BAB II

LANDASAN TEORI

Bab ini bertujuan untuk memberikan latar belakang teori akan teknologi *Content Managements System*. Dengan ini diharapkan pembaca dapat lebih memahami perkembangan proses pembangunan yang dijabarkan dalam penuisan tugas akhir ini.

2.1. Pengertian *E-learning*

E-learning terdiri dari dua bagian, yaitu „e‟ yang merupakan singkatan dari *electronic* dan *learning* yang berarti pembelajaran, jadi *e-learning* berarti pembelajaran dengan menggunakan jasa bantuan perangkat elektronika, khususnya perangkat komputer. *E-learning* atau pembelajaran melalui *online* dengan demikian merupakan pembelajaran yang pelaksanaannya didukung oleh jasa teknologi seperti telepon, audio, videotape, transmisi satelit atau komputer.

Dalam perkembangannya, komputer dipakai sebagai alat bantu pembelajaran, karena itu dikenal dengan istilah *Computer Based Learning* (CBL) atau *Computer Assisted Learning* (CAL). Saat pertama kali komputer mulai diperkenalkan khususnya untuk pembelajaran, maka komputer menjadi popular dikalangan anak didik. Hal ini dapat dimengerti karena berbagai variasi teknik mengajar bisa dibuat

dengan bantuan komputer tersebut, maka setelah itu teknologi pembelajaran terus berkembang dan dikelompokkan menjadi dua yaitu :

1. *Technology-based learning*
2. *Technology-based Web-learning*

Technology based-learning ini pada prinsipnya terdiri dari dua, yaitu *audio information technologies* (*audio tape, radio, voice mail, telepon*) dan *video information technologies* (*video tape, video text, video messaging*), sedangkan *technology based web-learning* pada dasarnya adalah *data information technologies* (*bulletin board, internet, email, tele-collaboration*). Karakteristik e-*learning* antara lain adalah:

1. Memanfaatkan jasa teknologi elektronik; dimana dosen dan mahasiswa, mahasiswa dan sesama mahasiswa atau dosen dan sesama dosen dapat berkomunikasi dengan relatif mudah dengan tanpa dibatasi oleh hal-hal yang protokoler
2. Menggunakan bahan ajar bersifat mandiri (*self learning materials*) yang disimpan di komputer sehinga dapat diakses oleh dosen dan mahasiswa kapan saja dan dimana saja jika yang bersangkutan memerlukannya; dan
3. Memanfaatkan jadwal pembelajaran, kurikulum, dan hasil kemajuan pembelajaran dapat dilihat setiap saat di komputer.

2.2. Siklus *Learning*

Siklus pengembangan *learning* memiliki fase-fase yaitu:

Tahap 1 : *Scope*

Tahap ini meliputi perencanaan dan analisis batasan dari perancangan *learning*, pendefinisian tipe dan tujuan *learning*, menentukan metode *learning* yang digunakan, sifat *learning*, domain *learning*, serta teknologi yang digunakan. Pada umumnya tujuan *learning* ini digunakan untuk menuntun pengajar, bukan sebagai pedoman siswa.

Tahap 2: *Creation*

Pada tahap ini perancangan membuat desain, mengembangkan dan melakukan proses *learning.* Hal ini yang dipertimbangkan pada tahap ini adalah sifat dari materi *learning* serta perancangan proses interaksi yang diharapkan terjadi saat *learning* berlangsung. Ditahap ini perancangan proses interaksi yang diharapkan terjadi saat *learning* berlangsung. Di tahap ini perancang mendesain sasaran *learning.* Teknologi, pengembangan interaksi, *skill* yang dibutuhkan, *tasks* dari pengajar serta *feedback* yang diharapkan.

Tahap 3 : *User Experience*

User experience adalah proses yang penting dalam menentukan penggunaan sumber daya *learning* selama tahap 1 dan 2. Tahap ini akan menjawab apakah desain *learning* berguna dan bernilai, bagaimana reaksi siswa terhadap isi, presentasi dan interaksi *learning*.

Tahap 4 : *Meta-Evaluation*

Meta-evaluation adalah proses mengevaluasi efektifitas proses perancangan *learning.* Eksplorasi terhadap keberhasilan dan kegagalan selama proses *learning* berlangsung dilakukan untuk melakukan perbaikan perancangan dan implementasi *learning* selanjutnya.

Tahap 5 : *Evaluation*

Tahap ini dilakukan untuk mengevaluasi *learning,* dapat dilakukan dengan pemberian ujian atau tugas secara berkelompok. Sifat evaluasi ini dapat berupa *formative* (dilakukan saat proses *learning* berlangsung) atau *summative (*dilakukan setelah seluruh materi disampaikan). Evaluasi juga dilakukan siswa dengan memberikan *feedback* atas proses *learning* yang telah dilakukan baik (dari sisi kualitas sumberdaya *learning*, instruksi yang diberikan, relevansi dan format *learning*) dengan cara langsung pada pengajar, atau melalui *questioner* dari institusi.

2.2.1. Domain Penelitian

Penelitian pada area *collaborative learning* dibagi menjadi tujuh domain yang berbeda yaitu:

1. *Control of collaborative learning*

Kendali dan interaksi kolaborasi mengacu pada cara penyampaian dalam sistem di lingkungan kolaborasi. Sistem *collaborative learning* dapat menjadi bagian yang menganalisis dan mengendalikan kolaborasi atau hanya bertindak sebagai alat pengantar kolaborasi. Sistem *collaborative learning* pada dimensi ini dapat

diklasifikasikan sebagai sistem yang aktif, pasif atau apapun yang ada dalam batasan keduanya.

2. *Tasks of collaborative leaning*

Dalam lingkungan *collaborative learning*, kolaborator dapat memiliki berbagai jenis *tasks*. *Tasks* yang umumnya ditemukan dalam lingkungan *collaborative learning* adalah :

a. *Collaborative concept_learning tasks*

b. *Collaborative problem_solving tasks*

c. *Collaborative design tasks*

3. *Theories of learning and cognition in collaboration*

Dillenbourg mengidentifikasikan tiga teori *learning* yang dapat digunakan dalam sistem *collaborative learning* yaitu : *socio-constructivist, socio-cultural* dan *shared cognition.*

4. *Design of collaborative learning context*

Dasar utama dari kolaborasi adalah *willingness* dari partisipan untuk berpesan dan berkolaborasi dalam *sense* yang konstruktif. Slavin melaporkan hasil studi yang dilakukan oleh Kuhn (1972) yang menemukan bahwa perbedaan yang kecil pada level kemampuan kognitif antara partisipan kolaborasi ternyata lebih mengembangkan pertumbuhan kognitif, dibandingkan dengan partisipan yang memiliki perbedaan yang jauh kemampuan kognitifnya. Hal ini menunjukkan bahwa

partisipan kolaborasi seharusnya mempunyai tingkat pengetahuan yang hampir sama untuk membangun kolaborasi yang konstruktif.

