BAB II

LANDASAN TEORI

2.1. Pengajaran Berbantuan Komputer

Kemajuan ilmu pengetahuan dan teknologi memberi pengaruh yang besar dalam berbagai bidang pengetahuan yang menyebabkan bidang pendidikan menuntut sistem pendidikan dan latihan dapat dilaksanakan secara lebih efisien dan efektif.

Munculnya berbagai model pendekatan sistem pengajaran membuktikan bahwa manusia ingin memperbaiki metode belajar mengajar. Konsep ini kemudian diterapkan kedalam pengajaran berprogram yang sejalan dengan timbulnya mesin pengajaran sebagai media bantu dalam proses pengajaran, contohnya komputer.

Disini peranan komputer adalah sebagai salah satu media yang dapat membantu teknologi pengajaran. Sistem komputer digunakan untuk menyampaikan pengajaran kepada siswa, melalui interaksi langsung antara siswa dengan materi yang diprogram kedalam komputer. Inilah yang disebut dengan Pengajaran Berbantuan Komputer atau Computer Assisted Instruction (*CAI*).

PBK terdiri dari tiga macam perangkat :

1. Perangkat Keras (*hardware*), merupakan perangkat fisik yang berhubungan dengan sistem PBK, termasuk didalam komputer dan perlengkapannya
2. Perangkat Lunak (*software*), adalah semua program yang mendukung pelaksanaan PBK mulai dari perancangan sampai implementasi

3. Perangkat Manusia (*brainware*), adalah manusia yang terlibat dalam pembuatan dan pengoperasian seperti programmer, analis sistem dan pemakai.

PBK dibentuk oleh 3 komponen yang terdiri dari :

a. Domain Materi

   Domain materi adalah kumpulan materi dari perangkat ajar yang disajikan dengan cara memindahkan bahan-bahan dari buku referensi ke program komputer.

b. Sistem Pengendali Pengajaran

   Sistem Pengendali Pengajaran digunakan untuk mengendalikan kinerja perangkat ajar agar lebih terarah dan sistematis.

c. Antarmuka Pemakai

   Antarmuka merupakan bagian yang penting dalam suatu perangkat ajar karena bagian ini yang berinteraksi dan menyajikan materi kepada pemakai.

Ada beberapa macam model-model pengajaran berbantuan komputer, yaitu :

1. Model Tutorial

   Dalam model ini pola dasarnya mengikuti pengajaran berprogram tipe bercabang dimana mata pelajaran disajikan dalam unit-unit kecil.

2. Model Praktek dan Latihan

   Dalam menggunakan model ini hendaknya semua konsep, peraturan atau prosedur terlebih dahulu sudah dipelajari oleh siswa. Program akan membimbing siswa melalui serangkaian contoh yang kemudian meningkat pada ketangkasan dan kelancaran dalam mempergunakan ketrampilannya.

Model latihan ini sangat cocok untuk tujuan latihan pelajaran matematika, praktek menerjemahkan bahasa asing, latihan membentuk kosakata, dan lain-lain.

3. Model Penemuan

   Penemuan adalah istilah umum untuk menjelaskan kegiatan yang menggunakan pendekatan induktif dalam pengajaran, misalnya penyajian masalah-masalah yang dipecahkan oleh siswa dengan cara mencoba-coba. Model ini mendekati kegiatan belajar di laboratorium dan kegiatan belajar nyata yang biasa dilakukan diluar kelas.

4. Model Simulasi

   Dengan model ini siswa dihadapkan kepada situasi kehidupan nyata. Misalnya dalam kehidupan modern memperlihatkan perusahaan penerbangan yang menggunakan simulasi-simulasi penampilan pesawat terbang berkomputer canggih sebagai bagian integral dalam melatih terbang para awak pesawatnya.

5. Model Permainan

   Kegiatan permainan dapat melibatkan unsur-unsur simulasi, seperti halnya permainan bisa mengakibatkan unsur-unsur pengajaran bergantung pada ada tidaknya ketrampilan yang dipraktekkan dalam permainan itu sebagai kegiatan akademis.

