

BAB 2

LANDASAN TEORI

2.1 Teori Umum

Untuk mendukung proses analisa dan perancangan *e-learning* pada *Avicenna Education*, penjelasan teori secara umum akan dijelaskan secara *detail* dalam bab ini. Teori umum yang akan dijelaskan dalam bab ini meliputi sistem, informasi, dan sistem informasi.

2.1.1 Sistem

Terminologi sistem yang digunakan dalam disiplin ilmu yang berbeda dapat memiliki makna yang berbeda pula. Meskipun berbeda ada kesamaan secara umum yang menggambarkan definisi sistem. Secara garis besar definisi sistem adalah sekumpulan obyek yang memiliki hubungan diantara elemen-elemennya.

Satzinger, Jackson, Burd (2012: 241) mendefinisikan sistem sebagai kumpulan beberapa elemen yang saling terkait dan bekerja secara bersama-sama untuk mencapai hasil tertentu. Definisi lain, sistem dapat diartikan sebagai suatu kumpulan sub-sub sistem yang saling terhubung dan bekerjasama untuk mencapai tujuan tertentu dalam memproses suatu *input* sehingga menghasilkan output yang mempunyai nilai lebih (Taufiq, 2013: 11). Sedangkan O'Brien, dan Marakas (2013: 26) berpendapat bahwa sistem merupakan sekumpulan elemen yang saling berhubungan dengan batasan yang jelas dan bekerjasama untuk mencapai tujuan dengan menerima *input* dan menghasilkan output dalam proses yang teratur. Berikut tiga fungsi dasar sistem:

1. *Input*: Melibatkan pengambilan dan perakitan elemen-elemen yang masuk ke sistem untuk diproses.
2. *Proses*: Melibatkan proses transformasi yang mengubah *input* ke *output*.
3. *Output*: Melibatkan perpindahan elemen yang telah diproduksi oleh proses transformasi ke tujuan akhirnya.

Berdasarkan definisi-definisi di atas dapat disimpulkan bahwa sistem adalah kumpulan elemen atau komponen dasar yang saling terhubung dan terkait untuk bekerjasama dalam memproses suatu *input* sehingga menghasilkan *output* atau tujuan yang mempunyai nilai lebih.

2.1.2 Informasi

Informasi secara umum dapat didefinisikan sebagai ide atau gagasan yang dikomunikasikan untuk tujuan tertentu. Informasi diolah dan diproses untuk suatu tujuan tertentu, dan tujuan tersebut digunakan untuk menghasilkan keputusan yang tepat.

Menurut Rainer, Prince dan Cegielski (2014: 14), informasi mengacu pada data yang telah disusun sehingga data tersebut bermakna dan bernilai bagi penerimanya. Sedangkan menurut O'Brien, dan Marakas (2013: 38), informasi adalah data yang telah diubah menjadi konteks yang berarti dan berguna bagi para pengguna akhir.

Berdasarkan penjelasan di atas dapat ditarik kesimpulan bahwa informasi merupakan data yang telah diolah menjadi komponen yang penting dan berarti dalam sebuah organisasi atau perusahaan yang dapat digunakan untuk kemajuan organisasi atau perusahaan.

2.1.3 Sistem Informasi

Sistem informasi dirancang dan dikembangkan untuk memberikan solusi sesuai dengan permintaan tertentu. Dengan demikian struktur dan cara kerja sistem informasi unik berdasarkan tujuan dan permintaan yang harus dipenuhi.

Menurut Rainer, Prince dan Cegielski (2014:7), sistem informasi merupakan serangkaian kegiatan mengumpulkan, memproses, menyimpan, menganalisa, dan menyebarkan informasi untuk tujuan yang spesifik. Sedangkan menurut O'Brien, dan Marakas (2013:5), sistem informasi merupakan kombinasi orang, *hardware*, *software*, jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi.

Berdasarkan penjelasan di atas maka dapat disimpulkan bahwa sistem informasi merupakan proses berbagai komponen yang saling terkait dan

bekerja sama dalam mengumpulkan, memproses, menyimpan dan menyebarkan informasi dalam suatu organisasi atau perusahaan.

2.2 Teori Khusus

Selain penjelasan teori umum, adapun teori-teori secara khusus yang dapat mendukung analisis dan perancangan *e-learning* di *Avicenna Education*. Teori khusus yang akan dijelaskan pada sub bab ini adalah mengenai *e-learning*, *ADDIE framework*, diagram-diagram pada *United Model Language (UML)*, *user interface*, bahasa pemrograman *PHP*, *MySQL*, *javascript*, aplikasi *framework*, *CodeIgniter (CI)*, *web application*, *notepad++*, dan *XAMPP*.

2.2.1 *E-learning*

Learning yang berarti pembelajaran merupakan kegiatan yang selalu terjadi di kehidupan manusia yang sangat penting untuk mencapai keberhasilan dimasa yang akan datang. Menurut Slameto (2003:2) pembelajaran merupakan proses bisnis seorang individu untuk memperoleh perubahan perilaku baru secara keseluruhan, sebagai hasil dari pengalamannya sendiri dalam berinteraksi dengan lingkungan. Menurut Sugandi (2004:9) menyatakan bahwa pembelajaran terjemahan dari kata “*instruction*” yang berarti *self instruction* (dari internal) dan *external instructions* (dari eksternal). Pembelajaran yang bersifat eksternal antara lain datang dari guru yang disebut *teaching* atau pengajaran. Dalam pembelajaran yang bersifat eksternal prinsip-prinsip belajar dengan sendirinya akan menjadi prinsip-prinsip pembelajaran. Dengan teknologi yang berkembang saat ini dapat dimanfaatkan oleh lembaga pendidikan agar lebih efektif dan efisien dalam melakukan kegiatan pembelajaran, yang dalam hal ini disebut dengan *e-learning*.

E-learning atau pembelajaran elektronik pertama kali diperkenalkan oleh universitas Ilionis di Urbana-Champaign dengan menggunakan sistem instruksi berbasis komputer (*computer assisted instruction*) dan komputer bernama *PLATO (Programmed Logic for Automated Teaching Operations)*. Sejak saat itu, perkembangan *e-learning* berkembang sejalan dengan perkembangan dan kemajuan teknologi. Perkembangan *e-learning* diawali dari tahun 1990 pada masa *CBT (Computer-Based Training)* dimana mulai

bermunculan aplikasi *e-learning* yang berjalan dalam PC standalone ataupun berbentuk kemasan CD-ROM. Isi materi dalam bentuk tulisan maupun multimedia (Video dan Audio) dalam format mov, mpeg-1, atau avi.

E-learning tersusun dari dua bagian, yaitu ‘e’ yang merupakan singkatan dari ‘*electronic*’ yang berarti ‘elektronik’ dan kata ‘*learning*’ yang berarti ‘pembelajaran’. Dari hal tersebut, dapat dilihat bahwa *e-learning* merupakan pembelajaran yang menggunakan media elektronik. *E-learning* dapat dilakukan dari mana saja asalkan terhubung dengan internet. Dari hal demikian, tentu akan banyak manfaat yang akan didapat oleh sebuah lembaga pendidikan, maka dari itu di Indonesia sendiri mulai menerapkan pembelajaran dengan metode *e-learning* tersebut. Di Indonesia mulai mengembangkan *e-learning* dari tahun 2000. Dengan adanya *e-learning* diharapkan dapat membantu untuk mempermudah proses pembelajaran sehingga lebih maksimal dalam mencapai tujuan pembelajaran. Dari definisi di atas, yang membedakan *e-learning* dengan *learning* pada umumnya adalah penggunaan teknologi elektronik dan internet.

