran. Ketika *MySQL* diluncurkan pertama kali pada pertengahan 1996, beberapa *bug* dengan cepat dapat diketahui dan diperbaiki. Sekarang *MySQL* sudah menjadi sangat stabil dan mampu penyimpan data yang besar hingga sanggup menampung data lebih dari 4 Giga*byte*s. *Database MySQL* mendukung hampir seluruh bahasa pemrograman, termasuk PHP dan C.

2.7.1 Karakteristik MySQL

Berikut ini digambarkan beberapa karakteristik dari database MySQL [13]:

a. Dapat berintegrasi dengan C, C++, Eiffel, Java, Perl, PHP, Python and Tcl APIs.
b. Dapat bekerja pada beberapa platform OS yang berbeda
c. Memiliki banyak tipe kolom: *signed/unsigned integers* dengan panjang 1, 2, 3, 4, and 8 bytes, tipe: FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, and ENUM.
d. Mendukung penuh operator and fungsi untuk *query* SELECT dan WHERE
   Contoh :
   SELECT CONCAT (first_name," ",last_name)FROM tbl_name WHERE income/dependents > 10000 AND age > 30;
e. Fungsi-fungsi SQL diimplementasikan secara penuh pada *class library* yang teroptimalkan sehinggsa proses kerja sistem menjadi cepat, dan biasaanya tidak akan memakan alokasi memori setelah proses inisialisasi query
f. Mendukung penuh pada SQL *query* : GROUP BY dan ORDER BY. mendukung pada group fungsi (COUNT(), COUNT(DISTINCT), AVG(), STD(), SUM(), MAX() and MIN())
g. Mendukung penuh pada *relational databases*
h. Dapat mensupport ODBC *(Open-DataBase-Connectivity)* untuk Win32 (dengan *source*)

i. Mampu menangani sampai dengan 32 *index* per tablennya, Setiap *index* terdiri atas 1 sampai 16 colum atau bagian-bagian colum. Panjang Maksimal index sampai 500 *bytes*. *Indeks* bisa menggunakan *fields char* atau *varchar*

j. Setiap kolom memiliki nilai default

k. Mampu menempatkan alokasi memory dalam sistem dengan efisien

l. Memiliki fasilitas myisamchk, yang dapat melakukan *checking*, optimasi dan perbaikan dengan sangat cepat

m. Data-data tersimpan dalam suatu *character set* yang terpilih

n. *Aliases* pada table dan kolom dimungkinkan pada standar SQL92

o. Server dapat memberikan pesan error kepada *clients* pada berbagai macam bahasa.

Clients dapat tersambung ke MySQL server dengan menggunakan *TCP/IP Sockets, Unix Sockets (Unixes),* atau *Named Pipes (NT).*

2.7.2 Kelebihan MySQL

Dengan karakteristik MySQL tersebut diatas maka dapat memberikan kelebihan sebagai berikut [13]:

a. Menghemat waktu proses pengisian data

b. Menghemat waktu proses pengambilan data

c. Proses pengambilan data lebih fleksibel.

d. Data dapat diakses secara bersama oleh lebih dari satu pengguna pada waktu yang bersamaaan.

e. Kemampuan akses data di berbagai tempat dengan menggunakan transmisi elektronik.

2.8 Pengertian *Apache*

Apache merupakan web *server* yang paling sering digunakan sebagai *server* internet dibandingkan *web server* lainnya. *Web server* merupakan *server* internet yang mampu melayani koneksi transfer di dalam protokol *HTTP* (*Hypertext Transfer Protocol*) saat ini *web server* merupakan inti dari *server-server* internet selain *e-mail server*, *ftp server*, dan *news server*.

Web server sendiri dirancang untuk dapat melayani berbagai jenis data, diantaranya *text*, *hypertext*, gambar, suara dan bentuk file *HTML*. Dari file *HTML* kemudian dapat dikaitkan ke file *HTML* lainnya yang hendak dipublikasikan lewat internet. (ADA) [7].

2.9 Pengertian Model Huhungan Peer-to-Peer (P2P)

Jaringan komputer P2P termasuk sebuah cabang **(***subset***)** dari bidang komputasi terdistribusi. Namun komputasi terdistribusi sendiri bukanlah cabang dari P2P. Sebutan "*peer-to-peer*" mengisyaratkan sebuah hubungan kesetaraan (*egalitarian relationship*) diantara para *peer* (baca: pengguna satu dengan yang lainnya). Dan yang terpenting, hubungan ini haruslah menghasilkan interaksi langsung antara komputer pengguna yang satu dengan komputer pengguna lainnya. Tanpa embel-embel ada komputer yang berstatus sebagai *client* dan berstatus sebagai *server*.Secara teknis, jaringan P2P (*peer-to-peer*) adalah sebuah jaringan yang memungkinkan semua komputer dalam lingkungannya bertindak / berstatus sebagai server yang memiliki kemampuan untuk mendistribusikan

sekaligus menerima berkas-berkas atau sumber daya (*resource*) yang ada dalam komputer mereka ke komputer lainnya.

Jaringan bertipe ini sangat banyak dijumpai di kantor-kantor yang tidak membutuhkan sebuah sentral pengaturan laiknya jaringan *client-server*. Di internet, jaringan P2P hidup dan berkembang melalui aplikasi-aplikasi populer seperti Napster dan Gnutella [11].

2.9.1 Sejarah Singkat P2P

Tahun 1979, Usenet, sebuah aplikasi terdistribusi (baca: tidak tersentralisasi/ *distributed*) yang dibuat oleh Tom Truscott dan Jim Ellis, lahir di Amerika Serikat. Aplikasi ini umumnya melayani penggunanya dengan *newsgroup*. Pada tahun-tahun itu, dunia belum mengenal dan mampu menikmati layanan internet sebaik dan secepat seperti saat ini. Umumnya, berkas-berkas yang berada di dalam komputer milik pengguna *usenet* dipertukarkan dalam bentuk *batch files* (berkas yang berisi data yang diproses atau ditransmisikan mulai dari awal hingga akhir). Biasanya, para pengguna saat itu saling bertukar data di malam hari yang larut. Itu adalah waktu di sebuah negara besar ketika jalur telepon untuk SLJJ (sambungan langsung jarak jauh) sedang sepi. Akibatnya, tidak ada cara yang efektif untuk membuat fungsi aplikasi ini menjadi tidak terdistribusi. Dengan kata lain, aplikasi ini tetap menjadi aplikasi yang tidak memiliki pusat kendali (*server*). Bahkan hingga hari ini.

Aplikasi P2P generasi awal lain yang sukses dan populer adalah FidoNet. Laiknya Usenet, FidoNet juga digunakan secara terdistribusi. Aplikasi ini dibuat oleh Tom Jennings pada tahun 1984 sebagai cara untuk bertukar pesan diantara

pengguna-penggunanya yang memiliki BBS (Bulletin Board System) yang berbeda.