2.2. Jenis Evaluasi Pengajaran

Bentuk evaluasi pengajaran dalam Pengajaran Berbantuan Komputer (PBK) yang mudah diterapkan adalah dalam bentuk evaluasi tes obyektif. Teknik tes obyektif yang dapat diterapkan antara lain :

1. Tes benar – salah (t*rue – false*)
2. Menjodohkan (*matching test*)
3. Tes pilihan ganda (*multiple choice test*)
4. Tes penilaian (*marking question*)

2.3. Pengertian Multimedia

Multimedia dilihat dari sudut pandang *user* adalah informasi dari komputer yang direpresentasikan melalui audio atau video dilengkapi dengan teks, image, grafik dan animasi. STE [8]

2.3.1. Unsur-unsur Multimedia

Unsur-unsur program multimedia : VAT [9]

1. Teks

   Unsur ini merupakan lapisan utama dari hampir seluruh program. Teks masih merupakan cara cepat untuk menyebarkan informasi, jadi program akan selalu memakainya.

2. Gambar

   Komputer multimedia dapat menampilkan gambar yang berkualitas seperti aslinya di monitor. Gambar yang dapat menggantikan kalimat-kalimat panjang.

3. Film

   Komputer dapat berubah menjadi sebuah televisi, atau komputer dapat menyimpan potongan-potongan film ke disk.

4. Animasi

   Animasi dapat mengekspresikan suatu unsur yang lebih bagus dari pada film. Beberapa orang dibidang industri menggunakan kata animasi untuk mendeskripsikan berbagai jenis gambar yang bergerak, termasuk film.

5. Suara

   Unsur suara digunakan untuk mendukung teks dan gambar-gambar, baik berupa gambar bergerak maupun yang tidak bergerak.

6. Kendali pelaksanaan program

   Kendali program multimedia yang memungkinkan pemakai untuk pindah kemana saja secara cepat.

2.3.2. Pengertian Multimedia Interaktif (*Multimedia Off Line*)

Multimedia interaktif merupakan media yang dianggap paling cocok untuk mengatasi permasalahan kekurang pahaman serta ketidaktertarikan anak-anak dalam mempelajari sesuatu. Oleh sebab itu penulis mencoba untuk mengimplementasikan permasalahan ini, sebab media interaktif ini memiliki kelebihan dari penggunaan elemen teks, gambar,animasi, film (video) dan suara yang dikemas dalam satu media menjadi satu kesatuan sehingga diharapkan anak-anak tertarik untuk mempelajarinya.

Multimedia pada perkembangan selanjutnya, produk desain grafis yang mengunakan komputer semakin komplek, salah satunya adalah multimedia, secara harfiah pengertian multimedia adalah :

Multi merupakan beberapa atau banyak

Media merupakan alat / jalur dalam menyampaikan berita, informasi atau pesan.

Jadi multi media adalah alat atau jalur yang digunakan untuk berkomunikasi dengan menggabungkan beberapa atau banyak alat. Istilah multimedia pertama kali dikenal dalam dunia teater yang mempertunjukan pergelaran dengan menggunakan pertunjukan gerak. musik dan monitor video untuk menambah dramatisasi suatu cerita tetapi sekarang ini pada umumnya multimedia dikenal sebagai suatu sistem. *Personal Computer* (PC) yang dapat mengintegrasikan teks, musik, narasi dan komunikasi data.

“Multimedia merupakan cara penyajian dan penyampaian infomasi secara menarik dan komunikatif, karena dengan multimedia kita dapat memadukan suara, gambar video teks dan grafik menjadi satu kesatuan, sehingga informasi yang disajikan /disampaikan akan lebih mudah diserap dan dimengerti”.

Media yang digunakan dalam produk multimedia ini sangat banyak dan beragam, tetapi secara garis besar dapat dikelompokkan menjadi dua bagian :

a. Media Lini Atas (*Above the Line Media*) seperti internet / jaringan informasi yang menghubungkan seluruh dunia
b. Media Lini Bawah (*Below the Line Media*) seperti CD-ROM yang meliputi CD interaktif, *Video Compact Disc* (VCD) dan lainnya.

2.4. Keunggulan Penyampaian Pesan Melalui Multimedia Interaktif

Interaksi	Pengguna secara aktif dapat berinteraksi langsung dengan komputer, pengguna menggunakan berbagai jenis peralatan komputer seperti keyboard, mouse, touchscreen penunjuk infrared dan sebagainya untuk melakukan interaksi dengan komputer
Animasi	Pengguna dapat melihat secara nyata bagaimana suatu proses itu terjadi
Suara	Mampu memberikan kesan yang lebih menarik dari multimedia dengan menekan tingkat kejenuhan dari pengguna. Efek suara mampu untuk mengikat pengguna dalam tempo yang cukup lama didepan komputer
Visual	Memberikan gambaran secara nyata tentang suatu situasi
Pengulangan	Pengguna dapat mengulang bagian-bagian yang diperlukan
Skip	Pengguna dapat bebas bergerak kebagian dan arah manapun dari tampilan yang disajikan
Pengekalan ingatan akan bertambah dari kajian	20% - Apa yang didengar 40% - Apa yang didengar dan dilihat 70% - Apa yang didengar, lihat dan buat.