Kombinasi *tasks* dan jumlah kolaborator yang terlibat dalam *learning* ditentukan oleh subjek domain, teori *learning* yang digunakan dan kemampuan sistem. Berikut ini adalah jenis-jenis rancangan lingkungan kolaborasi :

a. Dua atau lebih partisipan berkolaborasi satu sama lain menggunakan computer sebagai media. Sistem tidak melakukan apapun selain menyediakan channel komunikasi untuk kolaborasi tanpa memainkan peran yang aktif.
b. Dua atau lebih partisipan berkolaborasi satu sama lain menggunakan seorang tutor aktif yang mengendalikan dan mengarahkan interaksi kolaborasi.
c. Dua partisipan atau lebih partisipan berkerjasama menangani masalah pada tempat kerja yang sama, menggunakan tutor sedemikian rupa sehingga sama jika partisipan berkerja sendiri.
d. Dua atau lebih partisipan berkerjasama dengan sedikitnya satu partisipan menggunakan simulai sistem. Partisipan dapat menentukan aksinya secara langsung atau atas permintaan partisipan lain.

5. *Roles of the peers*

Pada lingkungan *collaborative learning,* tujuan akan dibagi menjadi *subtasks* yang akan dilakukan oleh tiap partisipan. Hal ini juga menunjukkan bahwa partisipan diberikan peran secara natural dan *applicable* pada domain yang diberikan. (Blave Vive Kumar, 1996) menggambarkan dua peran yaitu eksekutor, orang yang menangani masalah dan reflektor, orang yang meneliti dan mengomentari atas penyelesaian masalah.

Secara umum lingkungan *collaborative learning* dapat memiliki sekumpulan peran sebagai :

a. *Decomposing,* mengacu pada pekerjaan membagi penanganan masalah kedalam *tasks*, tiap *tasks* dibagi ke dalam sejumlah tujuan. Tujuan akan menjadi objek *learning* bagi pembelajarannya.

b. *Defining,* mengusulkan sebuah tujuan dari sebuah *tasks.*

c. *Critiquing,* hipotesa yang diusulkan seorang partisipan berita alternatifnya.

d. *Convicing,* aksi membandingkan sejumlah hipotesa dan mendukung salah satunya.

e. *Reviewing,* pekerjaan yang menjamin agar interaksi kolaborasi mengacu pada proses *learning* yang konstruktif.

f. *Referencing,* pekerjaan yang menyediakan fakta dan material terkait yang diminta oleh partisipan lain.

6. *Domain of collaboration*

Secara umum *collaborative learning* efektif pada domain dimana partisipan berada pada pekerjaan *skill acquisition, join planing, categorization* dan *memory tasks.*

7. *Teaching methodologies*

Sejumlah metodologi pengajaran yang diidentifikasi mendukung collaborative learning adalah :

a. Practice, partisipan diminta untuk menggunakan sebuah tujuan learning pada sebuah masalah spesifik.
b. Learning by teaching, metodologi ini mendukung learning dengan memiliki system sebagai learning tools.
c. Situated learning, mahasiswa menjadi partisipan dalam sebuah latihan sociocultural, kemampuan belajar dan kemampuan social berjalan bersama.
d. Negotiated learning, partisipan dan system bernegosiasi untuk mencapai tujuan belajar.
e. Discovery learning, mahasiswa mengeksplorasi sebuah lingkungan untuk proses learning.

2.2.1. Definisi *Collaborative Learning*

Beberapa definisi *Collaborative learning*

1. *Umbrella term* untuk berbagai jenis pendekatan edukasi yang melibatkan usaha bekerjasama secara intelektual antar mahasiswa atau mahasiswa dan pengajar.

2. *An instruction method in wich students work in groups toward a common academic goal* (suatu metode instruksi dimana para mahasiswa bekerja sama dalam suatu kelompok untuk mencapai tujuan akademik tertentu).

Tujuan utama penggunaan *collaborative learning*

1. Fokus pada belajar yang aktif

Usaha dalam the *Calculus Reform Movement* [5] mengindikasikan bahwa kelas menjadi tempat yang paling efektif saat para mahasiswanya ikut serta atau terlibat dalam materi yang disampaikan. Format pengajar memungkinkan seorang instruktur meringkas sejumlah materi dengan efesien namun hal ini tidak otomatis mengefektifkan proses penyampaian materi pada mahasiswa.

2. Membangun *skill* menulis dan komunikasi lisan

Colaborative learning mendorong mahasiswa untuk berkomunikasi satu sama lain, menyatakan respon pada pertanyaan, bekerja dalam lontaran pendapat yang berbeda-beda dan menuliskan kesimpulan dengan jelas.

3. Memberikan tanggungjawab belajar secara eksplisit

Jika dalam kelas lebih terfokus pada kerja secara kelompok, dan bukan pada instruktur, maka mahasiswa akan menyadari bahwa kelompok mereka tidak dapat mengikuti pelajaran sehari-hari bila tanpa persiapan. Hal ini akan mendorong kegiatan membaca dan penyelesaian pekerjaan lain pada suatu hari untuk disampaikan pada kelompok mereka di kemudian hari.

4. Memperjelas peran pengajar sebagai fasilitator dan mentor

Seorang instruktur akan semakin melepaskan *control* terhadap kelas apabila format kelas menekankan pada aktivitas kelompok, sehingga perannya akan menjadi pemberi tanggapan jika ada pertanyaan, sebagai pelatih kelompok secara individual, mengatasi kesulitan-kesulitan yang umum terjadi dan menyarankan suatu pendekatan baru.

5. Dapat mencakup materi lebih banyak atau lebih baik (untuk materi yang sama)

Dengan semakin aktifnya mahasiswa dalam kelas dan semakin reponsif atas proses belajar maka ditemukan bahwa kelas dapat mengalami percepatan sebanyak 20% , sehingga di semester selanjutnya materi dapat diperbanyak.

6. Membangun rasa percaya diri dan mandiri pada mahasiswa

Sebuah kelas yang melibatkan mahasiswa dan partisipasi kelompok, akan mengurangi ketergantungan mahasiswa pada pengajar dan mahasiswa belajar bagaimana cara belajar. Mahasiswa menjadi cakap saat membaca dan berlatih, dan mereka akan membuat strategi dalam menguasai pengetahuan yang baru.

7. Memiliki pengalaman bekerja secara kelompok

Saat kelompok kerja melakukan perancangan dan membuat sebuah program, anggota kelompok harus menentukan bagaimana struktur solusi yang diberikan, *tasks* apa yang akan ditangani suatu modul dan bagaimana bentuk *interface* dari modul tersebut kemudian *source code* dibuat, kelompok akan mengetahui apa yang berjalan dan yang mana yang tidak. Aktivitas seperti ini akan menekankan prinsip-prinsip *software engineering*, misalnya seperti kebutuhan akan spesifikasi yang baik.

8. Mendukung *peer review*

Saat bekerja pada proyek pemrograman sebagai bagian dari suatu kelompok, secara alami mahasiswa akan melihat sumber kelompok lain untuk mengetahui pendekatan yang digunakan, menganalisa efisiensinya dan memperkirakan kesalahan yang mungkin terjadi.