Baik Usenet maupun FidoNet dapat menjadi contoh betapa hebatnya teknologi P2P. Sampai detik ini, keduanya masih lestari. Uniknya, sekarang keduanya sudah tidak sendiri lagi. “Cucu-cucu” mereka sudah lahir dan ikut menggebrak dunia maya. Sebut saja Gnutella, Kazaa, Napster, dsb [11].

2.9.2 Manfaat Dan Kerugian P2P

Manfaat dari jaringan P2P agar semua *peer* dapat menyediakan sekaligus memanfaatkan *resource* komputer, termasuk *bandwith*, media penyimpanan, dan kemampuan komputasi yang ada di dalam jaringan tersebut. Dengan demikian, ketika node-node (komputer-komputer) telah banyak terhubung dan terjadi banyak permintaan terhadap sistem, kapasitas total yang dimiliki oleh sistem juga akan meningkat. Hal ini merupakan kontraproduktif dengan apa yang terjadi pada sistem *client-server*. Dalam sistem *client-server*, bertambahnya *client* justru dapat menyebabkan melambatnya transfer data di dalam sistem.Sifat terdistribusi yang dimiliki oleh jaringan P2P ini juga dapat meningkatkan kestabilan/kekokohan (*robustness*) sistem dari kemungkinan kegagalan (*system failure*). Kestabilan ini disebabkan oleh dua faktor. Pertama, adanya replikasi/penggandaan data yang terjadi di antara para pengguna (*peer*). Kedua, dengan memanfaatkan *resource* komputer *peer* itu sendiri untuk mencari data yang ada di dalam jaringan tanpa mengandalkan satu *resource* komputer *server* saja [11].

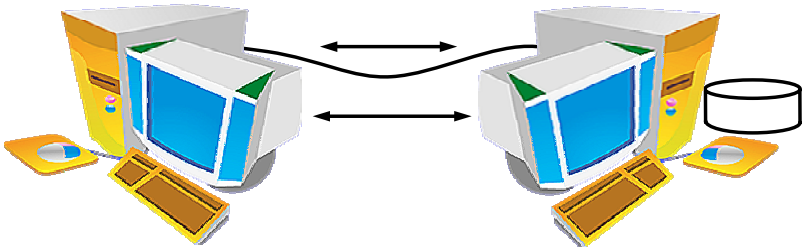
Selain itu pula terdapat kelebihan model dari hubungan P2P:

a. Tidak terlalu mahal, karena tidak membutuhkan *dedicated file server*
b. Mudah dalam konfigurasi programnya, hanya tinggal mengatur untuk operasi model hubungan P2P.

Kekurangan dari model hubungan P2P:

a. Tidak terpusat, terutama untuk penyimpanan data dan aplikasi
b. Tidak aman, karena tidak menyediakan fasilitas untuk keperluan itu.

Gambar topologi model tersentralisasi dapat dilihat pada Gambar 2.2 di bawah ini tersentralisasi model ini membutuhkan sebuah atau beberapa server yang digunakan untuk melakukan beberapa tugas atau fungsi tertentu.



Gambar 2.2 Topologi Model P2P [11]

2.10 Pengertian Multimedia

Pengertian multimedia menurut Ariesto Hadi Sutopo adalah "suatu komputer yang mempunyai alat output yaitu alat display dan hardcopy, dengan rekaman audio berkualitas tinggi, animasi, dan rekaman video". Kemampuan untuk menampilkan multimedia dengan kualitas tinggi sama halnya dengan kamampuan suatu televisi. Perekaman dari sesuatu yang hidup merupakan tambahan kemampuan yang melekat pada komputer simulasi. Penekanan pada

audio dan gambar yang real atau natural merupakan hal yang penting. Realisme dalam menampilkan suatu presentasi sangat diperlukan karena dapat membuat kesan bahwa data source yang digunakan lebih baik dan mudah dipahami oleh *audiens* [1].

2.10.1 Kategori Multimedia

Multimedia dapat didefinisikan menjadi dua kategori yaitu multimedia *content production* dan multimedia *communication* dengan definisi sebagai berikut [1]:

1. Multimedia *content production* adalah penggunaan dan pemrosesan beberapa media (*text, audio, graphics, animation, video, and interactivity*) yang berbeda untuk menyampaikan informasi atau menghasilkan produk multimedia (music, video, film, game, entertaiment, dll) Atau penggunaan sejumlah teknologi yang berbeda yang memungkinkan untuk menggabungkan media (*text, audio, graphics, animation,video, and interactivity*) dengan cara yang baru untuk tujuan komunikasi. Dalam kategori ini media yang digunakan adalah Media *Teks, Audio, Video, Animasi, Graph / Image, Interactivity* Media *Special Effect*.
2. Multimedia *communication* (*Publishing*) adalah menggunakan media (masa), seperti televisi, radio, cetak, dan Internet, untuk mempublikasikan, menyiarkan, mengkomunikasikan material advertising, publicity, entertaiment, news, education, dll. Dalam kategori ini media yang digunakan adalah TV, Radio, Film, Media Cetak, Musik, Game, Entertaiment, Tutorial, ICT (Internet).

2.10.2 Objek Multimedia

Setiap objek multimedia memerlukan cara penanganan tersendiri, dalam hal kompresi data, penyimpanan dan pengambilan kembali untuk digunakan. Multimedia terdiri dari beberapa objek diantaranya yaitu [1] :

1. *Text*

Teks merupakan dasar dari pengolahan kata dan informasi yang berbasis multimedia. Beberapa hal yang harus diperhatikan adalah:

a. *Hypertext*

Sebagian besar penggunaan link dalam multimedia interaktif berdasarkan penggunaan hypertext yang biasa disebut dengan hotword atau hotkey. Hal ini berarti bahwa user ingin mendapatkan informasi tentang kata atau sebagian tertentu, dilakukan dengan memilih kata dengan mouse dan membuka window yang berisi informasi tambahan dalam bentuk teks, grafik atau audio. Pada umumnya, hotword ditampilkan berbeda dengan teks lain dalam monitor, dibedakan dengan memberikan warna tau font berbeda, pointer mouse berubah pada saat berada diatas hotword, dan lain-lain. Hal ini dapat memudahkan user untuk mengenali teks yang mempunyai hubungan dengan informasi lebih lanjut.

b. *Auto-hypertext*

Beberapa *authoring* software mempunyai fitur yang disebut auto-hypertext. Dengan fasilitas yang ada, pada pengembangan multimedia tidak perlu menentukan tanda khusus pada teks yang mempunyai hubungan dengan link. Tetapi, program mengenali teks yang mempunyai informasi tertentu dan

langsung secara otomatis menampilkan informasi bila teks dipilih oleh user. Dalam hal ini, user ridak dapat membedakan teks yang mempunyai informasi lebih lanjut atau tidak, sehingga fasilitas ini tidak memudahkannya. Tetapi, developer dapat menghemat lebih banyak waktu karena program secara otomatis membuat link hypertext.