2.5. Software

Software atau program desain yang digunakan dalam pembuatan media aplikasi tutorial cara cepat belajar membaca Al-Qur'an menggunakan metode Iqra' pada anak berbasis multimedia interaktif adalah sebagai berikut :

1. Corel Draw 12

   Program ini merupakan program untuk membuat gambar 2D dan program ini dipakai untuk mengubah file gambar JPG menjadi SWF dan format lainnya..

2. Macromedia Flash MX 2004

   Program ini dipakai untuk mengkomposisikan serta menjalankan perintah interaktif perpindahan halaman dengan penggabungan image ke bentuk animasi dan interaktif, hasilnya berupa file SWF.

2.6. Hardware

Hardware / perangkat keras yang digunakan dalam proses pembuatan aplikasi tutorial ini adalah :

1. Processor Intel Pentium 4 Celeron(tm) 2 GHz
2. Mainboard ECS P4VM2
3. Memory DDR 256 MB PC 2700
4. VGA Card Gforce 2 MX 400 64 MB
5. CD-ROM Drive 52x Samsung
6. Harddisk 60 GB Seagate 7200rpm
7. Speaker Multimedia Simbadda CST 7000
8. Monitor BRG 14'

9. Stabilizer 500 watt AVR 777
10. Keyboard + Mouse
11. Printer Epson C43SX
12. Sistem Operasi Windows XP Professional FD.

2.7. Rekayasa Perangkat Lunak

Pembuatan aplikasi ini meliputi tahapan-tahapan yang terdapat pada proses rekayasa perangkat lunak. Tahapan-tahapan yang harus dilalui adalah sebagai berikut :

1. Definisi dan Analisis Masalah

   Proses pembuatan perangkat lunak dimulai dengan definisi permasalahan dari perangkat lunak yang akan digunakan. Pada tahap ini kegiatan yang dilaksanakan antara lain :

   a. Penentuan topik dan materi yang akan dikembangkan dalam perangkat lunak.
   b. Tujuan umum dan sasaran pemakai hasil implementasi perangkat lunak.
   c. Sumber daya yang akan digunakan dalam perancangan dan implementasi baik perangkat lunak atau perangkat kerasnya.

2. Perancangan

   Tahap perancangan dibagi dua yaitu :

   a. Perancangan global

      Perancangan global berisi penjabaran tujuan dan materi yang sudah diterapkan dalam tahap definisi dan analisa masalah. Pada tahap ini

juga ditentukan bentuk perangkat lunak yang akan dikembangkan. Tahap ini bertujuan memberikan gambaran umum tentang sistem yang dirancang dan sebagai persiapan perancangan detail.

b. Perancangan detail

Perancangan detail menekankan pada kegiatan yang mengarah pada persiapan menuju tahap implementasi antara lain perancangan struktur menu, antarmuka, navigasi, aliran data, kamus data, spesifikasi proses dan basis data.

3. Implementasi

Implementasi merupakan tahap pengubahan hasil perancangan ke dalam bentuk bahasa pemrograman yang dimengerti oleh komputer.

4. Uji Coba

Uji coba dilakukan untuk mengetahui apakah hasil implementasi perangkat lunak yang telah dikembangkan sudah bebas dari kesalahan pemrograman baik berupa kesalahan logika maupun kesalahan sintak atau tujuan pembuatan perangkat lunak sudah tercapai.

5. Pemeliharaan

Perangkat lunak yang sudah dikembangkan masih mungkin akan mengalami perubahan-perubahan, misalnya perkembangan perangkat keras sehingga perlu adanya penyempurnaan perangkat lunak yang disesuaikan dengan perkembangan yang ada, adanya perkembangan kurikulum atau materi sehingga perlu disesuaikan materi ajarnya.

Pada setiap tahapan rekayasa perangkat lunak dilakukan proses validasi, sehingga jika ada suatu tahapan terdapat kesalahan akibat tahap sebelumnya, maka proses dapat kembali ke tahap sebelumnya.

2.8. Definisi UML (*Unified Modeling Language*)

UML adalah bahasa grafis untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML berorientasi objek, menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi, pemaduan beberapa notasi di beragam metodologi, usaha bersama dari banyak pihak, didukung oleh kakas – kakas yang diintegrasikan lewat XML. Standar UML dikelola oleh OMG (*Object Management Group*).

