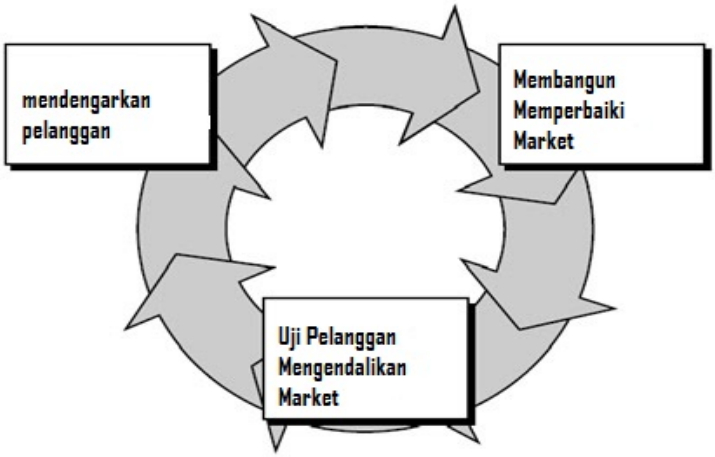
BAB II
LANDASAN TEORI

2.1 Metode Prototype

Dalam perancangan Tugas Akhir ini penulis menggunakan metode *Prototype*. *Prototype Model* adalah salah satu metode pengembangan perangkat lunak yang banyak digunakan. Dengan Metode Prototyping ini pengembangan dan pelanggan dapat saling berinteraksi selama proses pembuatan sistem. Sering terjadi seorang pelanggan hanya mendefinisikan secara umum apa yang dibutuhkan, Pemrosesan dan data-data apa saja yang dibutuhkan. Sebaliknya disisi pengembang Kurang memperhatikan efesiensi Algoritma. Kemampuan sistem oprasi dan interface yang menghubungkan manusia dengan computer. [5]

Pada Prototyping model kadang – kadang klien hanya memberikan beberapa kebutuhan umum software tanpa detile input, proses atau detail output dilain waktu mungkin tim pembangun (developer) tidak yakin terhadap efesiensi dari algoritma yang digunakan, tingkat adaptasi terhadap sistem operasi atau rancangan form user interface. Ketika situasi seperti ini terjadi model prototyping sangat membantu proses pembangunan software. Proses pada prototyping bisa dijelaskan sebagai berikut

a. Pengumpulan Kebutuhan : developer dan klien bertemu dan menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya. Detail kebutuhan mungkin tidak dibicarakan disini, pada awal pengumpulan kebutuhan.
b. Perancangan : Perancangan dilakukan cepat dan rancangan mewakili aspek software yang diketahui. Dan rancangan ini menjadi dasar pembuatan prototype.
c. Evaluasi Prototype : klien mengevaluasi prototype yang dibuat dan dipergunakan untuk memperjelas kebutuhan software.



Gambar 2.1 Model *Prototype* menurut *Roger S. Pressman, Ph.D.* [5]

2.1.1 Tahapan-tahapan *Prototype*

Tahap-tahap pengembangan *Prototype* model menurut *Roger S. Pressman, Ph.D.* adalah :

1. Mendengarkan pelanggan

   Pada tahap ini dilakukan pengumpulan kebutuhan dari system dengan cara mendengar keluhan dari pelanggan. Untuk membuat suatu system yang sesuai kebutuhan, maka harus diketahui terlebih dahulu bagaimana system yang sedang berjalan untuk kemudian mengetahui masalah yang terjadi.

2. Merancang dan Membuat Prototype

   Pada tahap ini, dilakukan perancangan dan pembuatan prototype system. *Prototype* yang dibuat disesuaikan dengan kebutuhan system yang telah didefinisikan sebelumnya dari keluhan pelanggan atau pengguna.

3. Uji coba

   Pada tahap ini, *Prototype* dari system di uji coba oleh pelanggan atau pengguna. Kemudian dilakukan evaluasi kekurangan-kekurangan dari kebutuhan pelanggan. Pengembangan kemudian kembali mendengarkan keluhan dari pelanggan untuk memperbaiki *Prototype* yang ada.

2.1.2 Kelebihan Metode Prototype

Kelebihan metode ***Prototype***:

1. Adanya komunikasi yang baik antara pengembang dan pelanggan
2. Pengembangan dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan
3. Lebih menghemat waktu dalam pengembangan system
4. Penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkannya.

2.1.3 Kekurangan Metode Prototype

Kekurangan metode *Prototype :*

1. Resiko tinggi yaitu untuk masalah-masalah yang tidak terstruktur dengan baik, ada perubahan yang besar dari waktu ke waktu, dan adanya persyaratan data yang tidak menentu.
2. Interaksi pemakai penting. Sistem harus menyediakan dialog on-line antara pelanggan dan komputer.
3. Hubungan pelanggan dengan komputer yang disediakan mungkin tidak mencerminkan teknik perancangan yang baik.

2.2 Metode Iqro

Metode iqro' adalah suatu metode membaca Al-Qur'an yang menekankan langsung pada latihan membaca. Adapun buku panduan iqro' terdiri dari 6 jilid di mulai dari tingkat yang sederhana, tahap demi tahap sampai pada tingkatan yang sempurna. [4]

Metode iqro' ini dalam prakteknya tidak membutuhkan alat yang bermacam-macam, karena ditekankan pada bacaannya (membaca huruf Al-Qur'an dengan fasih). Bacaan langsung tanpa dieja. Artinya diperkenalkan nama-nama huruf hijaiyah dengan cara belajar siswa aktif (CBSA) dan lebih bersifat individual.

Metode pembelajaran ini pertama kali disusun oleh H. As'ad Humam di Yogyakarta. Buku metode Iqro' ini disusun/dicetak dalam enam jilid sekali. Di mana dalam setiap jilidnya terdapat petunjuk mengajar dengan tujuan untuk

memudahkan setiap peserta didik (santri) yang akan menggunakannya, maupun ustadz/ustadzah yang akan menerapkan metode tersebut kepada santrinya. Metode iqro; ini termasuk salah satu metode yang cukup dikenal dikalangan masyarakat, karena metode ini sudah umum digunakan ditengah-tengah masayarakat Indonesia.