c. *Text searching*

Pencarian teks merupakan fitur yang memudahkan user dengan memasukan suatu kata (atau memilih dari suatu daftar kata) dalam program multimedia, user dapat dengan cepat memperoleh informasi yang berhubungan dengan kata tersebut. Hal ini serupa dengan pencarian teks dalam indeks suatu buku, kemudian dapat membuka halaman tertentu untuk memperoleh informasi lebih lanjut. Dalam program multimedia, proses pencarian dapat dilakukan lebih cepat di bandingkan dengan penggunaan indeks pada pencarian dalam buku. Beberapa *outhoring tool* dilengkapi dengan kemampuan pencarian teks yang *fleksible* dan baik, sehingga mempermudah user untuk menggunakannya.

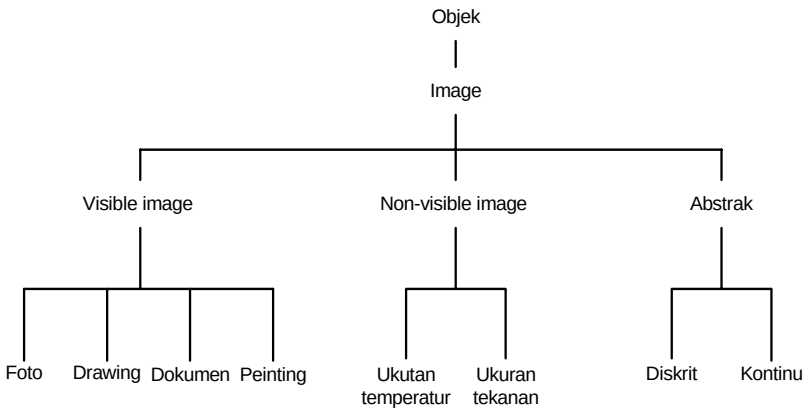
d. *Import* dan *Export Text*

Beberapa teks yang digunakan dalam program multimedia mungkin telah ada dan dibuat sebelumnya dengan pengolahan kata, atau dapat memasukan data yang telah tersimpan dalam database. Bila file mempunyai ukuran besar, maka tidak memerlukan waktu untuk memasukan kembali kedalam multimedia secara manual, sehingga diperlukan paket outhoring yang dapat mengimpor teks dan database dari file lain. Pada umumnya program multimedia dapat membaca file teks ASCII. ASCII adalah singkatan dari *American Standard*

Code for Information Interchange, yaitu format karakter yang dapat digunakan pada semua komputer dan program.

2. *Image*

Secara umum *Image* atau *Grafik* berarti still image seperti foto dan gambar. Manusia sangat berorientasi pada visual (*visual oriented*), dan gambar merupakan sarana yang sangat baik untuk menyajikan informasi. Semua objek yang disajikan dalam bentuk grafik adalah bentuk setelah dilakukan encoding dan tidak mempunyai hubungan langsung dengan waktu. Gambar 2.4 memperlihatkan hirarki dari kelas objek yang termasuk didalam kategori *image.* Kelompok ini termasuk tipe data seperti dokumen, *image,* faksimil, fractal, bitmap dan still foto.



Gambar 2.3 Hirarki Objek Image Dalam Multimedia [1]

Keterangan gambar 2.3:

a. *Visible image*

Kelompok visible image termasuk drawing (seperti blue print engineering, gambar denah, dan lain-lain), dokumen (di-scan sebagai image), painting (hasil scan yang dibuat dengan aplikasi paint pada komputer), foto (hasil scan

atau dimasukan komputer langsung dengan kamera digital, dan still frame yang diambil dari kamera video.

b. *Non-visible image*

Non-visible image adalah image yang tidak disimpan sebagai image tetapi ditampilkan sebagai image. Contohnya, ukuran tekanan, ukuran temperatur, dan tampilan meteran lainya.

c. *Abstrak*

Image abstrak sebenarnya bukan image yang terdapat dalam kenyataan, tetapi dihasilkan oleh komputer seperti dalam perhitungan matematik.

3. *Clip art*

Clip art adalah kumpulan dari image dan objek sederhana, seperti gambar telepon, komputer, bunga, yang dapat digunakan dalam aplikasi sebagai still image atau animasi. Banyak paket authoring dilengkapi dengan library dari clip art, sehingga user dapat dengan mudah menggunakannya. Hal ini sangat membantu bila tidak mempunyai scanner atau alat lain yang dapat digunakansebagai alat input untuk memasukan gambar ke dalam komputer.

4. *Animation*

Animasi digunakan untuk menggambarkan objek yang bergerak agar kelihatan hidup. Untuk membuat satu durasi animasi memerlukan jumlah gambar (*Frame*) yang cukup banyak. Jumlah *frame* tiap detik (*Frame per second* / *fps*) merupakan satuan yang akan menghasilkan kuaalitas animasi. Makin banyak *frame* per detik maka makin baik kualitas animasi yang dihasilkannya. Dalam authoring software, biasanya animasi mencangkup kemampuan recording dan

playback. Fasilitas yang dimiliki oleh software animasi mencakup integrated animation tool, animation clip, import animasi, record-ing, playback, dan transition effect.

5. *Audio*

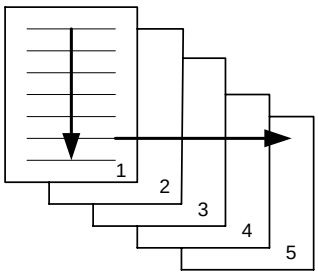
Suara digunakan untuk lebih memperjelas pengertian yang ditampilkan dengan cara lain. Contohnya, narasi merupakan kelengkapan dari penjelasan yang dilihat melalui dari *video,* selain itu suara dapat memberi kejelasan karakteristik suatu gambar misalnya musik dan suara efek (*sound effect*).