UML adalah bahasa pemodelan untuk menspesifikasikan, memvisualisasikan, membangun dan mendokumentasikan artifak-artifak dari sistem. BAM [1]

1. Didalam *system intensive process*, metode diterapkan sebagai proses untuk menurunkan atau mengevolusikan sistem
2. Sebagai bahasa, UML digunakan untuk komunikasi yaitu alat untuk menangkap pengetahuan mengenai satu subyek dan mengekspresikan pengetahuan (*sintax*) yang memperdulikan subyek untuk maksud komunikasi. Subyek adalah sistem yang dibahas
3. Sebagai bahasa pemodelan, UML fokus pada pemahaman subyek melalui formulasi model dari subyek. Model memuat pengetahuan pada subyek, dan aplikasi dari pengetahuan ini berkaitan dengan intelejensia

4. Berkaitan dengan unifikasi.UML memadukan praktek rekayasa terbaik sistem informasi dan industri, meliputi beragam tipe sistem (perangkat lunak dan non perangkat lunak), *domain* (bisnis, perangkat lunak) dan proses siklus hidup
5. Begitu diterapkan untuk menspesifikasikan sistem, UML dapat digunakan untuk mengkomunikasikan “apa” yang diperlukan dari sistem dan “bagaimana” sistem dapat direalisasikan
6. Begitu diterapkan untuk menvisualisasikan sistem, UML dapat digunakan untuk menjelaskan sistem secara visual sebelum direalisasikan
7. Begitu diterapkan untuk membangun sistem, UML dapat digunakan untuk memandu realisasi sistem serupa dengan “*blueprint*”
8. Begitu diterapkan untuk mendokumentasikan sistem, UML dapat digunakan untuk menangkap pengetahuan mengenai sistem pada seluruh siklus hidup.

UML bukanlah :

1. Bahasa pemrograman visual, tapi bahasa pemodelan visual
2. Spesifikasi kakas, tapi spesifikasi bahasa pemodelan
3. Proses, tapi yang memungkinkan proses – proses.

2.9. Tujuan UML

Tujuan utama perancangan UML adalah :

1. Menyediakan bahasa pemodelan visual yang ekspresif dan siap pakai untuk mengembangkan dan pertukaran model-model yang berarti

2. Menyediakan mekanisme perluasan dan spesifikasi untuk memperluas konsep-konsep inti
3. Mendukung spesifikasi independen bahasa pemrograman dan proses pengembangan tertentu
4. Menyediakan basis formal untuk pemahaman bahasa pemodelan
5. Mendorong pertumbuhan pasar kakas berorientasi objek
6. Mendukung konsep-konsep pengembangan level lebih tinggi seperti komponen, kolaborasi, *framework* dan *pattern*.

2.9.1. Diagram dan Teknik Pemodelan UML

Diagram mengemukakan banyak hal, penggunaan notasi yang terdefinisi baik dan ekspresif adalah penting pada proses pengembangan perangkat lunak, yaitu :

1. Notasi standar memungkinkan pengembang mendeskripsikan skenario atau rumusan arsitektur dan kemudian mengkomunikasikan secara tidak ambigu
2. Notasi yang bagus membebaskan otak untuk berkonsentrasi pada masalah-masalah yang lebih lanjut
3. Notasi yang baik memungkinkan mengeleminasi keperluan pemeriksaan konsistensi dan kebenaran keputusan keputusan dengan menggunakan *tool* terotomatisasi.

2.9.2 Diagram struktur

Diagram ini untuk memvisualisasi, menspesifikasikan, membangun dan mendokumentasikan aspek statik dari sistem. Diagram struktur di UML terdiri dari :

1. Diagram kelas (*Class diagram*)
2. Diagram objek (*Objek diagram*)
3. Diagram komponen (*Component diagram*)
4. Diagram *deployment* (*Deployment diagram*)

2.9.3 Diagram perilaku

Diagram ini untuk memvisualisasi, menspesifikasi, membangun dan mendokumentasikan aspek dinamis dari sistem. Diagram perilaku di UML terdiri dari :

1. Diagram *use-case* (*Use case diagram*)
2. Diagram sekuen (*Sequence diagram*)
3. Diagram kolaborasi (*Collaboration diagram*)
4. Diagram *state*chart (*Statechart diagram*)
5. Diagram aktivitas (*Activity Diagram*)

1. Diagram kelas (*Class diagram*)

Diagram ini menunjukkan sekumpulan kelas, *interface* dan kolaborasi dan keterhubungannya. Diagram kelas ditujukan untuk pandangan statik terhadap sistem.