2.3 Pendekatan Cara Belajar Siswa Aktif (CBSA)

Pengertian Pendekatan Cara Belajar Siswa Aktif (CBSA)

Pendekatan pembelajaran dapat diartikan sebagai titik tolak atau sudut pandang kita terhadap proses pembelajaran, yang merujuk pada pandangan tentang terjadinya suatu proses yang sifatnya masih sangat umum, di dalamnya mewadahi, menginspirasi, menguatkan, dan melatari metode pembelajaran dengan cakupan tertentu.

Menurut Nana Sujana (1988), dikatakan bahwa CBSA adalah suatu proses belajar-mengajar yang menggunakan berbagai metode yang subjek didiknya terlibat secara intelektual dan emosional, sehingga subjek didik betul-betul berperan dan berpartisipasi aktif dalam kegiatan belajar. [7]

Menrut Misbah Partika (1987), dikatakan CBSA adalah proses belajar mengajar yang menggunakan berbagai metode yang menitik beratkan kepada keaktifan yang bersifat fisik, mental, emosional maupun intelektual untuk mencapai tujuan pendidikan yang berhubungan dengan wawasan kognitif, afektif dan psikomotor secara optimal. [7]

Bertitik tolak dari beberapa definisi tersebut di atas dapat diambil suatu kesimpulan bahwa Cara Belajar Siswa Aktif (CBSA) merupakan suatu pendekatan yang diterapkan dalam proses belajar-mengajar dengan menekankan pada keterlibatan kemampuan peserta didik, baik secara fisik, mental, intelektual maupun emosionalnya sehingga diperoleh hasil belajar yang berupa keteerpaduan antar aspek kognitif, afektif dan psikomotor dalam kesatuan pribadi peserta didik yang utuh seperti yang diinginkan dalam tujuan pendidikan nasional. [7]

Cara Belajar Siswa Aktif (CBSA) merupakan suatu pendekatan sebagai urutan pembelajaran yang mengarah kepada pengoptimalisasian pelibatam intelektual-emosional siswa dalam proses pembelajaran, dengan pelibatan fisik siswa apabila diperlukan. Peningkatan CBSA dari suatu proses pembelajaran berarti pula mengarahkan proses pembelajaran yang berorientasi pada siswa atau dengan kata lain menciptakan pembelajaran berdasarkan siswa (*Student Based Instruction*). Konsep CBSA yang dalam bahasa Inggris disebut Student Active Learning (SAL) dapat membantu pengajar meningkatkan daya kognitif pembelajar. [7]

2.4 Pengertian Perancangan Sistem

Pengertian Perancangan Sistem menurut George M. Scott dalam buku *Jogiyanto* HM tahun 1991 halaman 196 dapat diuraikan sebagai berikut :

"Desain system menentukan bagaimana suatu system akan menyelesaikan apa yang mesti diselesaikan, tahap ini menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu system sehingga setelah instalasi dari system akan benar-benar memuaskan rancang bangun yang telah diterapkan pada akhir analisi system. (Jogiyanto HM, 1991 : 196). [3]

2.5 Pengertian Media Pembelajaran

Menurut Arsyad (2002), kata media berasal dari bahasa latin *medius* yang secara harfiah berarti 'tengah','perantara',atau 'pengantar'. Menurut Bovee yang dikutip Ouda Teda Ena (2001), media adalah sebuah alat yang mempunyai fungsi menyampaikan materi.[1]

Pembelajaran adalah sebuah proses komunikasi antara murid, pengajar dan materi. Komunikasi tidak akan berjalan tanpa bantuan sarana penyampaian pesan atau media (Ouda Teda Ena, 2001). Gerlach dan Erly (1971) yang dikutip Arsyad (2002) mengatakan bahwa media apabila dipahami secara garis besar adalah manusia, materi, atau kejadian yang membangun kondisi yang membuat siswa mampu memperoleh pengetahuan, ketrampilan, dan sikap. Secara lebih khusus, pengertian media dalam proses belajar mengajar cenderung diartikan sebagai alat-alat grafis, photografis, atau elektronis untuk menangkap, memproses, dan menyusun kembali informasi visual dan verbal. [1]

2.6 Pengertian Multimedia

Multimedia dapat diartikan sebagai "lebih dari satu media". *Multimedia* biasa berupa kombinasi antara teks, grafis animasi, suara dan gambar. Namun pada bagian ini perpaduan dan kombinasi dua atau lebih jenis media ditekankan kepada kendali computer sebagai penggerak keseluruhan gabungan media ini. Dengan demikian Arti *Multimedia* yang umumnya dikenal dewasa ini adalah berbagai macam kombinasi grafik, teks, suara, video, dan animasi. Penggabungan ini merupakan suatu kesatuan yang secara bersama-sama menampilkan informasi, pesan atau isi pelajaran . [1]

Konsep penggabungan ini dengan sendirinya memerlukan beberapa jenis peralatan perangkat keras yang masing-masing tetap menjalankan pengendali seluruh peralatan itu. Jenis peralatan ini adalah computer, Video Kamera, video cassette recorder (VCR), CD player, compact disc. Kesemua peralatan ini harus kompak dan kerja sama dalam menyampaikan informasi kepada pemakainya.

Informasi yang disajikan melalui *multemedia* ini berbentuk dokumen yang hidup, dapat dilihat dilayar monitor, dan dapat didengar suaranya, dilihat gerakanya (video atau animasi). *Multimedia* bertujuan untuk menyajikan informasi dalam

bentuk yang menyenangkan, menarik, mudah dimengerti, dan jelas. Informasi akan mudah dimengerti karena sebanyak mungkin indera, terutama telinga dan mata, digunakan untuk menyerap informasi tersebut.