2.3 *Internet*

2.3.1 Pengertian *Internet*

Istilah *internet* berasal dari bahasa Latin inter, yang berarti *"antara"*. Secara kata per kata *internet* berarti jaringan antara atau penghubung. Memang itulah fungsinya, *internet* menghubungkan berbagai jaringan yang tidak saling bergantung pada satu sama lain sedemikian rupa, sehingga mereka dapat berkomunikasi. Sistem apa yang digunakan pada masing-masing jaringan tidak menjadi masalah, apakah sistem DOS atau UNIX.

Secara harfiah, *internet* (kependekan dari kata *inter-network*) ialah rangkaian komputer yang berhubung menerus ke beberapa rangkaian. *Internet* ialah sistem komputer umum, yang berhubung secara global dan menggunakan TCP/IP sebagai protokol pertukaran paket (*packet switching communication protocol*). Cara menghubungkan rangkaian dengan kaedah ini dinamakan internetworking.

Sementara jaringan lokal biasanya terdiri atas komputer sejenis (misalnya DOS atau UNIX), *internet* mengatasi perbedaan berbagai sistem operasi dengan menggunakan "bahasa" yang sama oleh semua jaringan dalam pengiriman data. Pada dasarnya inilah yang menyebabkan besarnya dimensi *internet*.

Internet merupakan hubungan antar berbagai jenis komputer dan jaringan di dunia yang berbeda sistem operasi maupun aplikasinya di mana hubungan tersebut memanfaatkan kemajuan media komunikasi (telepon dan satelit) yang menggunakan protokol standar dalam berkomunikasi yaitu protokol TCP/IP.

Dengan demikian, definisi *internet* ialah "jaringannya jaringan", dengan menciptakan kemungkinan komunikasi antar jaringan di seluruh dunia tanpa bergantung kepada jenis komputernya.

2.3.2 Sejarah Internet

Internet dimulai pada tahun 1969. Pada waktu itu Departemen Pertahanan Amerika Serikat membangun sebuah jaringan komputer melalui proyek yang disebut ARPANET (*Advanced Research Projects Administrator Administration Network*) yang dimana mereka menjelaskan dan mendemonstrasikan bagaimana dengan

hardware dan software komputer yang berbasis UNIX dapat melakukan komunikasi dalam jarak yang tidak terhingga melalui saluran telepon. Proyek ARPANET merancang bentuk jaringan, kehandalan, seberapa besar informasi yang dipindahkan dan akhirnya semua standar yang mereka tentukan menjadi asal mula pembangunan protokol baru yang sekarang dikenal sebagai TCP/IP (*Transmission Control Protocol/Internet Protocol*).

Pada awalnya, dibangunnya proyek tersebut adalah untuk kegiatan militer. Pada saat itu Departemen Pertahanan Militer Amerika Serikat merancang jaringan komputer yang tersebar dengan menghubungkan komputer di daerah-daerah vital untuk mengatasi masalah bila terjadi serangan nuklir dan untuk menghindari terjadinya informasi terpusat, yang apabila terjadi perang dapat muda dihancurkan.

Di tahun tersebut, jaringan itu hanya menghubungkan empat komputer yang tersebar di negara California dan Utah di Amerika Serikat, dan pada tahun 1969 mereka membentuk satu jaringan terpadu. Kemudian jaringan ini berkembang begitu cepat dan menghubungkan komputer-komputer yang berada di universitas, lembaga penelitian pemerintah dan instansi militer yang letaknya di kota yang berbeda. Secara umum ARPANET diperkenalkan pada bulan Oktober 1972. Sejak awal, jaringan ini dirancang tetap berfungsi walaupun sebagian jalur rusak akibat perang.

Tidak lama kemudian proyek ini berkembang pesat di seluruh daerah, dan semua universitas di negara tersebut ingin bergabung, sehingga membuat ARPANET kesulitan untuk mengaturnya. Kemudian untuk solusinya, ARPANET dipecah

menjadi dua bagian yaitu “MILNET” yang dilakukan untuk keperluan militer dan “ARPANET” baru yang lebih kecil untuk keperluan non-militer. Oleh sebab itu masing-masing komputer dapat berhubungan dengan komputer lain melalui berbagai macam jalur alternatif. Gabungan kedua jaringan tersebut akhirnya dikenal dengan nama DARPA internet, yang kemudian disederhanakan menjadi internet.

Pada tahun 1980-an, *National Science Foundation* membentuk NSFnet, yang menghubungkan setengah lusin super komputer pada kecepatan tertinggi waktu itu. NSFnet akhirnya mengambil alih ARPAnet.

a. *File Transfer Protocol* (FTP)

FRP (*File Transfer Protocol*) adalah suatu protokol khusus yang memungkinkan untuk mentransfer file dari suatu komputer ke komputer lainnya dalam suaru jaringan internet. Dengan FTP, dapat dilakukan transfer file data grafik, data audio, video, spreadsheet, dokumen, serta file-file dengan format yang bermacam-macam jenisnya.

b. *Terminal Emulator* (Telnet)

Telnet adalah suatu program pengguna internet untuk berinteraksi dengan komputer lain yang seakan-akan pengguna-pengguna fasilitas ini berkerja di komputer lain yang dihubungkannya.

c. *Information Browsing* (Gopher)

Gopher merupakan sarana pertama yang memudahkan pencarian informasi di internet. Gopher memberikan kemudahan untuk menyusun daftar-daftar sehingga mudah untuk mencaru suatu informasi yang diinginkan.

d. *Advanced Browsing* (WWW, Mosaic)

WWW (*World Wide Web*) adalah hypermedia pada internet. Dokumen-dokumen informasi pada WWW disertai dengan grafik berwarna dan mendukung sistem multimedia dan memiliki link ke informasi lainnya. Untuk memperoleh dokumen WWW, kita menggunakan aplikasi browse Mosaic, Netscape, Websurfer, dan lainnya.

2.3.3 Peranan Internet

Sebelumnya penyebaran informasi lebih banyak dilakukan melalui media cetak (surat kabar, majalah, buku, atau brosur). Proses ini dapat memakan waktu yang cukup lama, bahkan hingga berhari-hari. Belum lagi ditambah dengan waktu pengiriman yang sangat bergantung pada lokasi yang dituju.

Kehadiran web menjadikan internet mengalami banyak sekali perubahan. Dengan menggunakan format HTML (*Hypertext Markup Language*), sebuah dokumen dapat memuat teks dan gambar sekaligus. Bahkan dengan perkembangan HTML, dokumen dalam bentuk multimedia (perpaduan grafik, animasi, suara, dan video) mampu ditampilkan.

Internet tidak hanya tempat orang berkomunikasi, tetapi juga menjadi tempat orang mendapatkan berita, informasi, dan pengetahuan. Di negara yang maju, ada yang menggunakan internet sebagai sambungan untuk video. Dengan demikian, informasi yang tersedia di internet menjadi daya tarik semua orang.

2.4 *Content Management System* (CMS)

2.4.1 Pengertian Sistem

Pengertian dan definisi sistem pada berbagai bidang berbeda-beda, tetapi meskipun istilah sistem yang digunakan bervariasi,semua sistem pada bidang-bidang tersebut mempunyai beberapa persyaratan umum, yaitu sistem harus mempunyai elemen, lingkungan, interaksi antar elemen, interaksi antara elemen dengan lingkungannya, dan yang terpenting adalah sistem harus mempunyai tujuan yang akan dicapai.