2. Diagram objek (*Object diagram*)

Diagram ini menunjukkan sekumpulan objek dan keterhubungannya. Diagram ini menunjukkan potongan statik dari instan –instan yang ada di diagram kelas. Diagram ini untuk memperlihatkan satu prototipe atau kasus tertentu yang mungkin terjadi. Diagram obyek menyediakan notasi grafis formal guna memodelkan objek, kelas, dan saling keterhubungan. Diagram objek berguna untuk *abstract modeling* dan perancangan program –program sesungguhnya. Pada pendekatan ini, bentukan dasar dari sistem perangkat lunak adalah objek atau kelas. Kelas adalah deskripsi dari objek –objek yang *common*. Setiap objek mempunyai identitas,*state* dan perilaku.

3. Diagram use-case (*Use case diagram*)

Diagram ini menunjukkan sekumpulan kasus fungsional dan *actor* (jenis kelas khusus) dan keterhubungannya.

4. Diagram sekuen (*Sequence diagram*)

Diagram ini menunjukkan interaksi yang terjadi antar objek. Diagram ini merupakan pandangan dinamis terhadap sistem. Diagram ini menekankan pada basis keberurutan waktu dari pesan-pesan yang terjadi.

5. Diagram kolaborasi (*Collaboration diagram*)

Diagram ini juga merupakan diagram interaksi. Diagram ini menekankan pada organisasi struktur dari objek-objek yang mengirim dan menerima pesan.

6. Diagram *state*chart (*Statechart diagram*)

Diagram ini adalah *state-machine diagram*, berisi state, transisi, kejadian dan aktivitas. *Statechart* merupakan pandangan dinamis dari sistem. Diagram ini penting dalam memodelkan perilaku antarmuka, kelas, kolaborasi dan

menekankan pada urutan kejadian. Penting untuk sistem reaktif yang dipicu kejadian di dunia nyata.

7. Diagram aktivitas (*Activity diagram*)

Diagram ini untuk menunjukkan aliran aktivitas di sistem. Diagram ini adalah pandangan dinamis terhadap sistem. Diagram ini penting untuk memodelkan fungsi sistem dan menekankan pada aliran kendali di antara objek – objek.

8. Diagram komponen (*Component diagram*)

Diagram ini menunjukkan organisasi dan kebergantungan di antara sekumpulan komponen. Diagram ini merupakan pandangan *static* terhadap implementasi sistem.

9. Diagram deployment (*Deployment diagram*)

Diagram ini menunjukkan konfigurasi pemrosesan saat jalan dan komponen komponen yang terdapat didalamnya. Diagram ini merupakan pandangan *static* dari arsitektur. Pilihan model dan diagram yang digunakan dipengaruhi oleh bagaimana persoalan ditangani dan bagaimana solusi dibentuk. Abstraksi, fokus pada relevan sambil mengabaikan rincian rincian yang tidak relevan merupakan kuncinya. Karena itu, setiap sistem kompleks perlu didekati melalui sekumpulan pandangan model yang hampir independent. Tidak ada satu pandangan tunggal yang memadai untuk sistem kompleks.

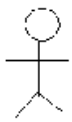
2.10. Notasi dalam UML (*Unified Modeling Language*)

UML menyediakan beberapa notasi dan artifak standar yang bisa digunakan sebagai alat komunikasi bagi para pelaku dalam proses analisis dan desain.

Artifak di dalam UML didefinisikan sebagai informasi dalam berbagai bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat lunak.Yang perlu diperhatikan untuk menjaga konsisten antar artifak, selama proses analisis dan desain adalah bahwa setiap perubahan yang terjadi pada satu artifak harus juga dilakukan pada artifak sebelumnya.

1. Actor

Actor adalah segala sesuatu yang berinteraksi dengan sistem aplikasi komputer. Jadi *actor* ini bisa berupa orang, perangkat keras, atau mungkin objek lain dalam sistem yang sama. Biasanya yang dilakukan *actor* adalah memberikan informasi pada sistem dan atau memerintahkan sistem untuk melakukan sesuatu.