Kemampuan teknologi elektronika makin besar. Bentuk informasi grafis, video, animasi, suara, dan lain-lain dengan mudah dapat dihasilkan dengan mutu yang cukup baik. Misalnya video kamera berfumgsi merekam video yang diinginkan untuk kemudian ditransfer dan digabungkan dengan animasi, grafik dan teks yang dihasilkan komputer.

Multimedia sendiri terdiri dua kategori, yaitu *movie linear* dan *non linear (*interaktif). *Movie Linear* dapat berinteraksi dengan aplikasi web yang lain melalui penekanan sebuah tombol navigasi, pengisian form, animasi logo, animasi bentuk, dengan singkronisasi suara. Untuk *movie linear* pada prinsipnya sama dengan *Movie non linear.* Akan tetapi dalam *movie* ini tidak ada penggabungan seperti pada *movie non linear* hanya animasi-animasi biasa.

2.6.1 Objek Multimedia

Multimedia oleh Ariesto Hadi Sutopo (2003: 196), diartikan sebagai kombinasi dari macam-macam objek *Multimedia*, yaitu teks, image, animasi,audio, video dan link interaktif untuk menyajikan informasi. .[1]

Setap objek *multimedia* memerlukan cara penanganan tersendiri, dalam hal kompresi data, penyimpanan, dan pengambilan kembali untuk digunakan. *Multimedia* terdiri dari beberapa objek, yaitu teks, grafik, image, animasi audio, video dan link interaktif.

a. **Teks**

Teks merupakan dasar dari pengolahan kata informasi berbasi *multimedia.* Beberapa hal yang perlu diperhatikan adalah penggunaan *hypertext, auto-hypertext, text style, import text,* dan *export text.*

1) Hypertext

Sebagian besar pengguna link dalam *multimedia* interaktif berdasarkan penggunaan *hypertext* yang biasa disebut *hotword* atau *hotkey*. Hal ini berarti bahwa pengguna ingin mendapatkan informasi tentang kata atau sebagai kalimat 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 pada monitor. Untuk membedakan *hotword* dengan teks lain dapat dilakukan dengan memberikan warna atau font berbeda. Pointer mouse berubah pada saat berada diatas *hotword* dan lain-lain. Hal ini dapat memudahkan pengguna untuk mengenali teks yang mempunyai hubungan dengan informasi lebih lanjut. Untuk mengembangkan program *multimedia* yang berorientasi pada teks (text-oriented). Seperti panduan pengguna (manual reference), maka harus dipilih authoring yang mempunyai kemampuan *hypertext* yang baik.

2) 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 pengguna. Dalam hal ini, pengguna tidak dapat membedakan teks yang mempunyai informasi lebih lanjut atau tidak, sehingga fasilitas ini tidak memudahkannya. Tetapi, perancanggan dapat menghemat lebih banyak waktu karena program secara otomatis membuat link hypertext.

3) Text searching

Pencarian teks merupakan fitur yang memudahkan pengguna dengan memasukan suatu kata (atau memilih dari suatu daftar kata) dalam program multemedia, pengguna dapat dengan cepat memperoleh informasi yang berhubungan dengan kata tersebut. Hal ini serupa dengan pencaria teks dalam indeks suatu buku, kemudian dapat membuka halaman tertentu untuk memperoleh informasi lebih lanjut. Dalam program multemedia, proses pencarian dapat dilakukan lebih cepat dibandingkan dengan penggunaan indeks pada pencarian dalam buku. Bebera autoring tool dilengkapi dengan kemampuan pengguna untuk menggunakannya.

4) Import text dan *export text*

Beberapa teks yang digunakan dalam program *multimedia* mungkin telah ada dan dibuat sebelumnya dengan pengolah kata, atau dapat memasukkan data yang telah tersimpan dalam basis data. Bila file mempunyai ukuran besar maka tidak memerlukan waktu untuk memasukkan kembali ke dalam *multimedia* secara manual, sehingga diperlukan *autoring* yang dapat mengimpor teks dan basis data dari file lain. Pada umumnya program multimedia dapat membaca file tesks *ASCH. ASCH* adalah singkatan dari *American Standart Code For Information Interchange*, yaitu format karakter standar yang dapat digunakan pada semua computer dan program.

b. Image

Secara umum *image* atau grafik berarti still *image* (gambar tetap) seperti foto 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.

1. *Visible*

Kelompok *visible image* termasuk *drawing* (seperti *blueprint engineering drawing*, gambar denah, dan lain-lain). document (di-scan sebagai *image*), paiting (hasil scan atau dibuat dengan aplikasi paint program pada komputer), foto (hasil scan atau dimasukkan komputer langsung dengan kamera digital), dan *still frame* yang diambil dari kamera video.

2. Non-visible

Non-visible image adalah image yang tidak disimpan sebagai image, tetapi ditampilkan sebagai image. Contohnya: ukuran tekanan, ukuran temperature, dan tampilan meteran lainnya.

3. Abstrak

Image abstrak sebenarnya bukan image yang terdapat dalam kenyataan, tetapi dihasilkan oleh komputer seperti dalam perhitungan matematika. Fractal merupakan contoh image abstrak yang baik. Fungsi diskrit menghasilkan image yang tetap dalam skala tertentu, sedangkan fungsi kontinyu membentuk image seperti fading atau dissolving.