Berdasarkan persyaratan ini, sistem dapat didefinisikan sebagai seperangkat elemen yang digabungkan satu dengan lainnya untuk suatu tujuan bersama. Kumpulan elemen terdiri dari manusia, mesin, prosedur, dokumen, data atau elemen lain yang terorganisir dari elemen-elemen tersebut. Elemen sistem disamping berhubungan satu sama lain, juga berhubungan dengan lingkungannya untuk mencapai tujuan yang telah ditentukan sebelumnya.

Sebuah sistem terdiri dari bagian-bagian saling berkaitan yang beroperasi bersama untuk mencapai beberapa sasaran atau maksud dan sistem juga adalah

himpunan dari unsur-unsur yang saling berkaitan sehingga membentuk suatu kesatuan yang utuh dan terpadu.

2.4.2 Pengertian Konten

Konten (*Content*) pada intinya, tipe atau unit apapun dari informasi *digital*. Bisa berupa teks, gambar, grafik, *video*, suara, dokumen, catatan dan sebagainya. Atau dengan kata lain apasaja yang kemungkinan bisa di kelola dalam format elektronik.

2.4.3 Pengertian Manajemen Konten

Manajemen Konten (*Content Management*) adalah pengelolaan secara efektif dari konten yang dijelaskan diatas, dengan menggabungkan aturan-aturan, proses dan atau aliran kerja dalam suatu bentuk dimana penyimpanan elektroniknya lebih dipertimbangkan untuk dikelola daripada tidak dikelola.

2.4.4 Pengertian *Content Management System* (CMS)

Content Management System atau lebih populer dengan singkatan CMS, secara ringkas atau singkatnya dapat disebut sebagai sebuah sistem yang mengelola konten atau wacana. Pertama kali muncul sebagai jawaban atas solusi dari kebutuhan manusia akan penyediaan informasi yang sangat cepat. Hanya apabila konten yang akan dikelola tersebut diketahui maka dapat dilakukan penyaringan berdasarkan pilihan yang tersedia.

Sistem sendiri dapat dideskripsikan sebagai sebuah alat atau gabungan atau peralatan yang memfasilitasi produksi yang efisien dan efektif dari keluaran yang diinginkan menggunakan manjemen konten.

a. CMS adalah sebuah alat yang memungkinkan suatu keragaman teknik dan staf non-teknik untuk mebuat, merubah, mengelola dan akhirnya menerbitkan beragam konten (seperti teks, gambar, grafik, *video*, suara, dokumen dsb), sementara itu dibatasi dengan aturan-aturan, proses dan alur kerja yang terpusat memastikan konten tersebut benar.

b. CMS adalah sebuah sistem yang memberikan kemudahan kepada para pengguna dalam mengelola dan mengadakan perubahan isi sebuah *website* dinamis tanpa sebelumnya dibekali pengetahuan tentang hal-hal yang bersifat teknis. Dengan demikian setiap orang, penulis, maupun editor, setiap saat dapat menggunakannya secara leluasa untuk membuat, menghapus atau bahkan memperbaharui isi *website* tanpa campur tangan langsung dari pihak *webmaster* .

c. CMS dapat diartikan sebuah sistem yang digunakan untuk mengelola isi sebuah *web* secara dinamis.

d. CMS dapat diartikan sebuah teknologi baru yang masih asing bagi sebagian orang yang dapat membawa kesuksesan untuk setiap *web.*

Berdasarkan beberapa definisi di atas, secara sederhana dapat dikatakan CMS adalah suatu sistem yang dibuat dengan menggunakan teknologi, dalam hal ini skrip

berbasis *server*, *database server* dan *webserver* oleh seorang yang mempunyai kemampuan memprogram *web* yang sangat baik agar pengelolaan *website* itu dapat dilakukan oleh siapa saja sehingga *content web* dapat selalu dinamis.

2.4.5 Karakteristik CMS

2.4.5.1. Fungsi CMS

Sebuah sistem CMS bisa dikatakan baik apabila telah memenuhi beberapa persyaratan yang meliputi beberapa fungsi pokok yaitu :

a. Fungsi pembuatan isi (*content creation*)

Dalam bagian ini fungsi dibuat agar diarahkan untuk mampu menangani pengisian informasi secara mudah dan efisien, dimana seorang penulis informasi tidak perlu menggunakan tag HTML dalam membuat sebuah halaman *web*.

b. Fungsi manajemen isi (*content management*)

Dalam bagian ini fungsi dibuat untuk mengatur pusat isian, baik penampilan data yang sudah atau belum terpakai sehingga dalam proses penampilan informasinya menjadi mudah.

c. Fungsi menerbitkan isi untuk ditampilkan (*publishing*)

Dalam bagian ini fungsi dibuat untuk memuat fleksibilitas dan pengembangan halaman *web* yang dihasilkan. tampilan (*Layout*) halaman dispesifikasikan melalui sebuah halaman dengan pola yang telah disediakan (*template*)

d. Fungsi penampil isi (*presentation*)

Dalam bagian ini fungsi dibuat untuk menampilkan isi terhadap informasi yang telah tersimpan untuk ditampilkan di atas *browser*.

2.4.5.2. Manfaat CMS

Bila CMS dapat diterapkan dengan baik, ada beberapa manfaat dari penggunaan CMS yang dapat dijabarkan sebagai berikut:

1. Manajemen Data.
2. Mengatur siklus hidup *website*.
3. Mendukung web templating dan standarisasi.
4. Personalisasi website.
5. Sindikasi; Sindikasi memberikan kemungkinan kepada sebuah *website* untuk membagi isinya kepada *website-website* yang lain.
6. Akuntabilitas dan efisiensi.

2.4.5.3. Prinsip CMS

CMS secara prinsip dapat digunakan untuk berbagai macam keperluan dan dalam berbagai kondisi, terutama yang berhubungan dengan dunia *website*, beberapa diantaranya yaitu:

1. Mengelola website pribadi.
2. Mengelola website perusahaan/bisnis.
3. Portal atau website komunitas.
4. Galeri foto, dan lain sebagainya.