2.2.1.1 Pengertian *E-learning*

E-learning memiliki definisi yang berbeda-beda menurut beberapa orang. Rusman (2012, p293) mendefinisikan *e-learning* merupakan segala aktivitas belajar yang menggunakan bantuan teknologi elektronik.

Clark dan Mayer (2016, p8) mendefinisikan *e-learning* sebagai instruksi yang disampaikan pada perangkat digital (seperti komputer desktop, komputer laptop, tablet, atau ponsel pintar) yang dimaksudkan untuk mendukung pembelajaran.

E-learning merupakan proses belajar secara efektif yang dihasilkan dengan cara menggabungkan penyampaian materi secara digital yang terdiri dari dukungan dan layanan dalam belajar (Karwati, 2014).

Menurut Horton (2012, p1), “*E-learning is the use of electronic technologies to create learning experiences*”, yang berarti *E-learning* adalah suatu penggunaan teknologi elektronik untuk menciptakan pengalaman belajar.

Menurut Effendi dan Zhuang (2005:6), *E-learning* adalah sebuah kegiatan pembelajaran yang menggunakan media elektronik atau teknologi informasi. Disini media elektronik yang digunakan adalah komputer yang terhubung dengan *internet* sebagai teknologi informasinya.

Sejalan dengan itu, Onno W. Purbo (dalam Amin, 2004) menjelaskan bahwa istilah "e" dalam *e-learning* adalah segala teknologi yang digunakan untuk mendukung usaha-usaha pengajaran lewat teknologi elektronik internet. Internet, satelit, tape audio/video, tv interaktif, dan CD-ROM adalah sebagian dari media elektronik yang digunakan.

Dari beberapa pendapat para ahli tersebut, dapat disimpulkan bahwa *e-learning* merupakan pemanfaatan teknologi elektronik untuk mendukung kegiatan pembelajaran.

2.2.1.2 Jenis-Jenis *E-learning*

Menurut William Horton (2012, p2) terdapat beberapa jenis *e-learning* yaitu sebagai berikut:

1. *Standalone Course*

Latihan yang dilakukan hanya oleh satu orang. Mereka melakukannya sendiri tanpa adanya interaksi yang terjadi dengan guru ataupun teman.

2. *Learning Games and Simulation*

Pembelajaran yang dilakukan dengan menampilkan simulasi dalam beraktifitas yang membutuhkan eksplorasi dan arahan dalam menyelesaikan pembelajaran.

3. *Mobile Learning*

Pembelajaran yang dapat dilakukan dimanapun diseluruh dunia dengan menggunakan perangkat mobile, seperti *mobile phone* dan *tablet*, mereka yang belajar melalui *mobile* berpartisipasi dalam *classroom courses* dan *standalone*. Mereka juga dapat berpartisipasi dalam aktifitas dimana mereka berinteraksi dengan objek maupun orang yang mereka temui ketika dalam proses pembelajaran.

4. *Social Learning*

Pembelajaran yang dilakukan dengan melakukan interaksi dalam sebuah komunitas yang telah berpengalaman. Komunikasi antara partisipan membutuhkan media *social-networking*, seperti diskusi *online*, *blogging* dan pesan teks.

5. *Virtual-classroom courses*

Kelas yang dibentuk seperti kelas sesungguhnya dalam bentuk *online*, dengan adanya latihan membaca, presentasi dan diskusi dengan menggunakan forum ataupun dengan media lain, dan ada tugas, termasuk pertemuan *online* yang *ter-synchron*.

2.2.1.3 Tipe *E-learning*

Menurut Rosen (2009, p60), *e-learning* terbagi ke dalam dua tipe yang berbeda, yaitu *synchronous* dan *asynchronous*.

1. Dalam pelatihan *synchronous*, siswa dan guru bertemu pada waktu yang telah ditentukan untuk sesi yang dipimpin instruktur.
2. Dalam pelatihan *asynchronous*, siswa menggunakan materi yang tersedia melalui *web* yang cukup lengkap untuk digunakan kapan saja, memungkinkan siswa untuk mengaksesnya sesuai kebutuhan.

2.2.1.4 Komponen *E-learning*

Menurut Romi Satria (2008), komponen yang membentuk *e-learning* ada tiga yaitu sebagai berikut:

1. **Infrastruktur *E-learning*:** Infrastruktur *e-learning* dapat berupa *personal computer* (PC), jaringan komputer, internet dan perlengkapan multimedia. Termasuk didalamnya peralatan *teleconference* apabila kita memberikan layanan *synchronous learning* melalui *teleconference*.
2. **Sistem dan Aplikasi *E-learning*:** Sistem perangkat lunak yang mem-virtualisasi proses belajar mengajar konvensional. Bagaimana manajemen kelas, pembuatan materi atau konten, forum diskusi, sistem penilaian (rapor), sistem ujian *online* dan segala fitur yang berhubungan dengan manajemen proses belajar mengajar. Sistem perangkat lunak tersebut sering disebut dengan

Learning Management System (LMS). LMS banyak yang *open source* sehingga bisa kita manfaatkan dengan mudah dan murah untuk dibangun di sekolah dan universitas kita.

3. **Konten *E-learning*:** Konten dan bahan ajar yang ada pada *e-learning system* (*Learning Management System*) konten dan bahan ajar ini bisa dalam bentuk *Multimedia-based Content* (konten berbentuk multimedia interaktif) atau *Text-based Content* (konten berbentuk teks seperti pada buku pelajaran biasa). Biasa disimpan dalam *Learning Management System* (LMS) sehingga dapat dijalankan oleh siswa kapanpun dan dimanapun. Sedangkan aktor yang ada dalam pelaksanaan *e-learning* boleh dikatakan sama dengan proses belajar mengajar konvensional, yaitu perlu adanya guru (instruktur) yang membimbing, dan siswa yang menerima bahan ajar dan *administrator* yang mengelola *administrasi* dan proses belajar mengajar.

2.2.1.5 Manfaat *E-learning*

Dalam pembelajarannya, *e-learning* memiliki beberapa manfaat bagi penggunaannya. Berikut manfaat *e-learning* menurut para ahli.

Menurut Pranoto, dkk (2009:309) manfaat *e-learning* antara lain sebagai berikut:

1. Meningkatkan partisipasi aktif.
2. Meningkatkan kemampuan belajar mandiri.
3. Meningkatkan kualitas materi pendidikan serta juga pelatihan.
4. Meningkatkan kemampuan untuk dapat menampilkan informasi dengan perangkat teknologi informasi, yang mana dengan perangkat biasa akan sulit dilakukan.