Beberapa aspek penting dari grafik adalah integrated drawing tool, clip art, impor grafik dan resolusi.

a) Integrated drawing tool

Pada umumnya autoring software mempunyai kemampuan untuk membuat gambar seperti garis. Grafik tersebut dibuat menggunakna mouse dan macam objek seperti garis, lingkaran, polygon dengan dukungan warna yang dihendaki. Hasil gafik pada umumnya sederhana, tetapi bermanfaat dalam program. Hal ini disebabkan oleh tidak dimilikinya kemampuan image yang komplek pada software.

b) Clip art

Clip art merupakan 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 pengguna dapat dengan mudah menggunakannya. Hal ini sangat membantu bila tidak mempunyai scanner atau alat lain yang dapat digunakan sabagai alat input untuk memasukkan gambar ke dalam komputer.

c) Impor grafik

Image yang baik biasanya berasal dari sumber lain, yaitu hasil fotografi yang baik. Secara umum image berarti gambar raster (halfone drawing), seperti foto. Beberapa paket authoring dapat mengimpor image dan format gambar tertentu seperti. PCX, BMP, JPG, GIF, dan lainnya.

Basis data karyawan dengan atribut seperti nama, alamat, dan lainnya akan afektif bila foto karyawan yang bersangkutan dapat ditampilkan. Demikian juga foto-foto seperti gedung dan lainlain sangat memerlukan penyimpanan yang besar. Hal inilah yang menyebabkan aplikasi multimedia disimpan dalam media penyimpanan yang cukup besar kapasitasnya seperti CD-ROM.

d) Resolusi

Resolusi grafik yang tinggi dapat menampilkan gambar dengan baik, tetapi tidak dapat diperoleh bila authoring software yang digunakan tidak mendukung resolusi tersebut. Demikian juga, beberapa software tidak dapat menampilkan resolusi lebih dari 640 x 480 pixel. Bila aplikasi sangat memerlukan resolusi yang tinggi, maka diperlukan authoring software yang mendukung resolusi dengan 256 warna atau true colour. Bila multimedia yang dihasilkan akan menggunakan bermacammacam graphic adapter, maka authoring tool harus mendukung bermacam-macam resolusi.

c. Animasi

Animasi berarti gerakan image atau video, seperti gerakan orang yang sedang melakukan suatu kegiatan, dan lain-lain. Konsep dari animasi adalah menggambarkan sulitnya menyajikan informasi dengan suatu gambar saja atau sekumpulan gambar. Demikian juga tidak dapat menggunakan teks untuk menerangkan informasi. Animasi seperti halnya film, dapat berupa *frame-based* atau *cast-based animation (animasiberbasis cast)* mencangkup pembuatan control dari masing-masing objek (kadang-kadang disebut cast member atau actor) yang bergerak melintasi latar belakang *(background)*. Hal ini merupakan bentuk umum animasi yang digunakan dalam permainan komputer dan *object-oriented software* untuk lingkungan window.

File animasi memerlukan penyimpanan yang jauh lebih besar dibandingkan dengan file gambar. Dalam authoring software, biasanya animasi mencangkup kemampuan *"recording"* dan *"playback"*. Fasilitas yang dimiliki oleh software animasi mencangkup *integrated animation tool*, *animation clip*, import animasi, *recording, playback,* dan *transition effect.*

1). Integrated animation tool

Walaupun sebagian besar authoring tool mendukung penggunaan animasi, tidak semuanya dapat digunakan untuk membuat dan menghasilkan file animasi. Bebera authoring tool menggunakan animasi yang dihasilkan dari software lain, atau komputer dengan platform lain, seperti **Macinosh Silicon Graphics,** dan lainnya. Untuk pembuatan aplikasi sederhana yang dilengkapi dengan animasi, dapat dipilih authoring tool yang menunjang pembuatan animasi. Namun bila diperlukan animasi yang lebih baik dengan software lain, maka pengguna *integrated animation tool* dapat memperoleh hasil yang baik.

2). Animation clip

Animation clip adalah *clip art* yang berisi file *animasi*. Banyak paket authoring dilengkapi dengan *library* dari animasi yang dapat digunakan pada komputer.

3). Impor file animasi

Seperti file grafik, multimedia memerlukan animasi dari file lain. Beberapa paket authoring dapat mengimpor animasi dari format file animasi tertentu sperti. FLL dan FLC. Disamping itu, image grafik dapat diimpor dari file grafik dan kemudian dibuat animasi, sehingga paket authoring dapat mengimpor image dari format grafik yang diperlukan.

Perlu diperhatikan juga bahwa authoring software yang digunakan dapat menampilkan warna dan resolusi dari file animasi yang diimpor.

4). Kemampuan recording dan *playback*

Tidak menjadi masalah file animasi yang digunakan, authoring tool harus dapat mengontrol bagaimana animasi direkam dan ditampilkan pada layar monitor. Contohnya, *playback control* harus dilengkapi dengan pilihan untuk *end user,* diantaranya*"pause","replay"*.Dan informasi *sekuens.*

5). Transition effect

Animasi dapat lebih menarik bila menggunakan efek transisi seperti : *fade in* dan *fade out, zoom,* rotasi objek dan warna. Namun, tidak sema authoring software, dilengkapi dengan kemampuan tersebut.

d. Audio

penyajian audio merupakan cara lain untuk lebih memperjelas pengertian suatu animasi. Contohnya, narasi merupakan kelengkapan dari penjelasan yang dilihat melalui video. Suara dapat lebih menjelaskan karakteristik suara gambar, misalnya music dan suara efek *(sound effect)*. Authoring software yang digunakan harus mempunyai kemampuan untuk mengontrol recording dan playback. Beberapa authoring software dapat merekam suara dengan macam-macam *sampling size* dan *sampling rate*. Bila narasi tidak perlu khawatir akan kemampuan *software* dengan audio apapun yang digunakan. Namun, perekam music yang baik memerlukan *sampling size* dan *sampling rate* yang tinggi. Beberapa macam authoring *software* dapat mengkonverensi suara, seperti format, WAV,. MID (MIDI), VOC, atau INS dan mungkin dihubungkan dengan sekuens dari animasi.

e. *Full-motion* dan *live video*

Full-motion video berhubung dengan penyimpanan sebagai video klip, sedangkan live video merupakan hasi pemrosesan yang diperoleh dari kamera. Beberapa rekaman menggunakan VCR, yang dapat menyajikan gambar bergerak dengan kualitas tinggi.

f. *Interactive link*

sebagian dari multimedia adalah interaktif, dimana pengguna dapat menekan atau objek pada layar monitor seperti tombola tau teks dan menyebabkan program melakukan perintah tertentu.