6. *Interactive link*

Sebagian dari multimedia adalah interaktif, dimana pengguna dapat menekan mouse atau objek pada screen deperti button atau teks yang menyebabkan program melakukan perinyah tertentu. Interactive link diperlukan bila pengguna menunjuk pada suatu objek atau button supanya dapat mengakses program tertentu. Interactive link duperlukan untuk menggabungkan beberapa elemen multimedia sehingga menjadi informasi yang terpadu. Cara pengaksesan informasi pada multimedia terdapat dua macam, yaitu linear dan non-linear.

a. *Information linear*

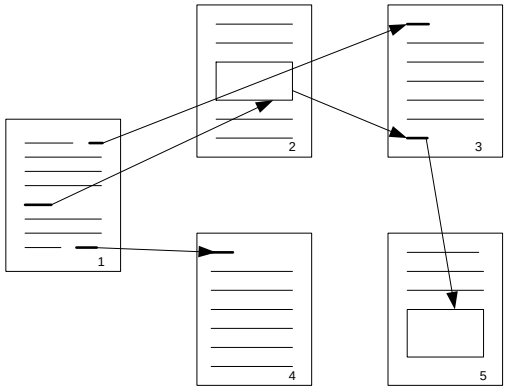
Informasi linier biasanya berupa media presentasi yang pada umumnya tidak dilengkapi alat untuk mengontrol user. Presentasi berjalan secara sekuensial sebagai garis lurus. Presentasi linier digunakan sesuai dengan jumlah audiens lebih dari satu orang. Contoh multimedia seperti ini adalah program TV dan film.



Gambar 2.4 *Information linier* [1]

b. *Information Non-linear*

Informasi non-linier biasanya dalam jenis ini pengguna hanya menggunakan satu komputer untuk satu orang, maka diperlikannya alat pengontrol seperti keyboard, mouse, atau alat input lainya. Hal ini disebut interaktif, dan multimedia yang dapat menangani interaktif user disebut interactive multimedia. Gambar 2.5 menjelaskan alur dari informasi yang berjalan.



Gambar 2.5 *Information Non-linier* [1]

2.11 Metodologi Yang Digunakan

Metodologi yang digunakan adalah *Object Oriented*. dengan menggunakan *Unified Modeling Language (UML)*, sebagai kakas utama dalam analisis dan perancangan sistem, untuk lebih jelasnya akan dijabarkan pengertian dari UML itu sendiri, tujuan menggunakan metode UML dan jenis-jenis diagram dalam UML [2].

2.11.1 Pengertian UML

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasabahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C. Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantic [7].

2.11.2 Tujuan UML

UML itu singkatan dari *Unified Modelling Language*. Sesuai dengan kata terakhir dari kepanjangannya, UML itu adalah salah satu bentuk language atau bahasa. Menurut pencetusnya, UML didefinisikan sebagai bahasa visual untuk menjelaskan, memberikan spesifikasi, merancang, membuat model, dan mendokumentasikan aspek-aspek dari sebuah system.

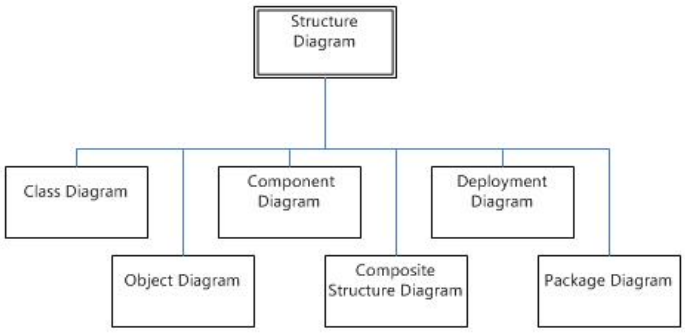
Karena tergolong bahasa visual, UML lebih mengedepankan penggunaan diagram untuk menggambarkan aspek dari system yang sedang dimodelkan. Memahami UML itu sebagai bahasa visual itu penting, karena penekanan tersebut membedakannya dengan bahasa pemrograman yang lebih dekat ke mesin. Bahasa

visual lebih dekat ke mental model pikiran kita, sehingga pemodelan menggunakan bahasa visual bisa lebih mudah dan lebih cepat dipahami dibandingkan apabila dituliskan dalam sebuah bahasa pemrograman.

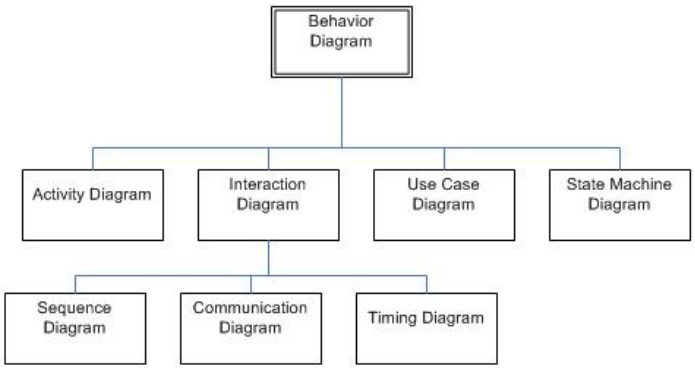
Sebenarnya hampir semua disiplin ilmu memiliki notasi, cara, atau bahasa dalam memodelkan problem dengan notasi diagram yang visual. Ambil contoh dibidang elektro, untuk menggambarkan sebuah system radio, insinyur-insinyur menggunakan diagram sirkuit kelistrikan yang sudah didefinisikan dengan jelas. Dengan diagram sirkuit ini, insinyur elektro bisa mengkomunikasikan komponen-komponen apa saja yang terdapat dalam sebuah system radio kepada insinyur elektro yang lain atau kepada teknisi.

UML adalah salah satu bentuk notasi atau bahasa yang sama yang digunakan oleh professional dibidang software untuk menggambarkan atau memodelkan sebuah system software. Sebelumnya ada banyak notasi atau bahasa lain untuk mencapai keperluan yang sama misalnya DFD (*Data Flow Diagram*) dan *Booch* Diagram. Tetapi sejak matang dan populernya teknologi pemrograman, perancangan, dan analisis berorientasi object, UML telah menjadi *de facto* standard language.

Apa saja yang bisa digambarkan / dimodelkan oleh UML? Sesuai dengan kata pertama dari kepanjangannya, UML mencoba untuk mendeskripsikan pemodelan sebuah system dari segala aspek: pemodelan struktur (aspek statis), pemodelan perilaku (aspek dinamis), dan pemodelan arsitektur. Gambar 2.6 dan Gambar 2.7 berikut ini menunjukan taksonomi pendiagraman UML [7].



Gambar 2.6 Taksonomi Pendiagraman UML [7]



Gambar 2.7 Taksonomi pendiagraman UML [7]

Secara detail UML tidak akan dibahas dalam posting ini, tetapi kita akan banyak menggunakannya dalam pembahasan-pembahasan selanjutnya. Jadi sambil melakukan pemodelan, dijelaskan juga dengan notasi UML yang cocok dengan konteks pemodelan yang dilakukan.

Ada tiga cara dalam memakai UML dalam melakukan pemodelan sistem diantaranya sebagai berikut [7]:

1. UML sebagai sketsa

   UML digambarkan dalam sketsa coretan-coretan dalam kertas atau *whitboard* secara tidak formal. Biasanya digunakan dalam sesi diskusi tim untuk membahas aspek tertentu dalam tahap analisis dan perancangan.