5. Forum.

6. Aplikasi E-Commerse.

2.4.5.4. Penggolongan CMS

Karena disadari akan pentingnya CMS, banyak para pakar membuat aplikasi CMS dengan berbagai fitur handal yang ditawarkan, hal ini dapat di lihat dengan maraknya topik mengenai CMS di internet. Beberapa situs *web* yang mengkaji mengenai CMS di internet yaitu www.kyantonius.com, www.oscom.org, www.duniakita.net, dan www.steptwo.com. Oleh karena itu berdasarkan aplikasinya, CMS dibagi menjadi 2 golongan yaitu:

a. CMS Komersil

Dibuat dan dikembangkan oleh perusahaan-perusahaan *software* yang menjalankan usahanya dengan motif mencari keuntungan. Beberapa aplikasi CMS yang ada di pasaran yang cukup terkenal yaitu STELLENT (*Stellent Content Management Suite),* VIGNETTE *(V7 Content Management Suite),* INTERWOVEN *(Team Site 6.0.)*

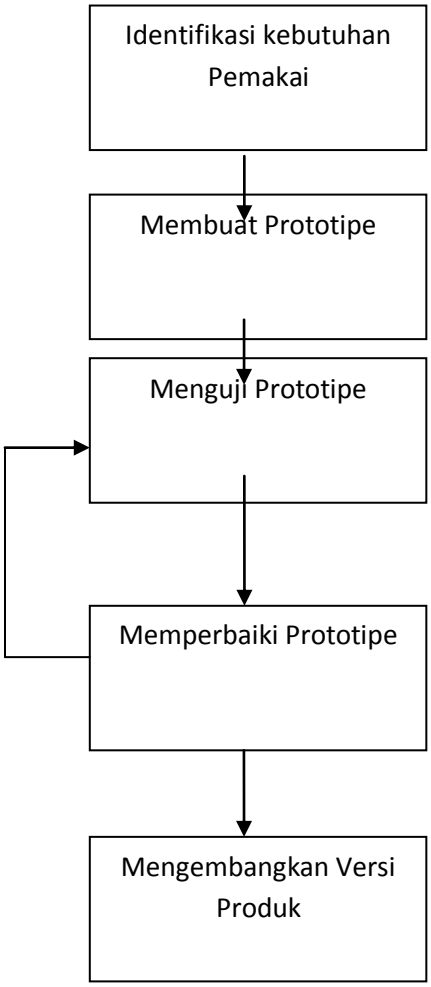
b. CMS *Open source*

Dibuat dan dikembangkan oleh sekelompok orang atau perusahaan yang intinya memberikan sebuah alternatif murah dan terjangkau kepada para pengguna. Tersedia secara gratis dan dapat dipergunakan sesuai dengan kebutuhan tanpa ada batasan. CMS jenis ini juga memberikan akses kepada penggunanya untuk mengetahui kode-kode pemograman yang disertakan,

sehingga memudahkan pengguna memodifikasi CMS di masa-masa yang akan datang. Kelebihan CMS jenis ini, yaitu kode pemrograman terbuka untuk umum, sehingga apabila ada *bugs*, para pengguna dapat saling bahu-membahu dalam hal melacak danmemperbaiki *bugs* yang ada, menambah dan meningkatkan fungsi dan kemampuan CMS dan memberikan dukungan teknis dan non-teknis kepada yang membutuhkan. Beberapa jenis aplikasi CMS *Open Source* yang ada dipasaran seperti ZOPE (*Content Management Framework)*, OPEN CMS (*Open CMS*), POSTNUKE (*Post Nuke*), PHP Nuke, MAMBO *OPEN SOURCE* (*Mambo Open source*).

2.5 Siklus Hidup Sistem (The Classic Life Cycle)

Siklus hidup sistem adalah proses evolusioner yang diikuti dalam menerapkan sistem atau subsistem informasi berbasis komputer. Dengan metode pengembangan *prototype* maka dapat menawarkan bagi pengembang system apabila tidak memiliki kepastian terhadap efisiensi algoritma, kemampuan penyesuaian dari sebuah system operasi atau bentuk-bentuk yang harus dilakukan oleh interaksi manusia dengan mesin. Dilihat dari situasi tersebut metode *prototype* menawarkan pendekatan yang terbaik.



- Pengenbang dan pemakai bertemu
- Pemakai menjelaskan kebutuhan user
- Pengembang mulai membuat *prototype*
- Pemakai menguji prototype dan memberikan kritikan atau saran
- Pengembangan melakukan modifikasi sesuai dengan masukan dari pemakai
- Pengembang merampungkan system sesuai dengan masukan terakhir

Gambar 2.1 Model *Prototype*

The *classic life cycle* merupakan pendekatan perkembangan perangkat lunak yang sistematik dan berurut, dimulai pada tingkatan dan kemajuan sistem pada seluruh analisis, perancangan, pengkodean, pengujian, dan pemeliharaan. Aktivitas-aktivitas dalam *the life cycle* ini melingkupi :

1. Rekayasa dan pemodelan sistem (*system engineering*)

Oleh sebab perangkat lunak selalu merupakan bagian dari sebuah sistem yang lebih besar, kerja dimulai dengan membangun syarat semua elemen sistem dan mengalokasikan beberapa *subset* dari kebutuhan ke perangkat lunak tersebut. Rekayasa dan pemodelan sistem menyangkut pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis dan disain tingkat puncak.

2. Analisis kebutuhan perangkat lunak

Pada aktivitas ini, proses pengumpulan kebutuhan diintensifikasi dan difokuskan, khususnya pada perangkat lunak. Kebutuhan baik sistem maupun perangkat lunak didokumentasikan.

3. Desain

Desain perangkat lunak sebenarnya adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda; struktur data, arsitektur perangkat luna, representasi *interface*, dan algoritma *procedural*. Proses desain menerjemahkan kebutuhan dalam sebuah representasi perangkat lunak yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode.

4. Pengkodean

Desain harus diterjemahkan kedalam bentuk bahasa mesin yang bisa dibaca. Langkah pengkodean dapat diselesaikan secara mekanis.

5. Pengujian

Proses pengujian berfokus kepada logika internal perangkat lunak memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional, mengarahkan pengujian untuk menemukan kesalahan dan memastikan input yang dibatasi akan memberikan hasil aktual, sesuai hasil yang dibutuhkan.

6. Pemeliharaan

Perangkat lunak akan mengalami perubahan setelah disampaikan kepada pelanggan. Perubahan akan terjadi karena kesalahan-kesalahan ditentukan, karena perangkat lunak harus disesuaikan untuk mengakomodasi perubahanperubahan didalam lingkungan eksternal, atau karena pelanggan membutuhkan perkembangan fungsionalitas atau unjuk kerja. Pemeliharaan perangkat lunak mengaplikasikan lagi setiap fase program sebelumnya dan tidak membuat yang baru lagi.

2.6 Delapan Aturan Emas Perancangan Antar Muka

Berikut ini merupakan prinsip-prinsip yang mendasari perancangan antar muka yang dapat diterapkan dalam banyak sistem:

1. Adanya konsistensi (*strive for consistency*.

Aturan ini mungkin sulit dilakukan karena terdapat banyak bentuk dari konsistensi. Serangkaian konsistensi aksi dibutuhkan dalam situasi yang

serupa, istilah yang sama dipergunakan dalam menu dan layar bantu, konsisten dalam warna, rancangan, huruf, dan lain-lainnya.

2. Pengguna yang sering menggunakan sistem dapat menggunakan *shortcut* (*enable frequent user to use shortcut*).

Jika frekuensi pengguna suatu sistem meningkat, user ingin mengurangi jumlah interaksi. Singkatan, kunci yang spesial, perintah tersembunyi, dan fasilitas makro dihargai oleh pengguna yang sering menggunakan sistem.

3. Memberikan *feedback* yang berisi informasi (*offer information feedback)*.

Setiap aksi pengguna harus mendapat feedback. Bagi aksi yang sering dan kecil, respon dapat berupa sesuatu yang sederhana. Akan tetapi, untuk aksi yang jarang dan utama, respon harus lebih penting.