Interactive link dengan informasi yang berkaitan seringkalli dihubungkan secara keseluruhan sebagai hypermedia. Secara spesifik, dalam hal ini termasuk *hypertexs (hotword), hypergraphics* dan *hypersound* menjelaskan jenis informasi yang dihubungkan.

Interactive link diperlukan bila pengguna menunjukan pada suatu objek atau tombol supaya dapat mengakses program tertentu. Interactive link diperlukan untuk menggabungkan beberapa elemen multimedia sehingga menjadi informasi yang terpadu.

2.7 Tools untuk Pengembangan Perangkat Lunak

Adapun *software* yang digunakan untuk pembuatan Program tutorial iqra yaitu dengan Berbasis *Flash CS5.*

2.7.1 Flash

Flash merupakan *software* yang memiliki kemampuan menggambar sekaligus menganimasikannya, serta mudah dipelajari (M. Amarullah Akbar *et al,* 2008). Flash tidak hanya digunakan dalam pembuatan animasi, tetapi pada zaman sekarang ini flash juga banyak digunakan untuk keperluan lainnya seperti dalam pembuatan *game,* presentasi, membangun web, animasi pembelajaran, bahkan juga dalam pembuatan film.

Animasi yang dihasilkan flash adalah animasi berupa *file movie*. *Movie* yang dihasilkan dapat berupa grafik atau teks. Grafik yang dimaksud disini adalah grafik yang berbasis vektor, sehingga saat diakses melalui internet, animasi akan ditampilkan lebih cepat dan terlihat halus. Selain itu flash juga memiliki kemampuan untuk mengimpor *file* suara, video maupun *file* gambar dari aplikasi lain.

Flash adalah program grafis yang diproduksi oleh Macromedia corp, yaitu sebuah *vendor software* yang bergerak dibidang animasi web. Macromedia Flash pertama kali diproduksi pada tahun 1996. Macromedia flash telah diproduksi dalam beberapa versi. Versi terakhir dari Macromedia Flash adalah Macromedia flash 8.Sekarang Flash telah berpindah *vendor* menjadi Adobe.

Adobe adalah *vendor software* yang membeli Flash dari *vendor* sebelumnya yaitu Macromedia. Sejak itu, Macromedia Flash berganti nama menjadi Adobe Flash. Versi terbaru dari Adobe Flash adalah Adobe Flash CS5 Professional. Dalam pembuatan animasi ini penulis sudah menggunakan Adobe Flash CS5 Profesioanl sebagai aplikasinya. [8]

2.7.2 Adobe Flash CS6

Adobe Flash CS6 adalah salah satu aplikasi pembuat animasi yang cukup dikenal saat ini. Berbagai fitur dan kemudahan yang dimiliki menyebabkan Adobe Flash CS6 menjadi program animasi favorit dan cukup populer. Tampilan, fungsi dan pilihan palet yang beragam, serta kumpulan *tool* yang sangat lengkap sangat membantu dalam pembuatan karya animasi yang menarik. **[8]**

Flash seperti *software* gado-gado dimana didalamnya terdapat semua kelengkapan yang dibutuhkan. Mulai dari fitur menggambar, ilustrasi, mewarnai, animasi, dan *programming*. Kita dapat mendesain gambar atau objek yang akan kita animasikan langsung pada Flash. Fitur programming pada Flash menggunakan bahasa *ActionScript*.

ActionScript dibutuhkan untuk memberi efek gerak dalam animasi.*ActionScript* di flash pada awalnya memang sulit dimengerti jika seseorang tidak mempunyai dasar atau mengenal flash. Tetapi jika sudah mengenalnya, kita tidak bisa lepas dari *ActionScript* karena sangat menyenangkan dan dapat membuat pekerjaan jauh lebih cepat dan mudah.

2.8 Dasar-Dasar Penggunaan Adobe Flash CS6

2.8.1 Halaman Awal

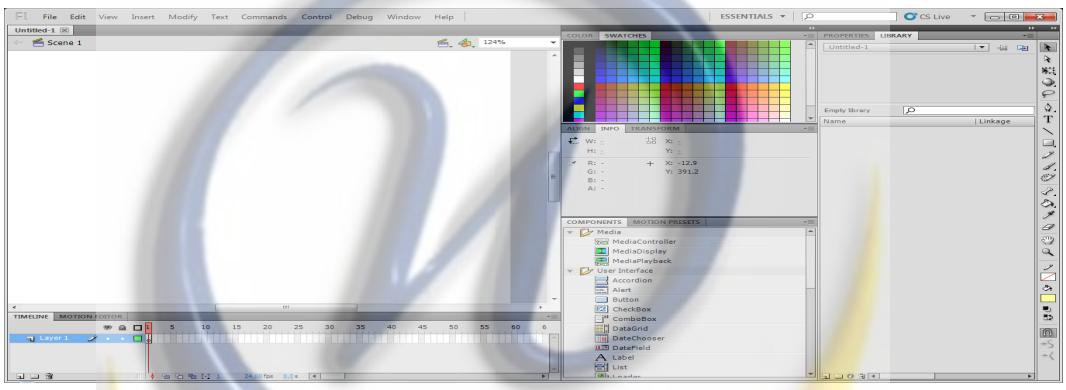
Halaman awal adalah tampilan yang pertama kali muncul ketika kita mengakses

Adobe Flash CS6 Professional. Cara mengakses Adobe Flash CS6 Professional pertama kali yaitu double klik pada *icon* yang ada di desktop atau lihat dari daftar program. Tampilan *start page* pertama kali membuka Adobe Flash CS6 Professional yaitu:

Gambar 2.2 Tampilan Start Page Adobe Flash CS6 Universitas

2.9 Jendela Utama

Jendela utama merupakan awal dari pembuatan program, pembuatannya dilakukan dalam kotak *movie* dan *stage* yang didukung oleh *tools* lainnya. Seperti yang pernah dijelaskan dalam sebuah tulisan "Jendela kerja flash terdiri dari panggung (*stage*) dan panel-panel. Panggung merupakan tempat objek diletakkan, tempat menggambar dan menganimasikan objek. Sedangkan panel disediakan untuk membuat gambar, mengedit gambar, menganimasi, dan pengeditan lainnya." (Diginnovac *et al*, 2008) Berikut ini adalah bentuk tampilan jendela utama pada Adobe Flash CS6.