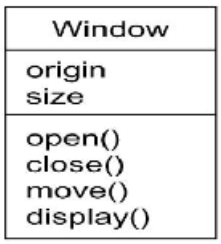


Gambar 2.1 Notasi Actor

2. Class

Class merupakan pembentuk utama dari sistem berorientasi objek karena *class* menunjukkan kumpulan objek yang memiliki atribut dan operasi yang sama. *Class* digunakan untuk mengabstraksikan elemen-elemen dari sistem yang sedang dibangun. *Class* bisa untuk merepresentasikan baik perangkat lunak maupun perangkat keras, baik konsep maupun benda nyata. Notasi *class* berbentuk persegi panjang berisi 3 bagian. Persegi panjang paling atas untuk nama *class*, persegi panjang paling bawah untuk operasi dan persegi panjang tengah untuk atribut. Atribut digunakan untuk menyimpan informasi, nama atribut menggunakan kata benda yang bisa dengan jelas merepresentasikan informasi yang disimpan

didalamnya. Operasi menunjukkan sesuatu yang bisa dilakukan oleh objek dan menggunakan kata kerja.



Gambar 2.2. Notasi class

3. Interface

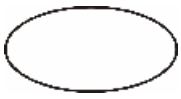
Interface merupakan kumpulan operasi tanpa implementasi dari suatu *class*. Implementasi operasi dalam *interface* dijabarkan oleh operasi dalam *class*. Oleh karena itu keberadaan *interface* selalu disertai oleh *class* yang mengimplementasikan operasinya. *Interface* ini merupakan salah satu cara mewujudkan prinsip enkapsulasi dalam objek.



Gambar 2.3 Notasi Interface

4. Use Case

Use case menjelaskan urutan kegiatan yang dilakukan *actor* dan sistem untuk mencapai suatu tujuan tertentu walaupun menjelaskan kegiatan. Namun *use case* hanya menjelaskan “apa” yang dilakukan oleh *actor* dan sistem, bukan “bagaimana” *actor* dan sistem melakukan kegiatan tersebut.



Gambar 2.4 Notasi use case

5. Interaction

Interaction digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar obyek. Biasanya *interaction* ini dilengkapi juga dengan teks bernama *operation signature* yang tersusun dari nama operasi, parameter yang dikirim dan tipe parameter yang dikembalikan.



Gambar 2.5 Notasi Interaction

6. Package

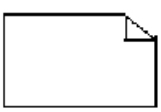
Package adalah kontainer atau wadah konseptual yang digunakan untuk mengelompokkan elemen-elemen dari sistem yang sedang di bangun, sehingga bisa dibuat model yang lebih sederhana. Tujuannya adalah untuk mempermudah penglihatan (*visibility*) dari model yang sedang dibangun.



Gambar 2.6 Notasi Package

7. Note

Note digunakan untuk memberikan keterangan dan komentar tambahan dari satu elemen sehingga bisa langsung terlampir dalam model. *Note* ini bisa ditampilkan ke semua elemen notasi yang lain.



Gambar 2.7 Notasi Note

8. Dependency

Dependency merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain. Elemen yang ada dibagian tanda panah adalah elemen yang tergantung pada elemen yang ada di bagian tanpa tanda panah. Terdapat 2 *stereotype* dari *dependency*, yaitu *Include* dan *Extend.* Individu menunjukkan bahwa suatu kegiatan dari elemen (yang ada digaris tanpa panah) memicu eksekusi bagian dari elemen lain (yang ada digaris dengan panah), misalnya untuk notasi A → B operasi yang ada di *class* A memicu dieksekusinya operasi yang ada di *class* B. *Extended* menunjukkan bahwa suatu bagian dari elemen di garis tanpa bisa disisipkan ke dalam elemen yang ada di garis dengan panah, misalnya untuk notasi A→ B suatu fungsi dari *use case* A bisa disisipkan ke dalam *use case* B atau dengan kata lain A optional untuk B. Kedua *stereotype* ini direpresentasikan dengan menambahkan *text include* atau *extend* di notasi *dependency*.



Gambar 2.8 Notasi Dependency

9. Boundary Class

Boundary class adalah *class* yang menghubungkan *user* dengan sistem. Oleh karena itu sering disebut juga sebagai *user interface class*.



Gambar 2.9 Notasi Boundary Class

10. Control class

Control class adalah *class* yang mengkoordinasi aktivitas dalam sistem. *Class* ini menghubungkan *boundary class* dengan *entity class*.



Gambar 2.10 Notasi Control class

11. Entity Class

Entity Class adalah *class* yang menghubungkan dengan data atau informasi yang digunakan oleh sistem. *Entity class* ini adalah *class* yang menyimpan dan mengelola data.



Gambar 2.11 Notasi Entity class