Menurut Fioriello (2009) manfaat *e-learning* bagi siswa yaitu sebagai berikut:

1. Siswa dapat mendaftar dan mendapatkan semua program *e-learning* di *World Wide Web*.

2. Siswa dapat belajar secara mandiri kapan dan dimanapun mereka berada, karena *e-learning* dapat diakses setiap waktu dimanapun mereka berada.
3. Siswa dapat mengatur materi kursus berdasarkan kebutuhan dan harapan mereka sendiri, mereka bisa memantau proses belajar dan dapat lebih memahami subjek pembelajaran tersebut.

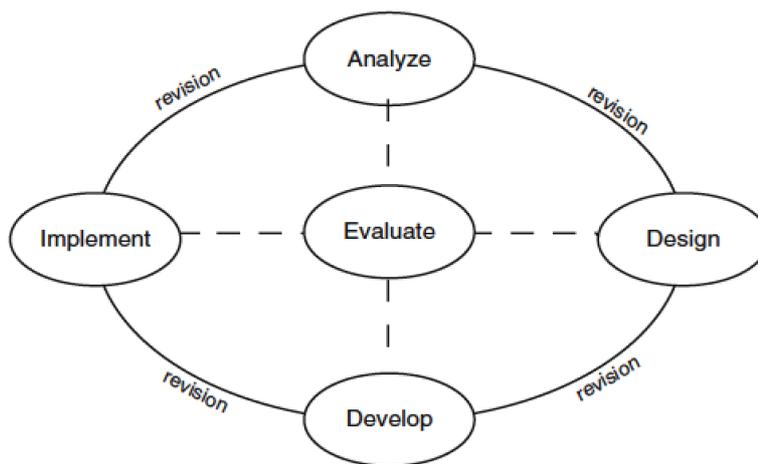
Dan manfaat *e-learning* menurut Smaratungga (2009) adalah sebagai berikut.

1. Meningkatkan kadar interaksi pembelajaran antara peserta didik dengan guru atau instruktur (*enhance interactivity*).
2. Memungkinkan terjadinya interaksi pembelajaran dari mana dan kapan saja (*time and place flexibility*).
3. Menjangkau peserta didik dalam cakupan yang luas (*potential to reach a global audience*).
4. Mempermudah penyempurnaan dan penyimpanan materi pembelajaran (*easy updating of content as well as archivable capabilities*).

2.2.2 ADDIE Framework

ADDIE (*Analysis, Design, Development, Implementation, Evaluation*) adalah *framework* yang dikembangkan sekitar tahun 1990-an oleh Reiser dan Mollenda. ADDIE merupakan salah satu model *Instructional Design* yang terbaik dalam mengembangkan sistem pembelajaran sistematis. Dalam membangun *e-learning*, ADDIE saat ini masih menjadi salah satu model yang efektif karena ADDIE hanyalah sebuah proses yang berfungsi sebagai *framework* untuk situasi yang kompleks dan sangat tepat dalam mengembangkan alat ajar pendidikan (Branch, 2009:2).

ADDIE memiliki kelebihan dibandingkan dengan model pengembangan sistem yang lain. ADDIE memandang *e-learning* lebih holistik dan komprehensif. ADDIE tidak hanya melihat *e-learning* secara teknis, tetapi ADDIE juga melihat *e-learning* dari sisi konten dan karakteristik siswa dan pengajar. Hal tersebut menjadikan ADDIE sangat tepat digunakan dalam membangun *e-learning*.



Gambar 2.1 ADDIE Model

Sumber (Branch, 2009:2)

Gambar 2.1 di atas menjelaskan tentang model ADDIE. Berikut adalah tahapan-tahapan yang ada pada model ADDIE.

1. *Analysis*

Tahap pertama model ADDIE adalah *analysis*. Tahap ini ada tahap yang penting dan krusial. Tujuan dari *analysis* adalah mengidentifikasi kemungkinan-kemungkinan penyebab *performance gaps*. Secara garis besar berikut analisis yang diperlukan dalam tahap *Analysis*:

- a. *Gap Analysis* merupakan analisis yang harus dilakukan dan dimengerti oleh *instructional designer* dalam mengidentifikasi adanya perbedaan kinerja antara kenyataan dan keinginan. Atau dengan kata lain mengidentifikasi *performance gap*. Gagne et al (2005:23) mendefinisikan gap sebagai kebutuhan akibat ketidaksesuaian antara yang diharapkan (*desired*) dan keadaan *existing* (*Actual*) yang terjadi di lapangan. Tahap ini lebih menganalisis kebutuhan yang diinginkan (*desired*) atau biasa disebut analisis kebutuhan. Analisis kebutuhan merupakan konsep penting karena analisis ini selain mengidentifikasi tujuan juga mengidentifikasi keadaan saat ini sehingga sistem yang dihasilkan lebih optimal. Analisis kebutuhan dalam mengembangkan *e-learning* dapat dicapai dengan cara:

1. Mengukur kinerja saat ini (*measure the Actual Performance*)

2. Mengidentifikasi dan mengkonfirmasi kinerja yang diharapkan (*confirm the Desired Performance*).
 3. Mendeskripsikan perbedaan yang terjadi antara kinerja saat ini dan kinerja yang diharapkan serta penyebabnya (*identify the cause for the performance gap*).
- b. Analisis karakteristik pembelajaran adalah analisis yang mengidentifikasi kebutuhan pembelajaran seperti apa yang dibutuhkan dalam proses pembelajaran, keterkaitan antara mata pelajaran, dan *learning outcomes*.
 - c. Analisis kebutuhan sumber daya merupakan analisis yang penekanannya terhadap kondisi lingkungan dan batasan kondisi tersebut. Hasil dari analisis ini adalah sumber daya yang dibutuhkan dalam suatu pembelajaran dan peralatan yang diperlukan untuk mendukung proses pembelajaran.

2. *Design*

Tujuan dalam tahap *design* adalah memverifikasi kinerja yang diharapkan dengan metode pengesanan yang cocok. Elemen *design* dalam konsep *Instructional System Design* akan menghasilkan rancangan atau *blueprint* yang menjadi landasan dan panduan pada proses pengembangan *e-learning*. Dalam tahap ini dilakukan perancangan berdasarkan kebutuhan. Hasil dari *design* ini adalah spesifikasi yang diperlukan *developer* dalam membangun *instructional support material*. *Prototype* yang dihasilkan adalah peluang awal dalam mendapatkan *feedback* dari pengguna dan menekankan aspek fungsionalitas, *feasibility*, dan keberadaan dari sebuah instruksi. Berikut beberapa langkah untuk menghasilkan desain yang optimal:

- a. Mendetailkan tujuan jangka panjang sebuah pembelajaran ke dalam *objectives* pembelajaran tersebut. Langkah ini bertujuan untuk menghasilkan *learning outcomes* yang tepat. *Learning outcomes* terbagi menjadi *intellectual skill*, *cognitive (learning) strategies*, *attitudes*, dan/atau *motor skill*.
- b. Menjelaskan tentang unit-unit utama, *outcomes* yang utama dalam setiap unitnya dan waktu yang harus dialokasikan dalam setiap unit pembelajaran.

- c. Mendekomposisi unit objektif dengan menspesifikan *learning outcomes* dalam setiap unit.
- d. Tahap berikutnya adalah memecah unit-unit yang ada dalam pelajaran dan juga aktifitasnya.
- e. Mendefinisikan spesifikasi pelajaran dan aktifitasnya. Tahap ini menekankan kepada pengembangan *external events* yang akan membawa kondisi pembelajaran yang lebih efektif.
- f. Mendesain kriteria dan spesifikasi penilaian siswa atas pencapaian yang diperoleh. Penilaian yang dibentuk merupakan sebuah pengukuran nilai yang valid dan *reliable*.