Gambar 2.3 Jendela Utama

Keterangan gambar :

1. **Menu Bar** adalah kumpulan yang terdiri atas dasar menu-menu yang digolongkan dalam satu kategori. Misalnya menu file terdiri atas perintah *New*, *Open*, *Save*, *Import*, *Export*, dan lain-lain.
2. **Timeline** adalah sebuah jendela panel yang digunakan untuk mengelompokkan dan mengatur isi sebuah *movie*, pengaturan tersebut meliputi penentuan masa tayang objek, pengaturan *layer*, dan lain-lain.
3. **Stage** adalah area untuk berkreasi dalam membuat animasi yang digunakan untuk mengkomposisi *frame-frame* secara individual dalam sebuah *movie*.
4. **Toolbox** adalah kumpulan *tools* yang sering digunakan untuk melakukan seleksi, menggambar, mewarnai objek, memodifikasi objek, dan mengatur gambar atau objek.
5. **Properties** adalah informasi objek-objek yang ada di *stage*. Tampilan panel *properties* secara otomatis dapat berganti-ganti dalam menampilkan informasi

atribut-atribut *properties* dari objek yang terpilih.

6. **Panels** adalah sebagai pengontrol yang berfungsi untuk mengganti dan memodifikasi berbagai atribut dari objek dari animasi secara cepat dan mudah.

2.10 Toolbox

Fasilitas *Toolbox* seperti telah dijelaskan sekilas diawal adalah sekumpulan *tool* atau alat yang mempunyai fungsi-fungsi tersendiri untuk keperluan desain (M. Amarullah Akbar *et al,* 2008). Berikut penjelasan setiap *tool* yang terdapat pada Toolbox.

1. **Arrow Tool**

   Arrow Tool atau sering disebut *selection tool* berfungsi untuk memilih atau menyeleksi suatu objek.

2. **Subselection Tool**

   Subselection Tool berfungsi menyeleksi bagian objek lebih detail dari pada *selection tool*.

3. **Free Transform Tool**

   Free Transform Tool berfungsi untuk mentransformasi objek yang terseleksi.

4. **Gradient Transform Tool**

   Gradien Transform Tool berfungsi untuk mentransformasi warna dari *fill* objek yang terseleksi.

5. **Lasso Tool**

   Lasso Tool digunakan untuk melakukan seleksi dengan menggambar sebuah garis seleksi.

6. **Pen Tool**

   Pen Tool digunakan untuk menggambar garis dengan bantuan titik-titik bantu seperti dalam pembuatan garis, kurva atau gambar.

7. **Text Tool**

   Text Tool digunakan untuk membuat objek teks

8. **Line Tool**

   Line Tool digunakan untuk membuat atau menggambar garis.

9. **Rectangle Tool**

   Rectangle Tool digunakan untuk menggambar bentuk bentuk persegi panjang

atau bujur sangkar.

10. Oval Tool

Oval Tool digunakan untuk membuat bentuk bulat atau oval.

11. Poly Star Tool

Poly Star Tool digunakan untuk menggambar bentuk dengan jumlah segi yang diinginkan.

12. Pencil Tool

Pencil Tool digunakan untuk membuat garis

13. Brush Tool

Brush Tool digunakan untuk menggambar bentuk garis-garis dan bentuk bebas.

14. Ink bottle

Ink Bottle digunakan untuk mengubah warna garis, lebar garis, dan *style* garis atau garis luar sebuah bentuk.

15. Paintbucket Tool

Paintbucket Tool digunakan untuk mengisi area-area kosong atau digunakan untuk mengubah warna area sebuah objek yang telah diwarnai.

16. Eraser Tool

Eraser Tool digunakan untuk menghapus objek

17. Hand Tool

Hand Tool digunakan untuk menggeser tampilan *stage* tanpa mengubah pembesaran.

18. Zoom Tool

Zoom Tool digunakan untuk memperbesar atau memperkecil tampilan *stage.*

19. Stroke Color

Stroke Color digunakan untuk memilih atau memberi warna pada suatu garis.

20. Fill Color

Fill Color digunakan untuk memilih atau memberi warna pada suatu objek.

21. Black and white

Black and White digunakan untuk memilih warna hitam dan putih saja.

22. Swap Color

Swap Color digunakan untuk menukar warna *fill* dan *stroke* atau sebaliknya dari suatu gambar atau objek.

2.11 Library

Fungsi dari *library* adalah sebagai wadah untuk menyimpan program-program

terpisah yang sudah jadi, seperti tombol, objek grafis, audio, video, dan lain-lain. Berikut tampilan panel *library*.

Gambar 2.4 Panel Library

2.12 Unifted Modelling Language (UML)

Unified Modelling Language (UML) adalah "keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO)". [1]

Berikut ini definisi *Unified Modeling Language* (UML) menurut para ahli:

1. Menurut (Hend, 2006) "*Unified Modeling Language* (UML) adalah bahasa yang telah menjadi standard untuk visualisasi, menetapkan, membangun dan mendokumentasikan artifak suatu sistem perangkat lunak".
2. Menurut (Adi Nugroho : 2005). "*Unified Modeling Language* (UML) adalah alat bantu analisis serta perancangan perangkat lunak berbasis objek".
3. Menurut (Joomla dari http://soetrasoft.com : 2007). *"Unified Modeling Language* (UML) merupakan standard modeling language yang terdiri dari kumpulan-kumpulan diagram, dikembangkan untuk membantu para pengembang sistem dan *software* agar bisa menyelesaikan tugas-tugas seperti: Spesifikasi, Visualisasi, Desain Arsitektur, Konstruksi, Simulasi dan testing serta Dokumentasi".