2. UML sebagai *blueprint* system

   Seperti diagram kelistrikan adalah blueprint dari komponen atau produk yang akan dihasilkan, UML juga bisa menggambarkan blueprint yang identik untuk sebuah system software.

3. UML sebagai bahasa pemrograman

   UML berfungsi sebagai bahasa pemrograman mencoba melakukan semuanya dengan UML sampai kepada produk jadinya. Analisis dan perancangan dilakukan dengan diagram-diagram yang ada dalam UML, sementara sebuah tool atau generator bisa menghasilkan produk akhir dari diagram-diagram ini.

Saat ini UML paling banyak digunakan dengan cara pertama dan kedua. Khusus dalam metode agile (cepat dan ringan). Pemograman berorientasi objek atau juga sering disebut OOP (Object Oriented Programming), merupakan paradigma baru dalam pengenbangab sistem perangkat lunak. Tidak seperti pendahulunya, yaitu pemograman dengan teknik terstruktur (Structured Programming) yang sering mengalami kegagalan dalam sistem perangkat lunak yang tidak sesuai dengan kebutuhan dan harapan user (User's Need and Expectations) dan kerap kali tidak dapat memenuhi jadual yang telah ditentukan sebelumnya, pemograman berorientasi objek dengan paradigma barunya, menjanjikan tingkat penggunaan ulang (Reusability), kinerja (Performance) serta keandalan (Reliability) yang lebih tinggi.

2.11.3 Diagram-diagram Dalam UML

UML mendefinisikan diagram-diagram sebagai berikut [2]:

1. *Use Case Diagram*

Diagram use case memperlihatkan pada kita hubungan-hubungan yang terjadi antara aktor-aktor dengan use case dalam system. Salah satu manfaat dari diagram use case adalah untuk komunikasi antara calon pengguna system perangkat lunak dapat mengamati diagram use case untuk dapat mendapatkan pemahaman yang utuh tentang system yang akan dikembangkan, dengan melihat use case akan dimengerti fungsionalitas apa yang aka dikembangkan oleh sistem

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya.

Use case mencakup aliran-aliran kerja (workflow) dalam sistem dengan sifat internal, sedangkan aktor-aktor mencakup segala sesuatu yang ada diluar sistem bersifat exsternal. Tentu saja ada perbedaan antara use case dan aktor dilihat dari jenis use case yang akan dibuat. Tabel menjelaskan perbedaan model use case antara use case bisnis dan aktor bisnis dengan use case sistem dan aktor dalam *use case* sistem.

Tabel 2.1 Perbedaan Objek-objek Dalam Model Bisnis dan Model Sistem [2]

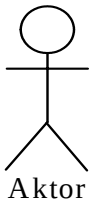
Nama Objek	Model Bisnis	Model Sistem
Use Case	Mendeskripsikan apa yang dikerjakan perusahaan	Mendeskripsikan sistem yang akan atau sedang dikembangkan dalam perusahaan
Aktor	Bersifat eksternal terhadap perusahaan	Bersifat eksternal terhadap sistem yang akan atau sedang dikembangkan
Pekerja bisnis	Bersifat internal terhadap perusahaan	Tidak digunakan

Dapat kita perhatikan tabel diatas bahwa secara konseptual ada kemiripan antara objek-objek pada model bisnis dan objek-objek pada model sistem. Perbedaannya adalah, pemodelan bisnis (business modeling) berfokus kepada organisasi atau perusahaan sedangkan pemodelan sistem lebih berfokus pada sistem di dalam organisasi atau perusahaan yang sedang kita kembangkan.

Untuk lebih jelasnya komponen dari diagram *use case* dapat didefinisikan sebagai berikut:

a. Aktor

Dalam pemodelan sistem dengan UML, aktor adalah seseorang atau sesuatu yang beriteraksi dengan sistem yang sedang kita kembangkan. Aktor berada di luar lingkup sistem perangkat lunak yang sedang kita kembangkan, bersifat eksternal. Secara simbol aktor digambarkan pada Gambar 2.8 sebagai berikut



Gambar 2.8 Simbol dari Aktor [2]

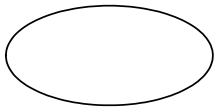
b. *Use Case*

Use case adalah peringkat tertinggi dari fungsionalitas yang dimiliki sistem. Dengan kata lain use case menggambarkan bagaimana seseorang akan menggunakan atau memanfaatkan sistem. Dengan melihat pada use case, para pengguna dapat melihat fungsionalitas apa yang akan disediakan oleh sistem dan menyetujui atau menolak lingkup sistem sebelum proyek berjalan lebih jauh dan akan menghabiskan waktu dan biaya

Saat akan memulai suatu proyek yang baru, pernyataan awal yang selalu muncul adalah ”Bagaimana kita menentukan use case”. Salah satu cara yang sangat baik untuk mengawalinya adalah dengan memperhatikan setiap dokumentasi yang pengguna siapkan. Suatu contoh, dengan memulai pertanyaan dari pesyaratan-persyaratan visi dan misi organisasi atau perusahaan. Selain itu kita juga dapat bertanya kepada diri kita sendiri apa saja kiranya yang diharapkan oleh para pengguna dari sistem perangkat lunak yang akan kita kembangkan. Untuk setiap pengguna (Aktor) dengan memikirkan hal sebagi berikut:

1. Apa yang akan pengguna kerjakan dengan sistem yang akan dikembangkan.
2. Apa yang para pengguna butuhkan untuk melihat informasi-informasi
3. Apakah yang perlu sistem lakukan saat terjadi event tertentu yang datang dari luar sistem.

Secara simbol *use case* digambarkan pada Gambar 2.9 sebagai berikut:



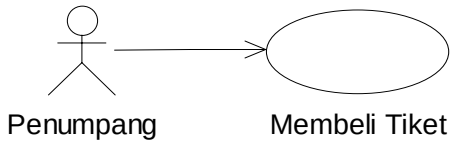
Gambar 2.9 Simbol dari *Use Case* [2]

c. Relasi

Use case dan aktor masing masing tidak berdiri sendiri. Mereka saling berhubungan dengan apa yang dinamakan relasi ada beberapa relasi yang dikanal dalam UML diantaranya:

1. Relasi Asosiasi

Relasi yang terjadi antara aktor dengan use case biasanya berupa asosiasi. Dalam UML, asosiasi digambarakan dengan garis lurus dengan kepala panah disalah satu ujungnya seperti terlihat pada Gambar 2.10 Relasi Asosiasi di bawah ini.