3. *Development*

Dalam *Instructional System Design (ISD)* proses pengembangan mengacu pada persiapan bahan-bahan yang akan digunakan dalam kegiatan belajar mengajar. Pengembangan yang dilakukan mengikuti *blueprint* dari proses design sebelumnya. Secara umum, situasi pengembangan terbagi menjadi empat kategori, yaitu:

- a. Bekerja dengan menggunakan kurikulum yang sudah ada (menambahkan dari material yang sudah ada).

Kondisi seperti ini terjadi ketika guru bekerja dengan menggunakan kurikulum, *textbook*, dan bahan referensi yang sudah diadopsi oleh sistem sekolah. Dalam *case* ini, guru akan mengembangkan rencana pembelajaran dengan menggabungkan gaya mengajar dengan muatan lokal sebagai tambahan kurikulum. Jadi, penggunaan bahan yang asli dibuat seminimal mungkin walaupun ada beberapa hal yang penting untuk kebutuhan siswa.

- b. Meredefinisi tujuan dari material yang sudah ada.

Kondisi ini terjadi dalam pengembangan pendidikan karyawan dan juga bisa terjadi di sekolah. Tujuan dari *instructional system design* dalam kondisi ini adalah melakukan analisis dan mendefinisikan kebutuhan serta tujuan dari organisasi, kemudian mengelaborasi bahan yang ada untuk menentukan modifikasi dan suplemen seperti apa yang akan muncul untuk memenuhi kebutuhan mereka.

- c. Menambahkan materi yang lama ke sebuah pembelajaran yang baru.
Kondisi ini terjadi ketika dalam pengembangan sistem yang baru, tidak memungkinkan seluruhnya berasal dari materi yang baru karena terkait waktu dan biaya. Keadaan ini terjadi di sekolah ketika ada pengembangan pembelajaran yang inovatif dan bukan bagian dari kurikulum yang lama. Sebagai solusi pembelajaran bisa diambil dan kombinasi dari matematika, bahasa, atau *science*.
- d. Membuat materi pembelajaran yang baru
Situasi ini ketika terjadi pembuatan materi untuk penelitian atau workshop, dimana instructional designer akan bekerja dengan keseluruhan proses ISD dan akan mengembangkan sebuah pembelajaran atau module yang baru.

Dalam level pengembangan ISD ada beberapa prinsip yang berkaitan dengan konteks di atas:

- a. *Well-established objectives*: Konten dan material yang baik berawal dari tujuan yang didefinisikan di awal. Dengan keberadaan material di pasar yang semakin besar, sehingga modul-modul bisa ditambahkan dan tujuan yang spesifik bisa diraih dengan tambahan modul tersebut. Dalam membangun bahan pembelajaran, tentu membutuhkan biaya, maka dengan tujuan yang didefinisikan dengan tepat akan mengurangi biaya.
- b. *Innovative objectives*: Jika inovasi tinggi sebagai tujuan, semakin besar material yang harus dikembangkan karena bahan yang diinginkan tidak ditemukan di pasar. Kondisi ini menuntut sumber daya finansial dan waktu yang mencukupi.
- c. *Team approach*: Pendekatan *team approach* menuntut peran *instructional designer* yang baik dalam sebuah *project*.
- d. *Instructional design versus media production*: *Instructional system design* harus mendominasi media produksi, karena teknologi memungkinkan pembuatan media yang sangat baik tetapi unsur utama *instructional* sangat terbatas.
- e. *Make or buy*: Selalu mengukur tentang membuat atau membeli sebelum mengambil keputusan dalam pengembangan keseluruhan material.

4. *Implementation*

Dalam model ADDIE, tahap implementasi terbagi menjadi dua tipe. Tipe pertama implementasi dilakukan di antara tahap pembuatan dan evaluasi. Tipe ini disebut *pilot testing* atau *field testing*. Tipe yang kedua, implementasi dilakukan ketika sistem benar-benar selesai pengembangannya.

Berikut lima prinsip yang dapat diaplikasikan dalam proses perencanaan implementasi:

- a. Mengembangkan *Learning Management System* (LMS) yang mencukupi persyaratan dari institusi.
Kompleksitas yang dibangun mengikuti persyaratan dan keinginan institusi. Dari yang hanya menyediakan buku penilaian siswa sampai mencakup catatan individual, kompetensi yang dibutuhkan, periode pengumpulan tugas, dan lain-lain.
- b. Penyediaan *student guidance* dan *support*
Menyediakan informasi yang jelas kepada siswa tentang materi yang dipelajari dan apa yang diharapkan dari pengajar. Bahan yang dikembangkan mengadaptasi time management dari pembelajaran, penilaian, tugas, dan aktifitas yang harus dilakukan.
- c. Perencanaan untuk perubahan
Mencakup cara melatih instruktur dan memberikan dukungan yang dibutuhkan untuk menjadi fasilitator yang efektif dan efisien.
- d. Perencanaan lingkungan pengantara (*staging*)
Instructional designer harus mempertimbangkan dan menyediakan aspek lingkungan pengantara yang memberikan pengaruh terhadap implementasi pembelajaran, termasuk kebutuhan teknologi, dukungan lokal dalam pendistribusian pengajaran, jadwal setiap kelas, dan ketersediaan pengajar.
- e. Perencanaan untuk *maintenance* sistem
Instructional designer harus merencanakan untuk maintain pembelajaran. Termasuk di dalamnya jadwal untuk mengevaluasi, penumpukan dan penggunaan hasil evaluasi, memonitor aktualitas konten pembelajaran,

dan memeriksa hubungan antar komponen dengan kurikulum dan relevansinya.

5. *Evaluation*

Merupakan tahap terakhir dalam proses ISD. Evaluasi bisa terjadi dalam semua tahap proses ISD, termasuk setelah fase pengembangan dan implementasi. Tipe evaluasi yang diterapkan sebagai berikut:

- a. *Materials evaluation*: Evaluasi yang dilakukan terhadap produk *instructional* ketikas sedang dalam pembuatan dan validasi. Evaluasi ini berhubungan dengan pengajar, dimana *instructional designer* menilai kemampuan pengajar dalam menguasai bahan yang ada dan presentasinya.
- b. *Proses evaluation*: Evaluasi dijalankan terhadap proses *instructional system design* itu sendiri. Evaluasi ini bertujuan menjaga agar fase dalam ISD dijalankan dengan baik dan mengadopsi konsep *total quality of management*.
- c. *Learner reactions*: Evaluasi yang fokus terhadap pelajar di akhir periode pengajaran dan evaluasi *outcomes* yang didapatkan oleh pelajar. Evaluasi berupa survey tentang kejelasan dan kualitas materi serta efektivitas pengajar. Selain itu menggali juga kenyamanan lingkungan yang ada.
- d. *Learner achievement*: Evaluasi merupakan evaluasi lanjutan dari evaluasi *outcomes*. Evaluasi ini penekanannya terhadap *design* dan *development*, yang digunakan untuk mengetahui sejauh mana pelajar mencapai tujuan pengajaran dan untuk mengetahui umpan balik tentang efektivitas dari pembelajaran. Termasuk menggambarkan apakah pelajar lulus atau tidak.
- e. *Instructional consequences*: Evaluasi ini mengandung tiga tahap review, yaitu transfer pelatihan ke lingkungan aplikasi yang lain, mengukur adanya keuntungan dari pembelajaran, dan seberapa optimal dan mudah aplikasi tersebut dipelajari.