Berdasarkan beberapa pendapat yang dikemukakan diatas dapat ditarik kesimpulan bahwa "*Unified Modeling Language* (UML) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk menvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis OO (*Object Oriented*)". [6]

UML dideskripsikan oleh beberapa diagram, diantaranya:

1. *Use Case* Diagram

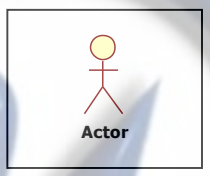
Use Case Diagram digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (*user*), sehingga pembuatan use case diagram lebih dititikberatkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah *use case diagram* merepresentasikan sebuah interaksi antara aktor dengan sistem.

Komponen-komponen yang terlibat dalam *use case* diagram :

2. Aktor

Pada dasarnya aktor bukanlah bagian dari *use case diagram*, namun untuk dapat terciptanya suatu *use case diagram* diperlukan aktor, dimana aktor tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat atau sistem lain) yang berinteraksi dengan sistem yang dibuat. Sebuah aktor mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari

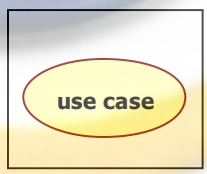
Sistem atau keduanya menerima dan memberi informasi pada sistem. Aktor hanya berinteraksi dengan *use case,* tetapi tidak memiliki kontrol atas *use case*. Aktor digambarkan dengan *stick man*.



Gambar 2.5 Aktor

3. Use case

Gambaran fungsionalitas dari suatu sistem, sehingga pengguna sistem paham dan mengerti kegunaan sistem yang akan dibangun.



Gambar 2.6 *Use Case*

Ada beberapa relasi yang terdapat pada *use case* diagram:

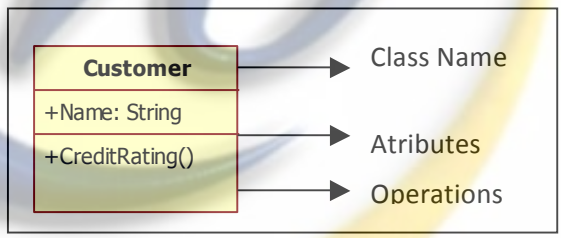
1. *Association*, menghubungkan link antar element.
2. *Generalization*, disebut juga pewarisan (*inheritance*), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.
3. *Dependency*, sebuah element bergantung dalam beberapa cara ke element lainnya.
4. *Aggregation*, bentuk *association* dimana sebuah elemen berisi elemen lainnya.

Tipe relasi yang mungkin terjadi pada *use case* diagram:

1. **<<*include*>>**, yaitu kelakuan yang harus terpenuhi agar sebuah *event* dapat terjadi, dimana pada kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya.
2. **<<*extends*>>**, kelakuan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan peringatan.
3. <<*communicates*>>, merupakan pilihan selama asosiasi hanya tipe *relationship* yang dibolehkan antara aktor dan *use case.*

4. *Class* Diagram

Class adalah sebuah spesifikasi yang 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 (metode/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain. [1]



Gambar 2.7 *Class*

Class memiliki tiga area pokok :

1. Nama (*Class Name*)
2. Atribut
3. Metode (*Operations*)

Pada UML, *class* digambarkan dengan segi empat yang dibagi beberapa bagian. Bagian atas merupakan nama dari *class*. Bagian yang tengah merupakan struktur dari *class* (atribut) dan bagian bawah merupakan sifat dari *class* (metode/operasi).

Atribut dan metode dapat memiliki salah satu sifat berikut :

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.

2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan *class* lain yang mewarisinya.
3. *Public*, dapat dipanggil oleh *class* lain.

Hubungan antar *Class* **:**

1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain.
2. Agregasi, yaitu hubungan yang menyatakan bagian ("terdiri atas..").
3. Pewarisan, yaitu hubungan hirarki antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metode *class* asalnya serta bisa menambahkan fungsionalitas baru. Sehingga *class* tersebut disebut anak dari *class* yang diwarisinya.
4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

5. *Statechart* Diagram

Menggambarkan semua *state* (kondisi) yang dimiliki oleh suatu objek dari suatu *class* dan keadaan yang menyebabkan *state* berubah. *Statechart diagram* tidak digambarkan untuk semua *class*, hanya yang mempunyai sejumlah *state* yang terdefinisi dengan baik dan kondisi *class* berubah oleh *state* yang berbeda.

State adalah sebuah kondisi selama kehidupan sebuah objek atau ketika objek memenuhi beberapa kondisi, melakukan beberapa aksi atau menunggu sebuah *event*. *State* dari sebuah objek dapat dikarakteristikkan oleh nilai dari satu atau lebih atribut-atribut dari *class*. *State* dari sebuah objek ditemukan dengan pengujian/pemeriksaan pada atribut dan hubungan dari objek. Notasi UML untuk *state* adalah persegipanjang/bujur sangkar dengan ujung yang dibulatkan.

Gambar 2.8 *Start state* dan *stop state*

Masing-masing diagram harus mempunyai satu dan hanya satu *start state* ketika objek mulai dibuat. Sebuah objek boleh mempunyai banyak *stop state*.

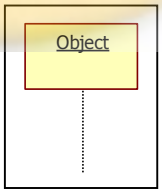
Gambar 2.9 *State transition*

Sebuah *state transition* dapat mempunyai sebuah aksi dan/atau sebuah kondisi penjaga (*guard condition*) yang terasosiasi dengannnya, dan mungkin juga memunculkan sebuah *event*. Sebuah aksi adalah kelakuan yang terjadi ketika *state transition* terjadi. Sebuah *event* adalah pesan yang dikirim ke objek lain di sistem. Kondisi penjaga adalah ekspresi *boolean* (pilihan Ya atau Tidak) dari nilai atribut-atribut yang mengijinkan sebuah *state transition* hanya jika kondisinya benar. Kedua aksi dan penjaga adalah kelakuan dari objek dan secara tipikal menjadi operasi.