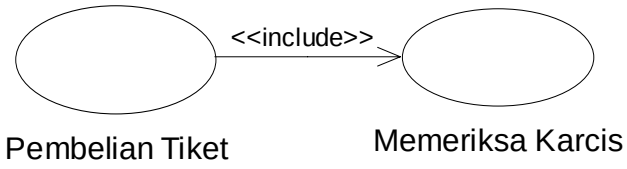


Gambar 2.10 Relasi Asosiasi [2]

2. *Include Relationship*

Relasi cakupan (include relationship) memungkinkan suatu use case untuk menggunakan fungsionalitas yang disediakan oleh use case yang lainnya. Relasi ini dapat digunakan dalam salah satu kasus diantaranya, jika dua atau lebih use case memiliki sejumlah fungsi yang identik, fungsionalitas-fungsionalitas yang sama ini dapat dipisahkan menjadi satu use case

tersendiri. Masing-masing use case yang lain kemudian dapat memiliki include relationship dengan use case yang baru.

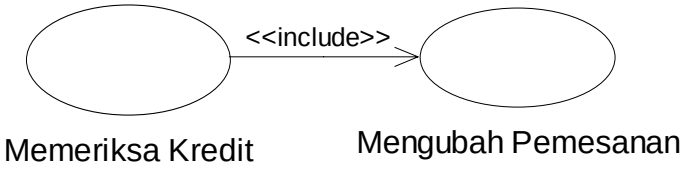


Gambar 2.11 *Include Relationship* [2]

Dalam kasus yang lain include relationship sangat membantu saat suatu use case memiliki sejumlah besar fungsionalitas. Dalam hal ini, include relationship dapat digunakan untuk memecahkannya dan memodelkanya menjadi dua atau lebih use case yang lebih kecil. Perhatikan pada Gambar 2.11 di atas terlihat contoh suatu use case ”Pembelian Tiket” akan menguji atau memeriksa apakah nomor kartu kredit yang diamasukan adalah sah dan jumlah uangnya mencukupi untuk melakukan transaksi. Karena fungsionalitas ini digunakan selama proses ”Pembelian Tiket”, maka relasi yang digunakan adalah include relationship.

3. *Extends Relationship*

Extend relationship memungkinkan suatu use case memiliki kemungkinan untuk memperluas fungsionalitas yang disediakan oleh use case yang lainya. Ini agak mirip dengan iclude relationship, namun pada extend relationship tidak harus terjadi apa yang diharapkan. Untuk jelasnya perhatikan Gambar 2.12 di bawah ini.



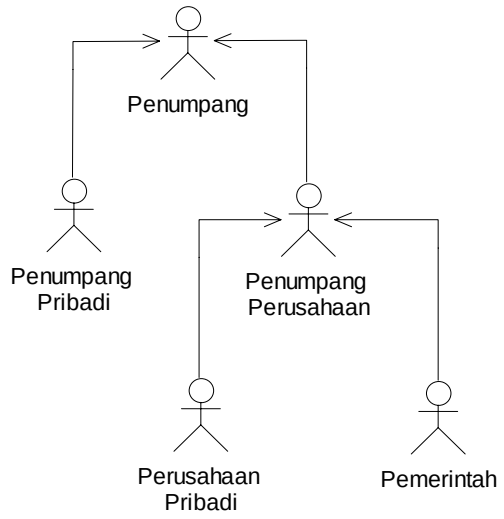
Gambar 2.12 *Extended Relationship* [2]

Dengan melihat Gambar diatas dimana *use case* "Memeriksa Kredit" memiliki hubungan relasi extend relationship dengan use case "Mengubah Pemesanan". Sementara use case "Mengubah Pemesanan" berjalan, *use case* "Memeriksa Kredit" berjalan jika dan hanya jika nilai pemesanan dirubah. Jika niali pemesanan tidak dirubah, *use case* "Memeriksa Kredit" tidak perlu berjalan.

4. Relasi Generalisasi

Relasi generalisasi digunakan untuk memperlihatkan bahwa beberapa aktor atau use case memiliki sesuatu hal yang bersifat umum.

Sebagai contoh dalam kasus pemesanan layanan penerbangan pada Gambar 2.13 di bawah ini.



Gambar 2.13 Generalisasi Untuk Penumpang [2]

Gambar diatas penumpang memungkinkan bisa dibedakan menjadi "Penumpang Pribadi" dan "Penumpang Perusahaan" dimana yang terakhir dapat dibedakan lagi menjadi "Perusahaan Pribadi" dan "Pemerintah". Penumpang adalah aktor yang bersifat abstrak, tidak punya instansi langsung. Penumpang kalau tidak penumpang pribadi haruslah penumpang perusahaan, tidak ada alternatif lain. Dalam hal ini, penumpang pribadi dan penumpang perusahaan memiliki instansiasi langsung sehingga ia dinamakan aktor konkret. Dengan cara yang sama kita bisa menjelaskan arah spesialisasi ke arah penumpang perusahaan pribadi dan penumpang perusahaan pemerintah.

Pada dasarnya generalisasi sangat berguna sebab kita bisa mengelompokan sifat-sifat yang umum dari sejumlah aktor atau use case menjadi aktor atau use case tunggal, kemudian bisa mewariskan sifat-sifat yang umum tadi kesejumlah aktor atau use case yang lain. Dengan cara ini, kita akan sangat menghemat penulisan kode-kode, atribut-atribut, serta aliran-aliran kerja yang bersifat umum.

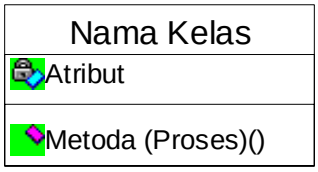
2. *Class Dagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok:

1. Nama (dan *stereotype*).
2. Atribut.
3. Metoda atau proses.

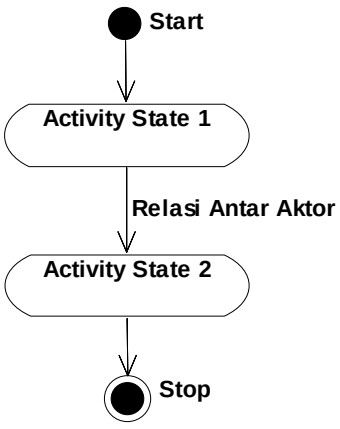
Bentuk dari kelas dapat di lihat pada Gambar 2.14 dibawah ini:



Gambar 2.14 Bentuk Kelas [2]

3. *Activity Diagram*

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, bagaimana mereka berakhir dan terdapat transisi dari satu state ke state lain , dapat dilihat contoh dari activity diagram pada Gambar 2.15 di bawah ini.



Gambar 2.15 Contoh *Activity Diagram* [2]

Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram*

khusus, dimana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

4. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas

tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan [2].

Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada *fase* desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message.

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

5. *Collaboration Diagram*

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, dimana *message* dari level tertinggi memiliki nomor 1. Messages dari level yang sama memiliki *prefiks* yang sama [2].