Setiap tipe evaluasi di atas terbagi menjadi tipe *formative* dan *summative*. Evaluasi *formative* adalah evaluasi yang terjadi saat sistem dalam proses pengembangan. Evaluasi *summative* merupakan evaluasi yang terjadi setelah fase implementasi melalui *feedback* dari siswa dan juga pengajar.

2.2.3 Object-Oriented Analysis and Design (OOAD)

Menurut Satzinger *et al* (2010, p59) pendekatan berorientasi objek yaitu memandang sistem informasi sebagai kumpulan objek yang berinteraksi yang bekerja bersama untuk menyelesaikan tugas. Karena pendekatan berorientasi objek memandang sistem informasi sebagai kumpulan objek yang berinteraksi, maka:

1. Analisis berorientasi objek (OOA) mendefinisikan semua jenis objek yang melakukan pekerjaan dalam sistem dan menunjukkan interaksi pengguna apa, yang disebut kasus penggunaan, yang diperlukan untuk menyelesaikan tugas.
2. Desain berorientasi objek (OOD) mendefinisikan semua jenis tambahan objek yang diperlukan untuk berkomunikasi dengan orang dan perangkat dalam sistem, menunjukkan bagaimana objek berinteraksi untuk menyelesaikan tugas, dan memurnikan definisi setiap jenis objek sehingga dapat diimplementasikan dengan bahasa atau lingkungan tertentu.
3. Pemrograman berorientasi objek (OOP) terdiri dari penulisan pernyataan dalam bahasa pemrograman untuk mendefinisikan apa yang setiap jenis objek lakukan.

Sedangkan menurut Whitten *et al* (2007, p25), *Object-Oriented Analysis and Design* (OOAD) merupakan suatu kumpulan alat dan teknik untuk mengembangkan suatu sistem yang akan menggunakan teknologi objek untuk membangun sebuah sistem dan piranti lunak.

2.2.4 Pengertian UML

Menurut Whitten dan Bentley (2009, p371), *Unified Modelling Language* (UML) merupakan sekumpulan permodelan yang digunakan untuk menggambarkan sistem perangkat lunak dalam hal objek.

2.2.5 Class Diagram

Menurut Satzinger *et al* (2010, p187), *Domain Model Class Diagram* adalah sebuah UML *class diagram* yang menggambarkan cara kerja *problem domain classes*, *association* dan *atributes*.

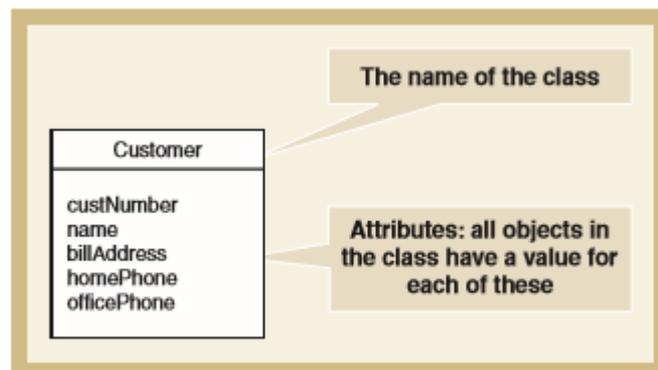
Berikut komponen *domain class diagram*:

Attributes : Karakteristik seperti objek yang memiliki nilai seperti warna, ukuran, bentuk, dan lain-lain.

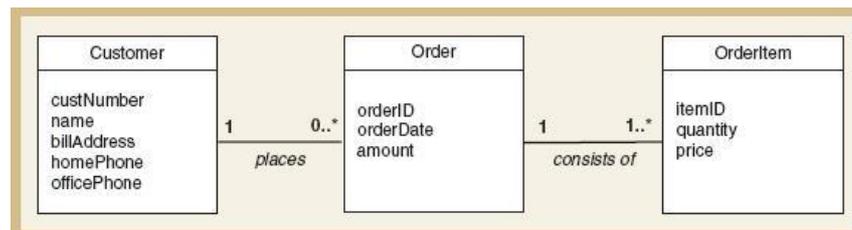
Class : Tipe atau klasifikasi dari objek yang sama.

Methods : Apa yang dapat dilakukan oleh sebuah objek.

Message : Komunikasi objek yang berhubungan.



Gambar 2.2 Simbol Domain Class Diagram



Gambar 2.3 Contoh Sederhana Domain Class Diagram

2.2.6 Activity Diagram

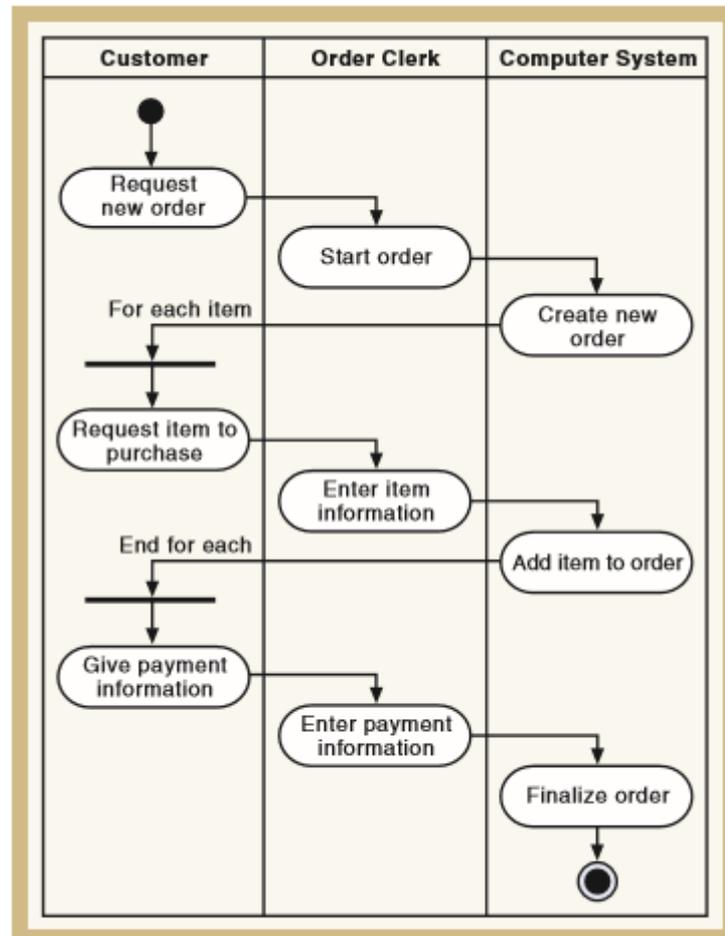
Menurut Satzinger *et al* (2010, p141), *Activity Diagram* merupakan salah satu tipe dari *workflow digram* yang menjelaskan aktifitas pengguna dan urutan alurnya.