6. *Sequence* Diagram

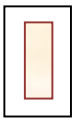
Menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem.

Dibawah merupakan simbol yang digunakan pada *sequence* diagram :



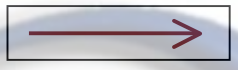
Gambar 2.10 *Object lifeline*

Object lifeline menunjukkan keberadaan dari sebuah objek terhadap waktu. Yaitu objek dibuat atau dihilangkan selama suatu periode waktu diagram ditampilkan, kemudian lifeline berhenti atau mulai pada titik yang tepat.



Gambar 2.11 *Activation*

Activation menampilkan periode waktu selama sebuah objek atau aktor melakukan aksi. Dalam *object lifeline*, *activation* berada diatas *lifeline* dalam bentuk kotak persegi panjang, bagian atas dari kotak merupakan inisialisasi waktu dimulainya suatu kegiatan dan yang dibawah merupakan akhir dari waktu.

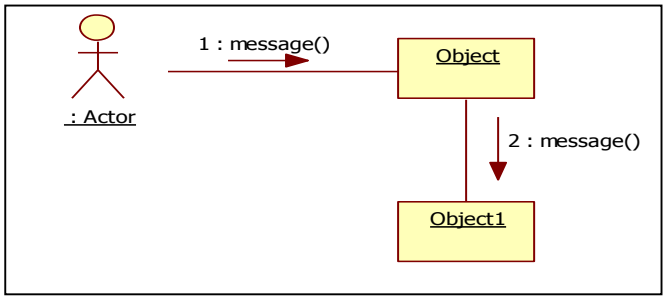


Gambar 2.12 *Message*

Message adalah komunikasi antar objek yang membawa informasi dan hasil pada sebuah aksi. *Message* menyampaikan dari *lifeline* sebuah objek kepada *lifeline* yang lain, kecuali pada kasus sebuah *message* dari objek kepada objek itu sendiri, atau dengan kata lain *message* dimulai dan berakhir pada *lifeline* yang sama.

7. *Collaboration* Diagram

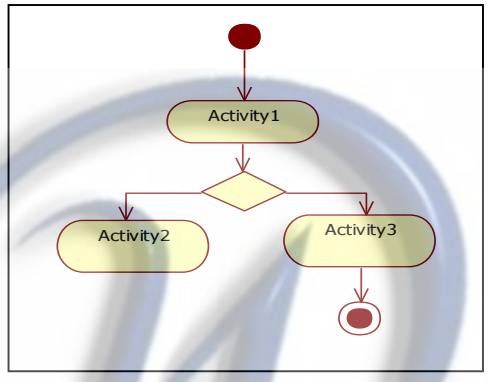
Diagram ini menggambarkan interaksi objek yang diatur objek sekelilingnya dan hubungan antara setiap objek dengan objek yang lainnya. Dalam menunjukkan pertukaran pesan, *collaboration diagram* menggambarkan objek dan hubungannya (mengacu ke konteks). Jika penekannya pada waktu atau urutan gunakan *sequence diagram*, tapi jika penekanannya pada konteks gunakan *collaboration diagram*.



Gambar 2.13 *Collaboration* Diagram

8. *Activity* Diagram

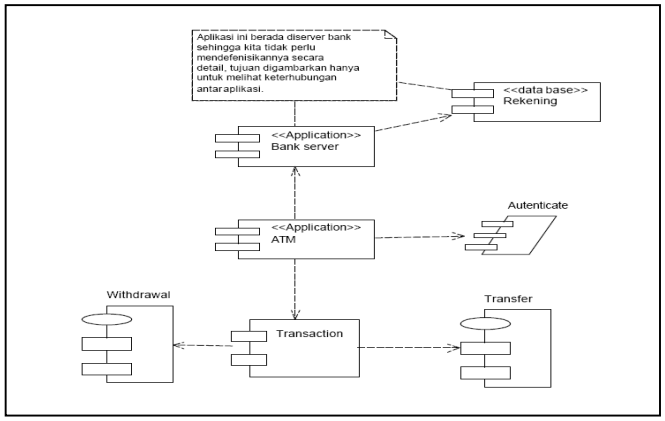
Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status. Pembuatan *activity diagram* pada awal pemodelan proses dapat membantu memahami keseluruhan proses. *Activity diagram* juga digunakan untuk menggambarkan interaksi antara beberapa *use case*.



Gambar 2.14 *Activity* Diagram

9. *Component* Diagram

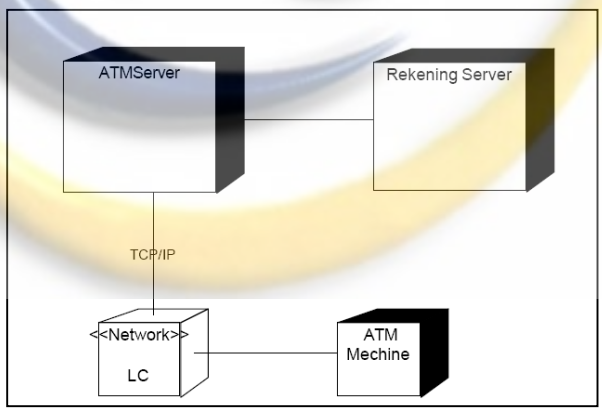
Menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*. Umumnya komponen terbentuk dari beberapa *class* 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.



Gambar 2.15 *Component* Diagram

10. *Deployment* Diagram

Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Di dalam *nodes*, *executeable component* dan objek yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh node tertentu dan ketergantungan komponen.



Gambar 2.16 *Deployment* Diagram