4. Merancang dialog dengan penutupan (*design dialogs to yield closure*).

Serangkaian aksi seharusnya diorganisasikan ke dalam kelompok: pemulaan, pertengahan, dan akhir. Feedback yang informative pada penyelesaian suatu kelompok aksi-aksi memberikan kepuasan penyelesaian bagi operator.

5. Menawarkan pencegahan kesalahan dan penanganan kesalahan sederhana (*offer error prevention and error handling*).

Perancangan suatu sistem sedapat mungkin membuat pengguna tidak dapat melakukan kesalahan serius. Jika pengguna melakukan kesalahan,

sistem harus mendeteksi kesalahan tersebut dan memberikan instruksi yang mudah dan jelas untuk perbaikan.

6. Mengijinkan pembalikan aksi yang mudah (*permit easy reversal of actions*).

Sedapat mungkin, pembalikan aksi dapat dilakukan. Fasilitas ini mengurangi kegelisahan pengguna karena pengguna tahu bahwa suatu kesalahan dapat dibatalkan, sehingga mendorong pengguna untuk melakukan penjelajahan pilihan-pilihan yang tidak biasa digunakan.

7. Mendukung tempat pengontrolan internal (*support internal locus of control*).

   Operator yang berpengalaman mempunyai perasaan yang kuat bahwa para operator tersebut bertanggungjawab terhadap suatu sistem dan sistem memberikan respon terhadap aksi-aksi yang dilakukan.

8. Mengurangi beban ingatan jangka pendek (*reduce short-term memory load*).

   Keterbatasan pemrosesan informasi manusia dalam ingatan jangka pendek memerlukan penayangan yang sederhana, menggabung penayangan banyak halaman dan waktu pelatihan yang cukup untuk kode dan rangkaian aksi.

2.7 Pengembangan Aplikasi Berbasis *Framework*

PHP adalah sebuah bahasa pemrograman yang memungkinkan seorang developer (*programmer* atau *system analist*) membuat sebuah aplikasi berbasis *web* yang *powerful* sekaligus mampu mengampu database berskala besar. Dalam perkembangannya, seorang *programmer* PHP seringkali dituntut untuk menyelesaikan berbagai macam aplikasi dengan tingkat kerumitan yang cukup tinggi dalam waktu singkat.

Di sisi lain, *programmer* juga dituntut untuk menciptakan sebuah dasar aplikasi yang bisa dikembangkan menjadi aplikasi lain dengan skala yang lebih besar dengan melibatkan banyak anggota tim.

Aplikasi web berskala besar seringkali diasosiasikan dengan indikasi-indikasi sebagai berikut :

a. Diakses oleh banyak orang (*public access*).
b. Melibatkan database dengan skala record diatas 1000.
c. Mempunyai banyak modul, seperti modul berita, modul administrasi, modul keuangan, modul pencarian tingkat lanjut, *modul polling* dan sebagainya.
d. Dikerjakan oleh sebuah tim pengembang dengan spesialisasi tugas.

Berbicara tentang aplikasi *web* berskala besar tentu tidak bisa dilepaskan dari pembagian peran anggota tim. Aplikasi *web*, terutama *web* berskala besar, tidak hanya membutuhkan seorang *programmer* saja, akan tetapi melibatkan pula seorang *web desainer*, *system analist*, *database maintainer*, manajer keuangan, manajer riset

dan promosi dan manajer proyek yang akan mengatur jalannya pembuatan, pengembangan dan pemeliharaan aplikasi tersebut. Tingkat kerumitan dan kesamaan cara pandang inilah yang melahirkan konsep kerangka kerja (*framework*) dalam pengembangan aplikasi berbasis *web*. *Framework* memudahkan anggota tim untuk bekerja bersama dengan satu cara pandang, sekaligus memudahkan pembagian kerja. Pada umumnya, *framework* terdiri dari file-file pustaka berupa kelas-kelas PHP yang bisa digunakan dan dikolaborasikan dalam aplikasi yang sedang dikembangkan.

Di *internet*, tersedia puluhan, bahkan ratusan *framework* dan *Content Management System* (CMS). Sebut saja Mojavi, Prado, Agavi, Seagull, Code Igniter, Fusebox, PHP Nuke, Post Nuke, Mambo, Joomla, Wordpress, Blogspot, Aura, Endonesia, dan masih banyak lagi. Ada beberapa *developer* yang menyediakan *file* inti (*core*) *framework* saja, beserta manualnya, tanpa memberikan contoh-contoh aplikasi secara detail, misalnya Fusebox dan Mojavi (saat ini riset Mojavi terhenti, tapi dikembangkan menjadi framework baru, yakni Agavi). Ada juga *framework* yang benar-benar dibuat untuk membangun aplikasi *enterprise*, memberikan contoh yang detail, manual yang lengkap, namun memiliki ukuran besar dan memerlukan instalasi PHP 5 supaya bisa berjalan dengan baik, misalnya Prado. Terkadang orang mengartikan framework adalah dasar dari CMS. Di sisi lain, CMS tak lain juga sebuah *framework* dengan bermacam modul yang datang bersamanya.

2.8 Terminologi MVC

MVC singkatan dari *Model-View-Controller*. MVC adalah pola desain software yang membantu secara logis memisahkan kode program, membuatnya lebih *reusable*, *maintainable*, dan lebih baik secara keseluruhan.

***Model* :** Representasi aktif dari struktur database. *Framework* berbasis MVC menekankan pentingnya disain *database* yang *valid*, termasuk hubungan antar tabel dalam database tersebut. *Model* memungkinkan *developer* untuk melakukan *query* antar *database* (bila memang diperintahkan oleh *controller*). Sebuah aplikasi MVC yang baik memisahkan antar tampilan dan logika kerja (*logic*). Model adalah sesuatu yang mewakili basis data yang diolah sedemikian rupa oleh *controller* (*logic*), untuk ditampilkan (*view*) hasil olahannya.

***View* :** Tampilan luar dari aplikasi. *User* akan berhadapan langsung dengan tampilan ini. *View* tidak boleh mengandung *bussiness logic*, alias logika kerja aplikasi. Bisa dikatakan, view adalah template dari tampilan aplikasi yang memudahkan untuk menggantinya di kemudian hari.

Controller : Inilah bagian yang paling signifikan dari sebuah aplikasi berbasis MVC. Benar dan tidaknya hasil olahan data akan sangat tergantung dari logika kerja aplikasi yang tersusun pada bagian controller ini. Controller akan mengolah data dari model, menyimpannya dalam variabel-variabel, dan menampilkannya pada view.