Berikut komponen *activity diagram*:

Activity : Elemen yang menggambarkan kegiatan.

Action : Elemen yang menggambarkan aksi dari suatu kegiatan.

- Initial State* : Elemen yang memperlihatkan dimana aliran aktivitas berawal.
- Final State* : Elemen yang memperlihatkan dimana aliran aktivitas berakhir.
- Decision* : Elemen yang menggambarkan suatu kondisi.



Gambar 2.4 Contoh Activity Diagram

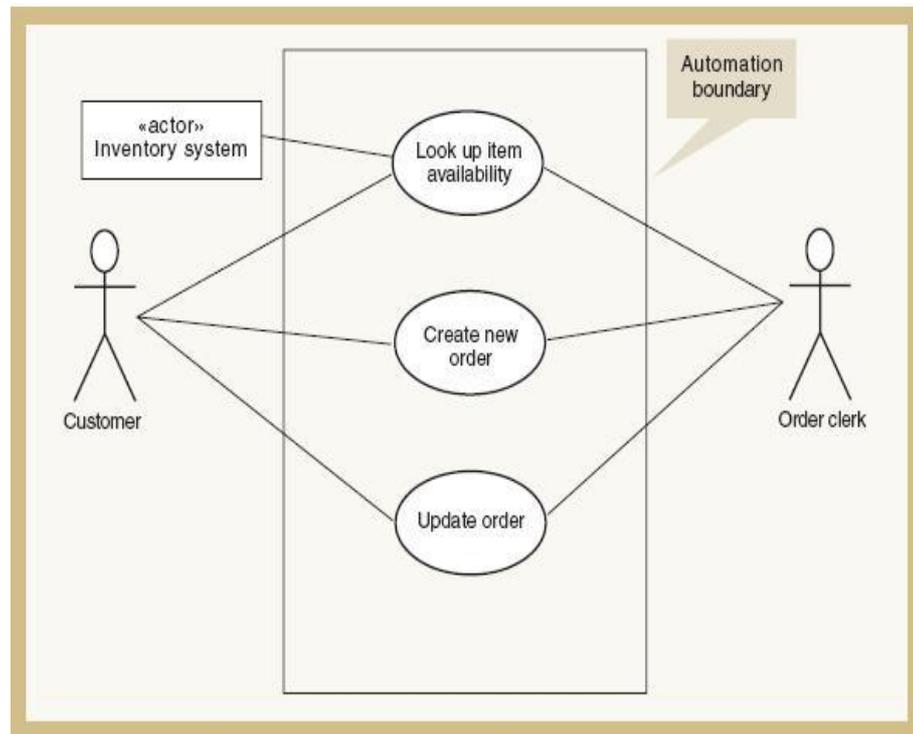
2.2.7 Use Case Diagram

Menurut Satzinger *et al* (2010, p242), *Use Case Diagram* merupakan diagram yang menggambarkan keberagaman peran pengguna dan bagaimana peranan tersebut menggunakan sistem. *Use Case Diagram* dapat diartikan juga sebagai diagram yang menunjukkan urutan pesan antara aktor eksternal dan sistem selama *use case* tersebut berlangsung.

Berikut komponen *Use Case diagram*:

- Actor* : Menggambarkan aktor pada diagram kelas.
- Case* : Menggambarkan proses pada diagram use case.
- Association* : Menggambarkan hubungan asosiasi.

- Extends* : Menggambarkan kebergantungan antar item.
Boundaries : Sebagai pembatas antara aktor dengan sistem.



Gambar 2.5 Contoh Use Case Diagram

2.2.8 Use Case Description

Menurut Satzinger *et al* (2010, p221), *Use Case Description* merupakan metode yang paling formal untuk mendokumentasikan *use case*. Ini adalah metode yang tepat untuk mendeskripsikan aliran internal dan aktivitas dari *use case* yang ada, meskipun memerlukan waktu yang lebih banyak untuk menjelaskan. Menurut Satzinger, *use case description* dibagi menjadi tiga tingkat *detail* :

1. Brief Description

Brief Description dapat digunakan untuk *use cases* yang sangat sederhana, terutama ketika sistem yang akan dikembangkan kecil, aplikasi dipahami dengan baik.

Create new order description

When the customer calls to order, the order clerk and system verify customer information, create a new order, add items to the order, verify payment, create the order transaction, and finalize the order.

Gambar 2.6 Contoh *Brief Description*

Sumber: (Satzinger, Jackson, & Burd, 2010, p172)

2. *Intermediate Description*

Intermediate-level use case description memperluas *brief description* untuk menyertakan aliran internal kegiatan untuk *use case*.

Flow of activities for scenario of <i>Order Clerk creates telephone order</i>
<p>Main Flow:</p> <ol style="list-style-type: none"> 1. Customer calls RMO and gets order clerk. 2. Order clerk verifies customer information. If a new customer, invoke <i>Maintain customer account information</i> use case to add a new customer. 3. Clerk initiates the creation of a new order. 4. Customer requests an item be added to the order. 5. Clerk verifies the item and adds it to the order. 6. Repeat steps 4 and 5 until all items are added to the order. 7. Customer indicates end of order; clerk enters end of order; system computes totals. 8. Customer submits payment; clerk enters amount; system verifies payment. 9. System finalizes order.
<p>Exception Conditions:</p> <ol style="list-style-type: none"> 1. If an item is not in stock, then customer can <ol style="list-style-type: none"> a. choose not to purchase item, or b. request item be added as a back-ordered item. 2. If customer payment is rejected due to bad-credit verification, then <ol style="list-style-type: none"> a. order is canceled, or b. order is put on hold until check is received.

Gambar 2.7 Contoh *Intermediate Description*

Sumber: (Satzinger, Jackson, & Burd, 2010, p172)

3. *Fully Developed Description*

Fully Developed Description adalah metode yang paling formal mendokumentasikan sebuah use case. Meskipun memerlukan waktu lebih untuk mengerjakan, jenis dari use case description ini dapat meningkatkan kemungkinan akan pemahaman mengenai proses bisnis.

Use Case Name:	Create new order	
Scenario:	Create new telephone order	
Triggering Event:	Customer telephones RMO to purchase items from the catalog.	
Brief Description:	When customer calls to order, the order clerk and system verify customer information, create a new order, add items to the order, verify payment, create the order transaction, and finalize the order.	
Actors:	Telephone sales clerk.	
Related Use Cases:	Includes: <i>Check item availability</i> .	
Stakeholders:	Sales department: to provide primary definition. Shipping department: to verify information content is adequate for fulfillment. Marketing department: to collect customer statistics for studies of buying patterns.	
Preconditions:	Customer must exist. Catalog, Products, and Inventory items must exist for requested items.	
Postconditions:	Order and order line items must be created. Order transaction must be created for the order payment. Inventory items must have the quantity on hand updated. The order must be related (associated) to a customer.	
Flow of Activities:	Actor	System
	1. Sales clerk answers telephone and connects to a customer. 2. Clerk verifies customer information. 3. Clerk initiates the creation of a new order. 4. Customer requests an item be added to the order. 5. Clerk verifies the item (<i>Check item availability use case</i>). 6. Clerk adds item to the order. 7. Repeat steps 4, 5, and 6 until all items are added to the order. 8. Customer indicates end of order; clerk enters end of order. 9. Customer submits payment; clerk enters amount.	2.1 Display customer information. 3.1 Create a new order. 5.1 Display item information. 6.1 Add an order item. 8.1 Complete order. 8.2 Compute totals. 9.1 Verify payment. 9.2 Create order transaction. 9.3 Finalize order.
Exception Conditions:	2.1 If customer does not exist, then the clerk pauses this use case and invokes <i>Maintain customer information use case</i> . 2.2 If customer has a credit hold, then clerk transfers the customer to a customer service representative. 4.1 If an item is not in stock, then customer can a. choose not to purchase item, or b. request item be added as a back-ordered item. 9.1 If customer payment is rejected due to bad-credit verification, then a. order is canceled, or b. order is put on hold until check is received.	