6. *Component Diagram*

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang

lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

7. *Deployment Diagram*

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, *server* atau piranti keras), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisikal [2].

Sebuah *node* adalah *server*, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.

2.12 Pengertian *Macromedia Flash*

Macromedia Flash adalah sebuah program animasi yang telah banyak digunakan oleh para animator untuk menghasilkan animasi yang propesional. Diantara program animasi yang ada, macromedia flash merupakan program yang paling fleksibel dalam membuat animasi seperti animasi interaktif, game, *company profile*, *movie*, dan tampilan animasi lainya. Software ini berguna untuk membuat animasi yang biasanya digunakan untuk berbagai keperluan di Internet. Misalnya, untuk membuat situs, banner iklan, logo yang beranimasi, serta animasi pelengkap lainnya [1].

2.12.1 Sejarah Program Macromedia Flash

Flash dikembangkan dari suatu aplikasi yang bernama SmartSketch. SmartSketch sendiri merupakan aplikasi untuk menggambar yang diluncurkan pada 1994 oleh FutureWave, bukan oleh Macromedia. Aplikasi ini cukup sukses di tengah pasar aplikasi menggambar yang dikuasai oleh Illustrator dan Freehand. Pada musim panas 1995, SmartSketch memperoleh masukan dari penggunanya agar SmartSketch dapat digunakan untuk membuat animasi. FutureWave sangat tertarik untuk membuat suatu aplikasi untuk membuat animasi. Namun FutureWave agak pesimis mengenai pemasarannya, karena pada saat itu animasi hanya didistribusikan dengan VHS atau CD-ROM.

Kemudian World Wide Web mulai mengembangkan sayapnya, dimana grafik dan animasi menjadi vital. FutureWave melihat kesempatan ini untuk memasarkan aplikasi yang mampu menghasilkan animasi dua dimensi. Kemudian SmartSketc dimodifikasi sehingga mampu menghasilkan animasi dengan menggunakan pemrograman Java sebagai player-nya. Namanya juga sedikit dimodifikasi menjadi SmartSketch Animator. Namun, nama SmartSketch Animator dirasakan kurang menjual, sehingga nama tersebut diubah menjadi CelAnimator. Tetapi kemudian, karena kuatir dicap sebagai aplikasi pembuat kartun, CelAnimator diubah menjadi FutureSplash Animator.

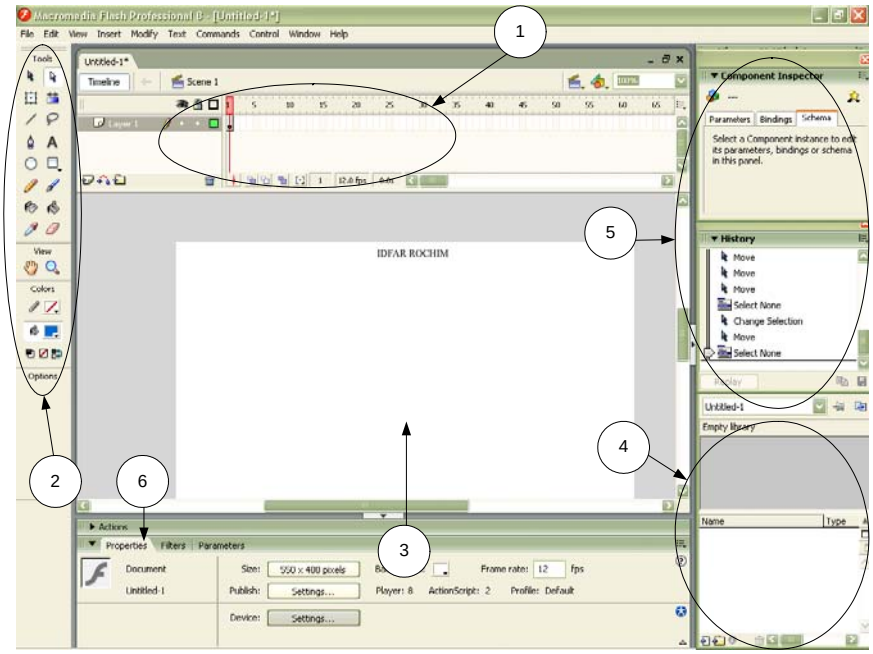
Walaupun dengan ide yang cukup revolusioner, FutureSplash sulit populer. Oleh karena itu FutureWave mendekati Adobe. Namun karena demo FutureSplash yang kurang memuaskan dengan lambatnya animasi, Adobe menolak memproduksi FutureSplash. Baru pada November 1996, Macromedia

mendekati FutureWave untuk bekerja sama. FutureWave menyetujui tawaran Macromedia. Kemudian FutureSplash Animator diubah namanya menjadi Macromedia Flash 1.0.

Sekarang ini Macromedia Flash sudah dirilis sampai Flash 6.0 atau lebih sering disebut Flash MX. User interface-nya berkembang terus sehingga lebih user friendly. Fitur-fitur baru pada setiap versi terus ditambah dan dikembangkan.

2.12.2 Lingkungan Kerja Dari Macromedia Flash

Lingkungan kerja dari *macromedia flash* dapat dilihat pada Gambar 2.16 dibawah ini [1].



Gambar 2.16 Lingkungan Kerja *Macromedia Flash* [1]

Keterangan Gambar 2.16:

1. *Timeline*, berisikan kotak-kotak kecil yang disebut *frame* yang diurut secara melintang dan *layer* yang diurutkan kebawah. Timeline digunakan untuk mengetahui panjangnya durasi animasi.

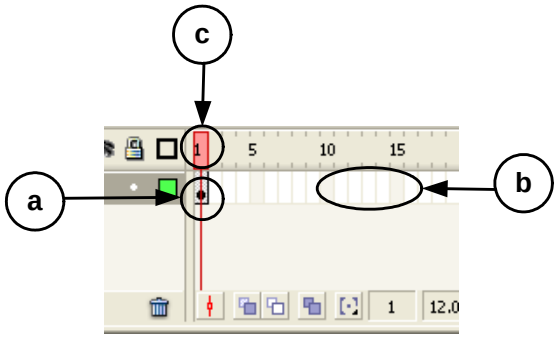
2. *Toolbar,* aplikasi ini menyebutkan *Toolbox*. Dibagi 4 kelompok : ***Tool*** berisi tombol-tombol untuk membuat dan mengedit gambar, ***View*** untuk mengatur tampilan lember kerja, ***Colors*** menentukan warna yang dipakai pada saat mengedit, ***Option*** sebagai alat bantu lain untuk mengedit gambar.
3. *Stage,* merupakan area kerja untuk melakukan pengeditan setiap *frame* dari *movie flash*. *Frame* yang dipilih di *timeline* akan muncul di layer ini. Objek atau gambar yang disimpan diluar *stage* tidak akan terlihat saat animasi dijalankan.
4. *Library*, merupakan tempat penyimpanan berbagai *symbol* yang digunakan dalam animasi, *Symbol* merupakan kumpulan gambar baik *movieclip*, tombol (*button*) dan gambar statis (*graphic*). Banyaknya *symbol* yang terdapat pada *library* tidak mempengaruhi ukuran *file flash*. Setip *symbol* yang dibuat akan tersimpan secara otomatis di *library*. Untuk menggunakan *symbo*l tersebut dalam stage cukup dengan meng-drag *symbol* tersebut.
5. *Panel,* berisi informasi suatu objek. Setiap panel biasanya terdiri dari beberapa menu tab. Untuk menampilkan *panel* yang lengkap, klik [***Windows***] [***Panel State***] [***Defaul Layout***].
6. *Properties*, digunakan untuk mengedit *properties* atau parameter-parameter dari objek-objek yang kita buat, atau *tool* yang sedang kita pakai, sehingga kita dapat mengatur objek atau *tool* menjadi lebih mudah.