2.9 CakePHP

2.9.1 Profil

CakePHP adalah sebuah *framework open source* yang digunakan untuk mengembangkan aplikasi web dengan dasar kerja **CRUD (*Create, Read, Update, Delete*)**. CakePHP juga menjadi salah satu *framework* pilihan yang memungkinkan developer untuk membuat sebuah aplikasi web dengan karakter pengembangan **RAD (*Rapid Application Development*)**, yang memungkinkan untuk digunakan dan dikembangkan menjadi aplikasi lain yang lebih kompleks. CakePHP terdiri dari *file-file* pustaka (*library*), kelas-kelas, dan infrastruktur runtime yang terinspirasi oleh *framework* Ruby on Rails. CakePHP juga banyak digunakan oleh para *programmer* yang memilih untuk bekerja dengan struktur yang rapi dan padat tanpa kehilangan fleksibilitas pengembangan *framework.*

Pada tahun 2005, Michal Tatarynowicz mulai menulis beberapa kelas untuk sebuah dasar aplikasi RAD dengan menggunakan bahasa pemrograman PHP. Ia menyadari bahwa beberapa kelas yang ia ciptakan sangat memungkinkan untuk dikembangkan menjadi sebuah framework yang lebih lengkap dan praktis. Akhirnya, Michal mempublikasikan hasil kerjanya di bawah lisensi MIT Amerika Serikat, menamainya dengan Cake dan menawarkan pengembangannya pada komunitas developer PHP dan saat ini proyek tersebut dikenal dengan nama CakePHP.

CakePHP bersifat terbuka, bebas didapatkan, dan bebas dikembangkan. Komunitas pengembang CakePHP mengajak seluruh programmer PHP untuk

bergabung dalam Cake Software Foundation Inc. yang membawahi proyek pengembangan CakePHP dan aplikasi berbasis CakePHP. Informasi tentang CakePHP bisa didapatkan pada situs resmi Cake Software Foundation di website www.cakephp.org.



Gambar 2.2 Logo Resmi CakePHP

Selain *website* resmi, para pengembang CakePHP juga biasa berdiskusi dengan menggunakan IRC (*Internet Relay Chat*). Di sini bisa dijumpai banyak ***Bakers***, sebutan untuk mereka yang menjadi developer CakePHP. Forum diskusi CakePHP ini bisa dikunjungi di channel #cakephp pada server irc.freenode.com.

2.9.2 Kelebihan

Selama ini, pemahaman yang kita peroleh tentang aplikasi *web* berbasis PHP adalah sebuah aplikasi yang terdiri dari file-file PHP yang berisi kode-kode dan tag html yang dicampur dengan kode-kode PHP atau lebih dikenal dengan nama spaghetti code.

Pengembangan aplikasi *web* dengan cara konvensional seperti ini tidaklah keliru, namun akan menjadi masalah besar apabila aplikasi ini membutuhkan banyak fitur dan pengembangan di kemudian hari. *Framework* CakePHP menjadi pilihan untuk mengatasi berbagai permasalahan yang muncul dalam pengembangan aplikasi *web*, karena beberapa kelebihannya, antara lain :

a. *Open Source*

CakePHP bebas didapatkan dan dikembangkan. CakePHP mempunyai lisensi MIT. Ini adalah salah satu syarat yang baik untuk berkembangnya sebuah *framework*.

b. Riset yang terorganisir dengan baik

Pengembangan framework bukanlah pekerjaan yang selesai dalam waktu 1-2 hari. Semakin terorganisirnya pengembangan sebuah *framework* akan berbanding lurus dengan kualitas framework yang dihasilkan. CakePHP memenuhi syarat ini dan dikembangkan dalam riset yang terorganisir dan berkesinambungan di bawah Cake Software Foundation.

c. Dokumentasi yang lengkap

Framework yang baik harus mempunyai dokumentasi yang lengkap, karena sekumpulan kelas tanpa dokumentasi tak lebih dari sebongkah file yang asing dan menambah masalah *developer*. Untuk referensi lebih lengkap (gunakan manual dan API CakePHP), silahkan buka http://www.cakephp.org.

d. Kompatibel dengan PHP 4 dan PHP 5

CakePHP berjalan mulus di server Apache yang menggunakan PHP 4 maupun PHP 5. Fleksibilitas dan kompatibilitas inilah yang banyak menarik minat para programmer untuk menggunakan *framework* CakePHP sebagai dasar untuk pengembangan aplikasi mereka.

e. Konsep CRUD terintegrasi

CakePHP menerapkan konsep CRUD (Create, Read, Update, Delete) terintegrasi yang membantu interaksinya dengan *database* dan menyederhanakan *query*.

f. Arsitektur OOP dan MVC

Kelas-kelas yang menjadi dasar dari CakePHP ditulis dengan konsep Object Oriented *Programming* yang memudahkan *programmer* untuk melakukan penambahan, pengurangan dan modifikasi kelas dan fungsi yang digunakan. CakePHP menggunakan arsitektur Model, *View, Controller* (MVC) yang memisahkan database logic (Model), presentation logic (View) dan *bussiness logic* (Controller).

g. Fitur Scaffolding

Ya, barangkali inilah fitur yang jarang didapat pada *framework* lain. CakePHP mempunyai fitur yang mampu men-generate prototipe aplikasi Sebelum menyusun source code-nya secara lengkap. Cukup tambahkan variabel $scaffold di controller, aplikasi *web* sudah bisa dinikmati.

h. Manajemen akses bagi user

CakePHP memungkinkan pengaturan user dan hak aksesnya dalam aplikasi yang dikembangkan, dengan sarana yang lebih mudah dipahami. Fitur ini dikenal dengan nama *Access Control List* (ACL).

i. Validasi dan sanitasi data

CakePHP mempunyai kelas-kelas dasar yang membantu programmer untuk melakukan sanitasi dan validasi data pada aplikasinya.

j. Komponen *Security, Session,* dan *Request Handling* yang terintegrasi

CakePHP menyediakan komponen-komponen untuk menangani masalah Session, Keamanan (Security) dan *Request Handling* yang sudah terintegrasi dalam kelas dasar CakePHP. Developer cukup menyertakan komponen tersebut pada controller aplikasi dan memasukkan parameter-parameter untuk mendapatkan hasil yang diinginkan.