Gambar 2.8 Contoh *Fully Developed Description*

2.2.9 *Persistent Object*

Menurut Satzinger *et al* (2010), *Persistent Object* adalah isi *database* yang *valid* dengan *primary key* sebagai identifier, dengan kata lain *persistent object* adalah *row-row* yang ada pada *table database*. *Object* ini akan tetap ada tanpa bergantung dari program atau proses. Dalam *design* yang mengikuti *three layer architecture*, *problem domain* setiap *class* biasanya *persistent*. Menyimpan objek ke dalam *memory* yang *persistent* (seperti magnetik atau optical disk) memastikan objek tetap ada. Objek ini dapat disimpan secara *persistent* dalam *file* atau *database management system*.

2.2.10 Sequence Diagram

Menurut Satzinger *et al* (2010, p435), *Sequence Diagram* merupakan diagram yang mendeskripsikan aliran informasi dalam sistem. *Sequence Diagram* memiliki tujuan utama untuk mengidentifikasi kolaborasi kelas dan apakah kelas tersebut harus mengirim pesan antara satu sama lain.

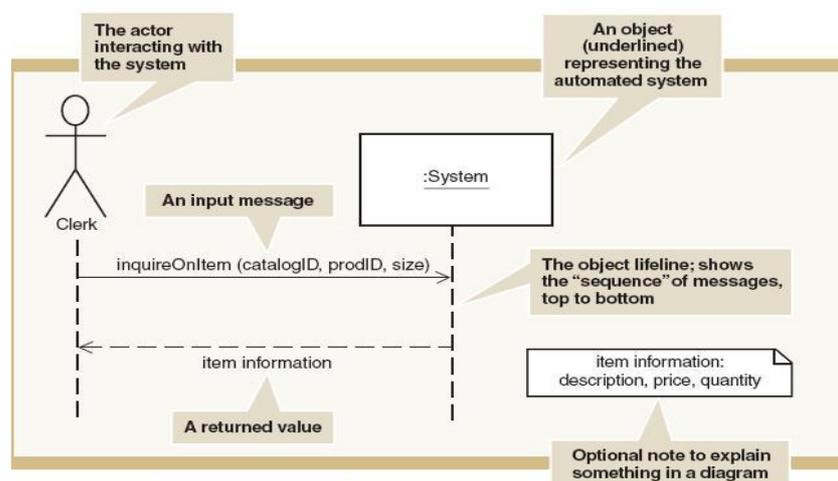
Berikut komponen *Sequence diagram*:

Boundary : Menambahkan kelas batasan (boundary) pada diagram.

Object : Menambahkan objek baru pada diagram.

Object message : Menggambarkan pesan antar dua objek.

Return message : Menggambarkan pengembalian dari pemanggilan prosedur



Gambar 2.9 Contoh Sequence Diagram

2.2.11 User Interface

Menurut Satzinger *et al* (2010, p32), *User Interface* merupakan sistem itu sendiri yang berhubungan dengan end *user* saat sedang menggunakan sistem seperti fisik, perseptual, dan konseptual.

2.2.12 Bahasa Pemrograman PHP

PHP (*Hypertext Preprocessor*) adalah bahasa komputer yang dibuat untuk pengembangan web dinamis. Pada umumnya PHP digunakan di server namun juga dapat berdiri sendiri sebagai aplikasi *graphical* (www.php.net). Pada analisis sistem rekomendasi metode *hybrid* untuk daerah wisata ini

digunakan bahasa pemrograman PHP basis pengetahuan. Penggunaan PHP dipilih karena PHP dan MySQL memiliki beberapa kelebihan seperti dinyatakan oleh Nugroho, B (2008) kelebihannya sebagai berikut:

- a. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- b. *Web server* yang mendukung PHP dapat ditemukan dimana – mana dari mulai IIS sampai dengan *Apache*, dengan konfigurasi yang relatif mudah.
- c. Dalam sisi pengembangan lebih mudah, karena banyaknya milis – milis dan *developer* yang siap membantu dalam pengembangan.
- d. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena referensi yang banyak.

PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (Linux, Unix, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

2.2.13 MySQL

MySQL merupakan *software* sistem manajemen *database* (*Database Management System* – DBMS) yang sangat populer di kalangan pemrogram web, terutama di lingkungan *Linux* dengan menggunakan skrip dan *Ped*.

Fungsi MySQL dapat dikatakan sebagai *interpreter query*, karena setiap kita menggunakan query SQL (perintah SQL) kita harus meletakkannya di dalam fungsi ini. Dengan kata lain, SQL tidak dapat dijadikan tanpa adanya fungsi MySQL. MySQL termasuk jenis *Relational Database Management System* (RDBMS). Sehingga istilah seperti tabel, baris dan kolom tetap digunakan dalam MySQL. Pada MySQL sebuah *database* mengandung beberapa tabel, tabel terdiri dari sejumlah baris dan kolom.

SQL merupakan kependekan *Structured Query Language*. SQL digunakan untuk berkomunikasi dengan sebuah *database*. SQL adalah bahasa yang meliputi perintah-perintah untuk menyimpan, menerima, memelihara, dan mengatur akses-akses ke basis data serta digunakan untuk memanipulasi dan menampilkan data dari *database*. (R.W. Rosari, 2008).

2.2.14 *Javascript*

Javascript merupakan *scripting language* yang terintegrasi dengan *web browser* untuk memberikan fleksibilitas tambahan bagi *programmer* untuk mengontrol elemen-elemen dalam halaman *web*. Aplikasi *javascript* sebenarnya cukup luas namun pada aplikasi yang dirancang, digunakan *javascript* yang dibatasi untuk DHTML (*Dinamic HyperText Markup Language*), yaitu *javascript* yang digunakan untuk mengakses *property*, *method*, dan *event handler* yang disediakan oleh DOM dan CSS.

Property didefinisikan sebagai *setting* nilai suatu objek tertentu. Contohnya adalah warna suatu teks, aksi untuk suatu *form*, nama *file* untuk suatu gambar dan *file* lainnya. *Method* adalah fungsi-fungsi yang dapat diterapkan dalam suatu objek. Misalnya *maximize* untuk *window*. *Event handler* menspesifikasikan bagaimana suatu objek merespon terhadap suatu kejadian, misalnya suatu *button* di klik dan *window* di *resize*. (Oktavian, Diar Puji. 2010).