2.12.3 Istilah-Istilah Dalam Flash

Dalam Flash terdiri dari 3 jenis *symbol* yang sering digunakan antara lain yaitu [1]:

1. *Movieclip*, merupakan *symbol* yang dapat memainkan animasi dalam aplikasi *flash*. *Movieclip* ini dapat berisi kontrol interaktif, *sound* maupun *movieclip* lainya. Secara *default*, *movieclip* ini akan memainkan animasi didalamnya secara berulang-ulang.
2. *Button* (Tombol), merupakan *movieclip* dengan empat *frame* interaktif, dimana masing-masing *frame* bereaksi sesuai dengan *pointer mouse* bereaksi terhadap *button*.
3. Grafik, merupakan gambar statis untuk membuat potongan-potongan animasi yang terangkum dalam *timeline*, Kontrol interaksi dan suara tidak dapat bekerja di dalam animasi *symbol* grafik.
4. *Timeline*, didalam *timeline* terdapat istilah diantaranya :
   a. *Key frame*, *frame* yang siap digunakan untuk menyimpan objek
   b. Frame
   c. *Playhead*, menunjukkan *frame* yang sedang berjalan

   Untuk lebuh jelasnya dapat dilihat pada Gambar 2.17 di bawah ini:



Gambar 2.17 Istilah Dalam *Timeline* [1]

2.12.4 Jenis *File* Dalam *Flash*

Dalam *macromedia flash* dikenal dengan 3 jenis *file* yaitu:

1. *File* Utama (*.fla)

   File ini merupakan *file* utama (*source*) dari *flash*, *file* jenis ini juga dapat dibuka dan di edit oleh *flash.*

2. *File* hasil ekspor (*.swf)

   File ini merupakan *file* hasil **run** dari *flash*. Biasa digunakan dalam hal web. File dalam format ini dapat dilihat apabila terdapat aplikasi *flash* atau *flash player* pada komputer.

3. *File* hasil *Publish*

   File hasi *publish* ini banyak jenisnya. Tergantung pada format yang akan kita *publish.* Contoh *file* tersebut yaitu *file flash* (*.swf), web (*.html), gambar (*.gif,*.jpg,*.png) *Windows Projector* (*.exe), *Machintos Projector* (*.hqx) dan *video* (*.avi,*.mov).

2.12.5 Pengertian Action Script Pada Flash

Macromedia Flash adalah sebuah software animasi yang sekarang menjadi software favorit para web designer untuk membuat webnya terlihat dinamis dan lebih atraktif. Bahkan sekarang Flash digunakan untuk berbagai keperluan, diantaranya untuk presentasi, proposal modern, e-card, game dll. Mengapa menggunakan Action Script dan apa manfaatnya, Action Script dirancang untuk memudahkan programmer untuk membuat aplikasi berbasis flash, keuntungannya antara lain adalah untuk system navigasi pada suatu situs atau presentasi, menghemat ukuran file, membuat hal-hal yang bersifat interaktif.

1. Tipe Data dan Variabel

Variabel adalah suatu penampung data. Penulisan variable diantaranya:

1. Tidak boleh menggunakan variabel yang sama dengan keyword atau literal dari Action Script sendiri, contoh true,false, null, dll.
2. Variabel harus unik, artinya nama variabel yang satu harus beda dengan nama variabel yang lainnya, contoh kingkong1, kingkong2.

Contoh tipe data dan variable dalam flash:

Tabel 2.2 Tipe Data dan Variable Dalam Flash

Tipe Data	Keterangan
String	Tipe data yang berupa kumpulan huruf, angka, atau symbol.
Number	Tipe data yang berupa angka.
Boolean	Tipe data yang berupa nilai yang berisi true(1) atau false(0).
Object	Kumpulan dari berbagai properties.
Null	Tipe data Null hanya memiliki satu nilai yaitu null,biasanya digunakan untuk mendefinisikan data yang belum terisi.
Undefined	Tipe data Null hanya memiliki satu nilai yaitu undefined.
Movie Clip	Movie Clip adalah symbol untuk menghandle animasi, tipe data ini hanya mengacu ke elment berupa grafik.

2. Operator

Operator digunakan untuk memanipulasi variable diantaranya dapat dilihat pada tabel 2.3 simbol dari operator dalam flash dibawah ini.

Tabel 2.3 Simbol Dari Oprator Dalam Flash

Symbolic operators	
--	-- (decrement)
++	++ (increment)
!	! (logical NOT)
!=	!= (inequality)
!==	!== (strict inequality)
%	% (modulo)
%=	%= (modulo assignment)
&	& (bitwise AND)
&&	&& (short-circuit AND)
&=	&= (bitwise AND assignment)
()	() (parentheses)
-	- (minus)
*	* (multiplication)
*=	*= (multiplication assignment)
,	, (comma)
.	. (dot)
?:	?: (conditional)
/	/ (division)
//	// (comment delimiter)
/*	/* (comment delimiter)
/=	/= (division assignment)
[]	[] (array access)
^	^(bitwise XOR)
^=	^= (bitwise XOR assignment)
{}	{} (object initializer)
|	| (bitwise OR)
||	|| (logical OR)
|=	|= (bitwise OR assignment)
~	~ (bitwise NOT)
+	+ (addition)
+=	+= (addition assignment)
<	< (less than)
<<	<< (bitwise left shift)
<<=	<<= (bitwise left shift and assignment)
<=	<= (less than or equal to)
<>	<> (inequality)
=	= (assignment)
-=	-= (subtraction assignment)
==	== (equality)
>	> (greater than)
>=	>= (greater than or equal to)
>>	>> (bitwise right shift)
>>=	>>= (bitwise right shift and assignment)
>>>	>>> (bitwise unsigned right shift)