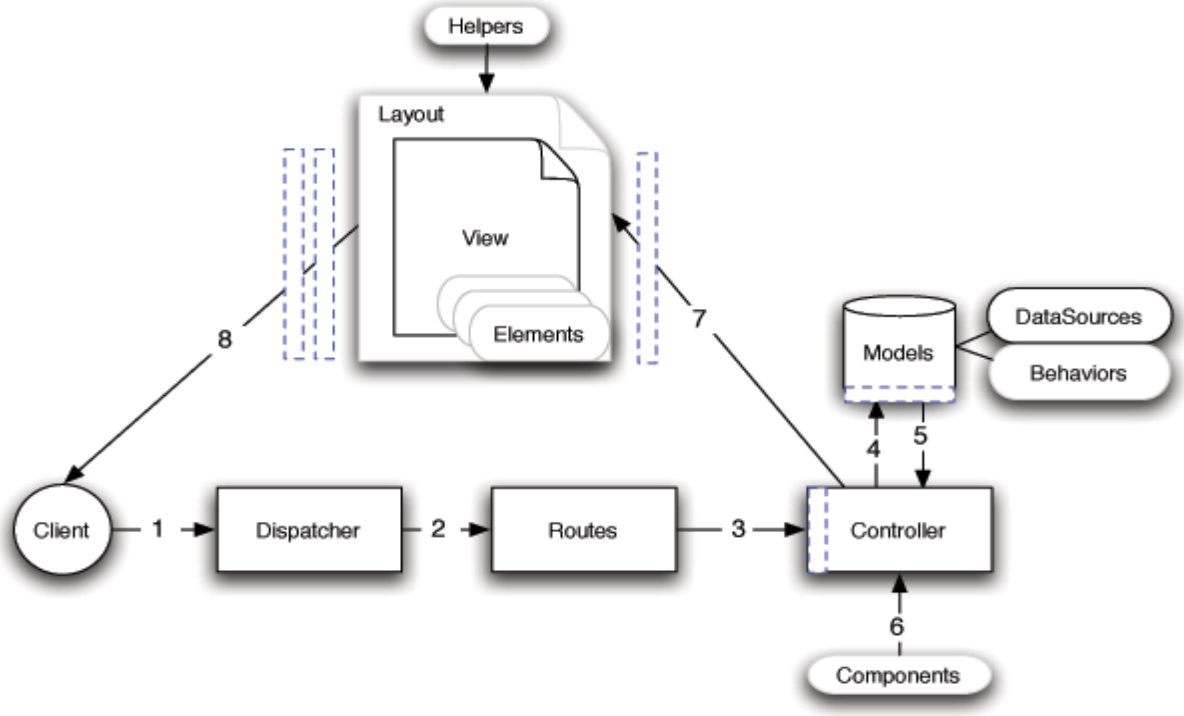
k. Metode templating yang simpel

Sebagaimana *framework-framework* yang lain, CakePHP juga mendukung metode *templating* yang sangat mudah digunakan untuk membantu *programmer* dan *web* disainer menciptakan tampilan aplikasi yang indah dan mudah dimodifikasi. CakePHP mempunyai kelas helper yang mendukung templating HTML, Ajax, Javascript dan masih banyak lagi.

l. Cocok untuk segala strukutur direktori

CakePHP mempunyai sistem konfigurasi yang menyediakan berbagai macam pilihan konfigurasi, sesuai dengan struktur direktori pengembangan aplikasi berbasis framework CakePHP.

2.9.3 Cara Kerja



Gambar 2.3 Struktur kerja CakePHP

Dari gambar di atas, cara kerja CakePHP sebagian besar sudah bisa dijelaskan.

1. Client mengklik link ke http://www.example.com/books/beli, dan browser client membuat request ke web server.

2. Route akan menterjemahkan URL untuk mengekstrak parameter dari request, yaitu : controller, action dan argument lainnya yang mempengaruhi logic bisnis saat request.

3. Dengan menggunakan route, URL yang direquest dipetakan ke action di controller (sebuah method dalam class controller). Dalam kasus ini method beli() di dalam class BooksController. Callback beforeFilter() di controller dipanggil sebelum logic action controller dieksekusi.

4. Controller dapat menggunakan model untuk memperoleh data aplikasi. Dalam contoh ini, controller menggunakan model untuk mengambil data pembelian terakhir dari database. Penggunaan model tidaklah diperlukan tapi semua controller CakePHP secara inisial membutuhkan setidaknya satu model.

5. Setelah model memperoleh data, model akan memberikannya ke controller Callback untuk model dapat dipanggil.

6. Controller dapat menggunakan component untuk memanipulasi data atau melakukan operasi lainnya (missal untuk manipulasi session, autentifikasi, atau mengirim email).

7. Saat controller selesai menggunakan model dan component untuk menyiapkan data, controller dapat memberikan data tersebut ke view dengan pemanggilan method set(). Callback di controller dapat dieksekusi sebelum

data dikirim. Kemudian logika di view dieksekusi, element dan / atau helper dapat disertakan (include). Secara default, view dirender di dalam layout.

8. Kemudian callback controller tambahan (seperti afterFilter()) dapat dipanggil. Setelah itu kode view yang dirender diberikan ke peraba Client.

2.10 *Hyper Text Markup Language* (HTML)

HTML merupakan kepanjangan dari *Hyper Text Markup Language* adalah suatu bahasa yang digunakan untuk membuat halaman-halaman hypertext (hypertext page) pada internet. Dengan konsep hypertext ini, untuk membaca suatu dokumen tidak harus melakukannya secara urut, baris demi baris, atau halaman demi halaman. Tetapi tidak dapat dengan mudah melompat dari satu topik ke topic lainnya yang disukai, seperti halnya jika melakukan pada online Help dari suatu aplikasi Windows. HTML dirancang untuk digunakan tanpa tergantung pada suatu platform tertentu (platform independent).

2.11 PHP

Menurut Luke Welling dan Laura Thomson(2001,p1), PHP adalah *server-side scripting language* yang didesain secara spesifik untuk web. Dalam *page HTML*, dapat dimasukkan *code* PHP yang akan dieksekusi setiap kali halaman dikunjungi. PHP *code* diterjemahkan di *web-server* dan dirubah menjadi HTML atau output lain yang akan dilihat oleh pengunjung halaman.

PHP diciptakan oleh Rasmus Lerdorf, seorang pemrogram C yang handal. Semula PHP hanya digunakan untuk mencatat jumlah pengunjung pada

homepagenya. Rasmus adalah seorang pendukung open source. Karena itulah ia mengeluarkan Personal Home Page Tools versi 1.0 secara gratis. Setelah mempelajari YACC dan GNU Bison, Rasmus menambah kemampuan PHP 1.0 dan menerbitkan PHP 2.0.

PHP mudah dibuat dan cepat dijalankan, PHP dapat berjalan dalam web server yang berbeda dan dalam sistem operasi yang berbeda pula. PHP dapat berjalan di sistem operasi UNIX, Windows 98, Windows XP, Windows NT, dan Macintosh.

Kelebihan PHP :

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 relative 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 system.

2.12 MySQL

MySQL adalah salah satu jenis database server yang sangat terkenal. Kepopulerannya disebabkab bersifat free (tidak perlu membayar untuk menggunakannya) pada berbagai platform (kecuali pada windows, yang bersifat software atau perlu membayar setelah melakukan evaluasi dan memutuskan digunakan untuk keperluan produksi). Perangkat lunak MySQL sendiri bisa di download dari http: // www.mysql.org atau http://www.mysql.com.

MySQL termasuk jenis RDMS *(Relational Database Management System)*. Itulah sebabnya istilah seperti tabel, baris, dan kolom digunakan pada MySQL. Pada MySQL, sebuah database mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom.

Menurut Menurut Luke Welling dan Laura Thomson(2001,p1), MySQL adalah sebuah relational database management systems yang sangat cepat dan kuat.

MySQL adalah sebuah perangkat lunak sistem management basis data SQL atau DBMS yang multithread, multiuser, dengan sekitar 6 juta instalasi diseluruh dunia. MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius. Tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi

mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Untuk melakukan administrasi dalam basis data MySQL, dapat menggunakan modul yang sudah termasuk yaitu *command-line* (perintah: mysql dan mysqladmin). Juga dapat diunduh dari situs MySQL yaitu sebuah modul berbasis grafik (*GUI*): *MySQL Administrator* dan *MySQL Query Browser*. Selain itu terdapat juga sebuah perangkat lunak gratis untuk administrasi basis data MySQL berbasis web yang sangat populer yaitu phpMyAdmin.