2.2.15 *Aplikasi Framework*

Aplikasi *framework* adalah suatu *framework* yang digunakan sebagai struktur standar implementasi dari aplikasi untuk suatu *operating* sistem tertentu. Aplikasi *framework* menjadi populer seiring berkembangnya *Graphical User Interface* (GUI). Membuat GUI secara otomatis menjadi lebih mudah menggunakan standar *framework*. Teknik *Object Oriented Programming* (OOP) selalu digunakan saat implementasi *framework*, misalnya suatu *unique parts* dari aplikasi dapat dengan mudah diturunkan dari *Class-Class* yang telah ada di *framework*. Menurut kamus Inggris-Indonesia yang disusun oleh John M. Echols dan Hassan Sadily, *framework* memiliki arti kerangka. Sehingga arti *framework* untuk dunia IT adalah arsitektur (kerangka kerja) umum dari suatu aplikasi. *Framework* menyediakan semacam panduan mengenai apa saja yang harus dikerjakan dalam mewujudkan suatu fungsionalitas. *Framework* merupakan struktur konseptual dasar, digunakan untuk memecahkan permasalahan yang kompleks terutama dalam penanganan dan kerumitan dalam membangun sebuah sistem untuk *software* atau perangkat lunak.

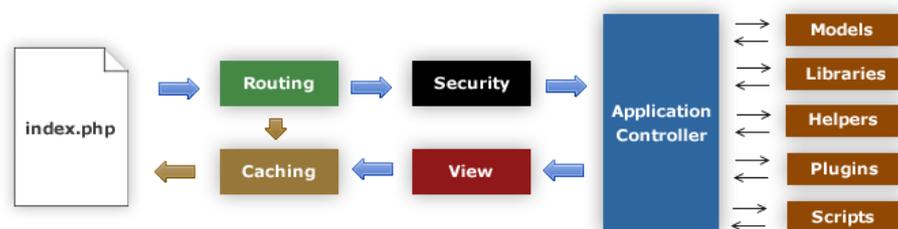
2.2.16 CodeIgniter

CodeIgniter adalah aplikasi *open source* untuk membangun *website* dinamis menggunakan PHP. *CodeIgniter* akan membantu *developer* untuk membuat aplikasi *web* dengan cepat dan mudah. Dengan *CodeIgniter*, jumlah kode yang diketikkan akan menjadi lebih sedikit dan hal ini akan mempermudah untuk membaca kode (Upton, 2007).

CodeIgniter membantu dalam mengikuti konsep MVC (*Model, View, Controller*). MVC adalah konsep pemrograman yang memisahkan pemrograman logik aplikasi dengan *interface*. Singkatnya, memungkinkan halaman *web* berisi sedikit sekali skrip PHP karena *file interface* terpisah dengan *file* skrip PHP.

1. *Model* mewakili struktur data. Umumnya kelas *model* akan berisi fungsi – fungsi yang akan membantu untuk mengambil, *insert*, dan *update* data di *database*.
2. *View* adalah informasi yang disajikan keuser. Sebuah *view* normalnya adalah sebuah halaman *web*, namun di CI sebuah *view* dapat juga menjadi bagian halaman seperti *header* atau *footer*. Dapat juga menjadi berbagai jenis “Halaman” yang lain.
3. *Controller* berfungsi sebagai penengah antara *Model, View* dan *resource* lain yang dibutuhkan untuk memproses HTTP *request* dan memproduksi sebuah halaman *web*.

Alur program yang berjalan pada aplikasi yang menggunakan konsep MVC (*Model View Controller*) menggunakan *framework CodeIgniter* yang tergambaran pada Gambar 2.10.



Gambar 2.10 Alur program konsep MVC pada *Framework Codeigniter*

1. *index.php* merupakan *controller* utama dari aplikasi ini yang berfungsi menginisialisasi *basic resource* yang dibutuhkan untuk menjalankan *framework CodeIgniter*.
2. Router menganalisa *HTTP request* untuk menentukan apa yang harus dilakukan dengan *HTTP request* itu.
3. Jika file *cache* masih ada, maka akan dikirim langsung ke *browser*, tanpa melewati eksekusi normal sistem.
4. Keamanan (*Security*), sebelum *controller* aplikasi di panggil, *HTTP request* dan data yang dikirim *user*, difilter untuk alasan keamanan.
5. *Controller* memanggil *model*, librari inti, *plugin*, *helper*, dan *resource* lainnya yang dibutuhkan untuk memproses *request* tertentu.

View yang sudah diproses, dikirim ke *browser* sebagai hasil yang terlihat. Jika status *caching* ON, *view* akan disimpan di *cache*, jadi jika ada *request* yang sama, *view* itu dapat ditampilkan lagi.

2.2.17 Web Application

Aplikasi *web* adalah sebuah program yang bila dieksekusi akan menghasilkan sebuah aplikasi yang dapat bekerja sesuai dengan yang diinginkan. Aplikasi *web* dibangun dengan menggunakan bahasa HTML (*Hypertext Markup Language*). Pada masa kini aplikasi web dikembangkan untuk memperluas kemampuan HTML dengan PHP dan ASP pada skrip objek. Aplikasi *web* dapat dibagi menjadi dua bagian yaitu aplikasi *web* dinamis dan aplikasi *web* statis. Aplikasi *web* merupakan sebuah aplikasi yang menggunakan teknologi *browser* untuk menjalankan aplikasi dan diakses melalui jaringan komputer (Remick, 2011).

Arsitektur aplikasi *web* meliputi klien, *web server*, *middleware* dan basis data. Klien berinteraksi dengan *web server*. Secara internal, *web server* berkomunikasi dengan *middleware* dan *middleware* yang berkomunikasi dengan basis data. Contoh *middleware* adalah PHP dan ASP. Pada mekanisme aplikasi *web* dinamis, terjadi tambahan proses yaitu server menerjemahkan kode PHP menjadi kode HTML. Kode PHP yang diterjemahkan oleh mesin PHP yang akan diterima oleh klien (Abdul Kadir, 2009).

Aplikasi adalah sebuah “mahakarya Symphony Orchestra” dari pelaku Teknologi Informasi, yang merupakan hasil kerjasama antara sumber daya manusia, tools dan pengguna dalam sebuah manajemen yang terintegrasi dimana didalamnya ada perencanaan, ujicoba, pelaksanaan dan pemeliharaan, dengan tujuan akhir untuk mendukung aktifitas manusia agar lebih efisien dan efektif (Muhammad Safri Lubis, 2011).

Jadi aplikasi adalah sebuah program hasil karya yang siap pakai. Program yang terbuat dengan beberapa tahapan yang melaksanakan suatu fungsi yang telah diperintahkan.

2.2.18 Notepad ++

Notepad++ merupakan *software open source* atau gratis yang dikembangkan oleh notepad-plus-plus.org. *Software* ini biasa digunakan sebagai *source code editor* dan bisa juga digunakan untuk pengganti notepad bawaan windows. Notepad++ mendukung beberapa bahasa. Dapat digunakan dilingkungan Microsoft windows. Penggunaanya diatur oleh GPL License. Notepad++ ditulis berdasarkan komponen atau aturan Scintilla (*Aturan Source Code Editing*), Notepad++ ditulis menggunakan bahasa C++ dan murni menggunakan win32 API dan STL.

2.2.19 XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL *database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl.