BAB II
TINJAUAN PUSTAKA

II.1 Teknologi yang digunakan

II.1.1 PHP

PHP (Hypertext processor) merupakan sebuah bahasa *server side scripting* yang sebagian sintaksnya mirip dengan bahasa pemograman C, Java, dan Perl yang ditambah dengan beberapa fungsi PHP yang spesifik (PHP documentation group, 2004)

II.1.1.1 Kelebihan dari PHP

Kelebihan dari pemakaian PHP sebagai bahasa pemograman aplikasi *web* yaitu :

II.1.1.1.1 Kepraktisan

PHP diciptakan dengan kepraktisan sebagai bagian dalam perancangannya. Tujuan awal dari pembuatnya bukan untuk membuat bahasa yang baru, namun untuk menyelesaikan permasalahan yang saat itu tidak memiliki solusi yang siap langsung digunakan, lebih jauh banyak dari evolusi awal php bukanlah hasil dari tujuan terpisah yakni untuk memperbaiki bahasa itu sendiri, tetapi lebih bertujuan untuk meningkatkan fungsionalitas bahasa PHP terhadap penggunanya. Hasilnya adalah sebuah bahasa yang minimalis, baik dalam hal kebutuhan pengguna dan kebutuhan sintaks bahasa. Bagi pemula, sebuah script untuk menghasilkan suatu output dalam bahasa PHP dapat berjumlah sesedikit mungkin, yakni hanya satu baris. Tidak seperti bahasa C, yang mewajibkan menggunakan library tambahan. Berikut ini adalah contoh penggunaan *script* php untuk menampilkan tanggal saat ini.

```
<?php echo date("F j, Y"); ?>
```

Contoh lain yang menggambarkan kecenderungan PHP dalam hal kesederhanaan yaitu kemampuannya dalam *nested function* (fungsi bersarang).

Sebagai contoh, pengguna dapat memberikan perubahan pada sebuah nilai dibaris yang sama dengan menumpukkan fungsi pada urutan tertentu, pada kasus berikut yang menghasilkan *pseudorandom string* yang terdiri dari lima karakter *alphanumeric*, a3jh8 sebagai contohnya.

```
<?php $randomstr = substr(md5(microtime()), 0, 5); ?>
```

Php adalah sebuah bahasa yang tidak memiliki aturan yang ketat, yakni tidak dibutuhkan untuk membuat, mengganti *type* atau menghapus sebuah variable secara terpisah, walau pengguna tidak dicegah untuk melakukannya. PHP menangani masalah itu secara internal, membuat variable pada saat mereka dipanggil dalam sebuah *script*, dan menggunakan sebuah formula secara otomatis mengubah tipe sebuah variable.sebagai contoh, PHP menganggap statement berikut ini adalah benar.

```
<?php

        $number = "5"; /* number adalah sebuah string */

        $sum = 15 + $number; /*menjumlahkan integer dengan
        string untuk menghasilkan integer */

        $sum = "twenty";    /*menimpa $sum dengan sebuah
        string*/

?>
```

PHP juga secara otomatis menghancurkan variable dan mengembalikan sumber daya yang digunakan perangkat keras kepada sistem ketika script selesai digunakan, pada kasus ini dan kasus – kasus lainnya, dengan mencoba untuk menangani banyak aspek administrtaif programming secara internal, PHP memungkinkan developer untuk lebih berkonsentrasi penuh pada aplikasi yang dikerjakannya.

II.1.1.1.2 Kekuatan

Pada awal pengenalan PHP5 secara tidak langsung mengarahkan pada fakta bahwa versi terbaru ini lebih kualitatif daripada kuantitatif dari versi

sebelumnya. Pada versi major sebelumnya (PHP4) dilengkapi dengan banyak tambahan kepada *default library* PHP, untuk memingkatkan beberapa ratus fungsi per-release. Saat ini terdapat 113 *library* yang tersedia, yang secara bersama mengandung lebih dari 1000 fungsi. Walaupun pengguna mengetahui kemampuan PHP untuk menggunakan database, memanipulasi informasi form , membuat halaman secara dinamis, pengguna mungkin mengetahui php mampu melakukan hal hal sebagai berikut :

(1). Membuat dan memanipulasi Macromedia Flash, Gambar, dan file PDF.
(2). Mengevaluasi password untuk pengenalan pola dengan membandingkannya dengan kamus bahasa dan secara mudah menemukan sebuah pola.
(3). Berkomunikasi dengan Lightweight Directory Access Protocol (LDAP)
(4). Mengartikan kalimat bahkan yang paling kompleks dengan menggunakan Reguler Expression baik POSIX atau PERL.
(5). Melakukan authentifikasi *User* terhadap informasi login yang disimpan dalam file biasa, database, maupun Microsoft Active Directory.
(6). Berkomunikasi dengan banyak jenis protokol, termasuk IMAP, POP3, NNTP, DNS, dan lain sebagainya.
(7). Berkomunikasi dengan banyak jenis solusi pemroses kartu kredit

II.1.2 MVC

MVC (Model View Controller) merupakan sebuah metode untuk mengorganisasikan sebuah website dinamis (Upton, 2007). Dengan penjelasan tiap bagian sebagai berikut :

(1). *Models* adalah object yang merupakan representasi dari data pokok.
(2). *View* menampilkan keadaan dari *model*. Komponen ini bertanggungjawab untuk menampilkan informasi pada pengguna akhir.
(3). *Controller* menawarkan untuk mengubah keadaan *model*. *Controller* juga bertugas untuk mencari informasi dari *model* dan menyediakan data dinamis untuk *views*.

Hasilnya ketika terjadi permintaan (request) dari user / pengguna dilakukan proses dengan urutan sebagai berikut :

(1). *Client* melakukan proses *request* terhadap *Controller* melalui *dispatcher.*

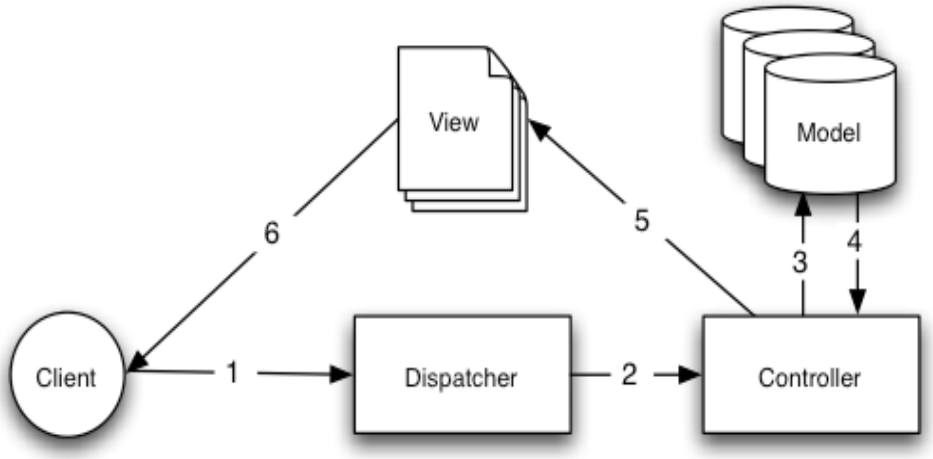
(2). *Dispatcher* menunjuk *controller* yang sesuai dengan *request client*

(3). *Controller* merespon dengan meminta data dari *Model*.

(4). *Model* memberikan data yang diminta oleh *controller.*

(5). *Controller* mengolah tampilan dan memberikannya pada *View*.

(6). *View* mengembalikan tampilannya kepada pengguna yang meminta layanan.



Gambar II .1 Alur MVC
(sumber http://book.cakephp.org)

Pemisahan aplikasi menjadi tiga buah komponen memberikan beberapa kelebihan diantaranya adalah :

(1). Meningkatkan skalabilitas (kemampuan aplikasi untuk berkembang), jika aplikasi menjadi lambat disebabkan karena lambatnya akses database, sebagai contoh, anda dapat meng upgrade hardware yang menjalankan database tanpa ada efek terhadap komponen lainnya.

(2). Membuat perawatan lebih mudah, karena tiap komponen memiliki ketergantungan rendah pada komponen lainnya, membuat perubahan terhadap

suatu bagian (baik memperbaiki atau menambah fungsionalitas), tidak menimbulkan efek terhadap komponen lain.

(3). Membuat aplikasi mudah dipindahkan, pemisahan yang jelas dari kode tiap komponen berarti tiap bagian dapat ditaruh dimesin terpisah jika diperlukan.

II.1.3 jQuery

jQuery adalah sebuah framework javascript yang bersifat unobstrusive (yang tidak membutuhkan teknik binding event pada HTML). jQuery memungkinkan penggunanya untuk memodifikasi bagian dari elemen HTML menggunakan *CSS (Cassade Style Sheet)* selector.

Fungsi utama dari jQuery dapat disebut $, yang digunakan untuk menggunakan *CSS (Cassade Style Sheet)* selector dan ditempatkan didalam function document.ready() seperti ini :

```
$(document).ready(function(){

      //script diletakkan disini

});
```

Penulisan ini mengakibatkan script (jQuery dan script lainnya) dijalankan ketika semua kode HTML telah diload pada server client.

II.1.3.1 Menggunakan jQuery

Jika pengguna ingin menggunakan jQuery untuk mendapatkan element berdasarkan ID, pengguna cukup mengetikkan sebagai berikut

```
Var money = $(#money);
```

menunjukkan sebuah selector ID pada CSS atau mendapatkan element dapat juga dilakukan dengan cara

```
Document.getElementById("money");
```

Untuk mendapatkan group dari element berdasarkan nama tag pengguna cukup memberikan $ pada element tipe selector :

```
Var paragraphs = $("p");
```

Dan untuk mendapatkan group dari element berdasarkan nama Class, pengguna cukup menggunakan Class Selector

```
Var tables = $("#content table.datatable");
```

Tables adalah sebuah array dari element table yang merupakan turunan dari element dengan ID content, dan yang memiliki sebuah Class datatable.

II.1.3.2 Kelebihan jQuery

Kelebihan dari penggunaan jQuery dapat dilihat dari adanya kemudahan sebagai berikut :

(1). Mengakses bagian dari halaman. Tanpa adanya library javascript, banyak baris code harus ditulis untuk menjelajahi *tree document object model* (DOM), dan untuk mendapatkan bagian tertentu dari struktur dokumen HTML. jQuery menawarkan mekanisme selector yang lebih efisien untuk mengambil bagian dari dokumen secara tepat yang akan diperiksa atau dimanipulasi.

(2). Memodifikasi tampilan dari sebuah halaman. CSS menawarkan metode yang kuat dari mempengaruhi cara bagaimana sebuah dokumen di render, tetapi kemudian gagal ketika banyak web browser tidak mengikuti standar yang sama. jQuery dapat menjembatani hal ini dengan memberikan standar yang sama dengan semua browser. Sebagai tambahan, jQuery dapat mengubah atau lebih class yang diterapkan pada satu bagian dari dokumen bahkan selama halaman tersebut direnderer.

(3). Mempengaruhi isi dari sebuah halaman, tidak terbatas hanya perubahan tampilan, jQuery dapat mengubah isi dari dokumen itu sendiri hanya dengan beberapa perintah sederhana. Teks dapat diubah, gambar dapat disisipkan atau ditukar, list dapat diatur kembali, atau seluruh struktur dari html dapat ditulis ulang dan diperluas, semua dengan API yang mudah untuk digunakan.

(4). Merespon interaksi pengguna didalam halaman. Interaksi yang paling kompleks tak berguna jika tidak dapat dikendalikan kapan waktunya.

Pustaka jQuery menawarkan cara yang lebih elegan untuk menangkap banyak jenis dari event, seperti jika pengguna menekan sebuah link, tanpa harus menambahkan kode HTML pada event handler. Pada saat yang bersamaan, API event handling menghancurkan ketidakkonsistensian browser yang sering mengganggu pengembang web.

(5). Menambah animasi pada sebuah halaman. Untuk secara efektif mengimplementasi interaktifitas, seorang desainer juga menyediakan respon balik secara visual kepada pengguna. Pustaka jQuery memfasilitasi hal tersebut dengan menyediakan banyak efek seperti timbul – hilang, pergeseran, dan juga *toolkit* untuk membuat sesuatu yang baru.

(6). Mengambil informasi dari server tanpa harus melakukan load kembali halaman, pola kode seperti ini yang dikenal sebagai *Asynchronous javascript and xml (AJAX)*, dan membantu pengembang web dalam membuat sebuah situs yang *responsive* dan kaya akan fitur. Pustaka jQuery menghapus kompleksitas dari *browser* tertentu dalam proses ini, yang memungkinkan pengembang untuk fokus pada fungsionalitas server.

(7). Menyederhanakan tugas jQuery umum. Sebagai tambahan dari fitur spesifik dari jQuery, pustaka ini menyediakan perbaikan kepada javascript dasar untuk membangun hal hal, seperti iterasi dan manipulasi array.

II.1.4 MySQL

MySQL adalah *database* relasional buatan MySQL AB yang dapat didapatkan secara gratis, dan juga memiliki kemampuan yang dapat digunakan untuk perusahaan sekelas enterprise, dan tiap versi terbarunya semakin mendekati standar SQL – 92, sehingga mudah digunakan (Gilmore, 2006).

II.1.4.1 Fleksibilitas

Pada saat ini terdapat 12 buah versi biner MySQL untuk berbagai macam platform, termasuk DEC OSF, FreeBSD, IBM AIX, HP-UX, Linux, Mac OS X, Novell Netware, OpenBSD, QNX, SGI IRIX, Solaris, dan Microsoft windows. Dan MySQL juga menyediakan source codenya tersedia untuk didownload jika

file binernya tak tersedia untuk platform yang pengguna inginkan, atau jika pengguna ingin mengkompilasi sendiri.

MySQL juga memiliki API yang banyak tersedia untuk digunakan oleh berbagai macam bahasa pemograman, diantaranya C, C++, PERL, Ruby, PHP, dan Java.

MySQL juga memiliki banyak cara untuk menangani data yang disimpan didalamnya (storage engine). Alasannya ada banyak cara tersebut dikarenakan adanya kebutuhan untuk algoritma penyimpanan/penanganan terhadap kasus – kasus tertentu. Beberapa storage engine yang dipakai MySQL adalah MyISAM, BDB, HEAP (yang berubah menjadi MEMORY pada versi 5), InnoDB, ISAM, dan MERGE. Dikarenakan pada satu database dapat berisi lebih dari satu table, dimana tabel – table tersebut memiliki fungsi tersendiri pula, maka MySQL memungkinkan untuk penggunaan storage engine yang berbeda antar table.

II.1.4.2 Kekuatan

MySQL memliki beberapa kekuatan yang membuatnya layak untuk diimplementasikan pada pembuatan aplikasi, diantaranya yaitu :

II.1.4.2.1 Fitur SQL untuk kelas Enterprise

Pada permulaannya MySQL dianggap tidak cocok untuk diterapkan pada level enterprise, dikarenakan tidak mempunyai fitur stored procedure, views, dan subquery. Namun pada MySQL versi 5 telah menjadi pembuktian bahwa MySQL dapat diadopsi ketingkat enterprise, dengan adanya fitur yang tidak ada sebelumnya.

II.1.4.2.2 Pengindeksan dan pencarian terhadap teks

Pada MySQL ver 3.23.23 telah ditambahkan kemampuan untuk pencarian dan pengindeksan terhadap kolom bertipe *string (CHAR, TINYTEXT, TEXT, MEDIUMTEXT, dan LONGTEXT).* Fitur ini meningkatkan performa ketika melakukan pencarian, dalam hal pencarian berdasarkan kesamaan dengan isi dari tiap baris.

II.1.4.2.3 Query caching

Query caching adalah salah satu cara MySQL untuk meningkatkan kecepatannya, dimana jika diaktifkan maka ketika terdapat terdapat perintah SELECT MySQL akan menyimpan query tersebut beserta hasilnya dimemory, maka apabila terdapat pemanggilan query yang kedua maka MySQL akan menggunakan query yang sebelumnya, untuk menghilangkan ketidak terbaruannya data MySQL menggunakan tehnik khusus untuk membuat ulang cache dan menghilangkan setelah request selesai.

II.1.4.2.4 Replikasi

Pada versi 3.2.1.15 MySQL telah menambahkan fitur replikasi. Replikasi memungkinkan database yang ditaruh dalam satu server MySQL untuk duplikasi ke yang lainnya, dimana dapat memberikan banyak keuntungan, sebagai contoh, dengan adanya database replikasi ditempat lain dapat meningkatkan ketersediannya, dimana jika database utama mengalami masalah maka dapat langsung menggantikannya. Dapat melakukan query dari client dapat dibagi ke database yang berbeda sehingga terjadi pembagian beban kedatabase lainnya. Keuntungan lainnya yaitu ketika terjadi backup data aplikasi menjadi offline dan menunggu proses backup selesai, pengguna dapat menjalankan backup pada database replikasinya, dimana hal ini memungkinkan waktu aplikasi ketika down mendekati 0.

II.1.4.3 Keamanan

MySQL memiliki banyak jenis tehnik pengamanan dan pilihan konfigurasinya. Yang memungkinkan penggunanya untuk mengontrol semua aspek yang berhubungan dengan cara kerja MySQL. Sebagai contoh dengan pilihan konfigurasi pengguna dapat melakukan hal – hal seperti :

(1). Pemilik daemon dapat menggunakan bahasa, port standar, lokasi penyimpanan data MySQL, dan pilihan global lainnya.
(2). Banyak memori yang digunakan pada tiap thread, cache dari query, table temporer, table joins, dan buffer dari index.

(3). Bermacam aspek dari kemampuan jaringan MySQL, termasuk lama waktu yang diijinkan untuk melakukan sebuah koneksi sebelum memutuskan koneksi tersebut, apakah untuk mencoba untuk mencari nama dari DNS, besarnya paket data yang diizinkan, dan lain sebagainya.

Pilihan konfigurasi keamanan yang dimiliki oleh MySQL juga sangatlah bagus, yang memungkinkan penggunanya untuk melakukan hal – hal seperti :

(1). Banyaknya jumlah query, update, dan koneksi yang diijinkan tiap jam
(2). Apakah seorang user harus menggunakan sertifikat SSL yang sah untuk terhubung ke suatu database.
(3). Tindakan yang dapat dilakukan seorang pengguna terhadap suatu database, table, bahkan kolom. Dimana seorang user mungkin dapat mengupdate sebuah tabel, namun tidak dapat menghapus data dalam tabel tersebut.

II.1.4.4 Pilihan lisensi yang fleksibel

MySQL ditawarkan kepada penggunanya dengan dua buah pilihan lisensi penggunaan yaitu :

II.1.4.4.1 MySQL Open Source Licence

MySQL AB menawarkan produknya dalam lisensi GNU Public licence (GPL), dimana para pemakainya dapat dengan bebas memakai, menggunakan, bahkan mendistribusikannya kembali. Namun berarti aplikasi yang digunakan apabila didistribusikan kembali harus menggunakan lisensi GPL.

II.1.4.4.2 Comercial Licence

Apabila pemakai MySQL ingin mendistribusikan aplikasinya tanpa kode pembuatnya (tanpa lisensi GPL) maka tersedia MySQL Commercial license dengan harga yang wajar.

II.1.5 CakePHP Framework

CakePHP framework yang gratis, opensource, rapid development framework untuk PHP, sebuah pondasi yang tersturktur untuk programmer untuk membuat aplikasi web. Tujuan utama CakePHP *framework* adalah agar dapat

membuat programmer bekerja secara terstruktur dan cepat tanpa kehilangan fleksibilitas.

II.1.5.1 Fitur fitur CakePHP

Beberapa fitur fitur yang ditawarkan oleh cakePHP adalah

(1). Web komunitas yang aktif dan bersahabat

(2). Lisensi yang fleksibel

(3). Kompatibel dengan PHP versi 4 dan PHP versi 5

(4). Interaksi database dengan *Create Read Update Delete (CRUD)* yang terintegrasi

(5). Aplikasi *Scaffolding*

(6). Penciptaan Kode

(7). Arsitektur MVC

(8). Request dispatcher dengan clean URL, costum URL dan routes

(9). Buid-in validation

(10). Template yang cepat dan fleksibel

(11). *View helpers* untuk AJAX, Javascript, HTML form dan banyak lagi

(12). Komponen untuk *email, cookie, security, session*, dan *request handling*

(13). *Access control List (ACL)* yang fleksibel

(14). Data Sanitization

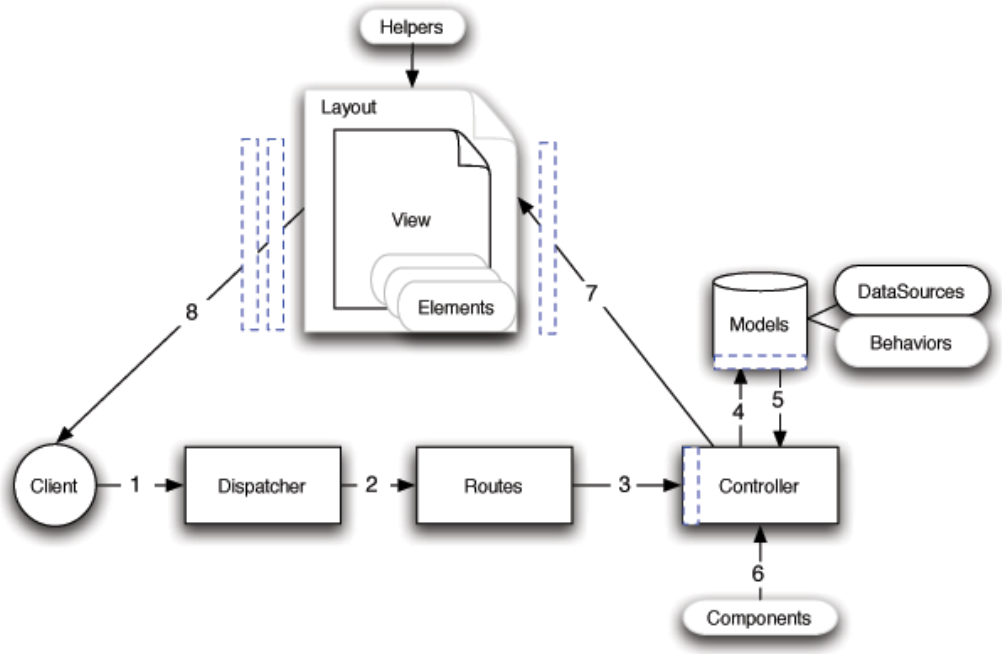
(15). *Caching* yang fleksibel

(16). Localization

(17). Dapat bekerja dari semua direktori website, dengan sedikit atau tanpa konfigurasi apache

II.1.5.2 Alur MVC pada CakePHP

Konsep MVC pada CakePHP digambarkan pada gambar dibawah



Gambar II .2 Alur MVC pada CakePHP
(sumber : http://book.cakephp.org)

Alur MVC pada CakePHP pada gambar diatas dijalankan dengan langkah langkah sebagai berikut:

(1). User mengklik sebuah link, browser user membuat sebuah *request* ke webserver.

(2). *Dispatcher* memparser URL link untuk mengekstrak parameter request dimana hasil yang diekstrak ialah controller, action, dan argument lain yang akan membawa efek kepada business logic selama request dijalankan.

(3). Dengan menggunakan routes, sebuah request URL ditujukan ke controller action (sebuah metode yang berada di controller yang telah dispesifikasikan). Controller before filter() callback dipanggil sebelum controller action logic dijalankan.

(4). Controller dapat menggunakan model untuk memperoleh akses ke aplikasi data. Apapun application model callback, behaviors, datasource dapat

digunakan pada operasi ini. Ketika penggunaan model tidak digunakan, semua controller CakePHP biasanya membutuhkan paling tidak satu buah model.

(5). Setelah model menerima data, data tersebut dikirim ke controller.

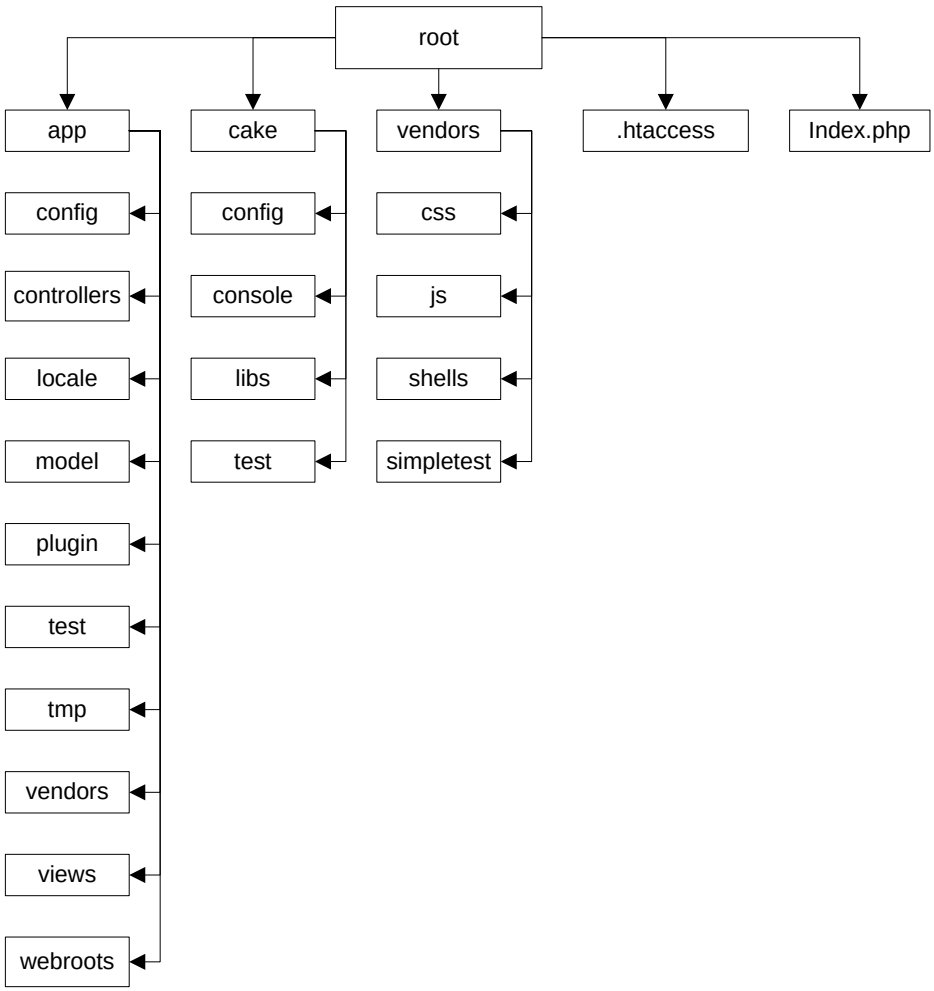
(6). Controller dapat menggunakan components untuk menyaring data atau melakukan operasi lain, contohnya : *manipulasi session, authentication*, atau mengirim email.

(7). Setelah controller selesai menggunakan model dan component untuk menyiapkan data, data tersebut dikirim ke view dengan menggunakan controller set method. Controller callback dapat digunakan pada saat data dikirim. View logic dilakukan, dimana didalamnya terdapat penggunaan element atau helpers. Biasanya view di render didalam layout.

(8). Hasil dari view yang direnderdikirim ke browser user.

II.1.5.2.1 Sturuktur Filesystem framework CakePHP

Struktur filesystem framework CakePHP digambarkan pada gambar



Gambar II .3 Struktur Filesystem Framework CakePHP

Pada framework CakePHP struktur filesystem secara global terbagi atas dua folder yaitu :

(1). Folder app, yaitu folder yang merupakan tempat ditaruhnya file – file yang membentuk CakePHP menjadi sebuah aplikasi web.

(2). Folder cake merupakan folder tempat file – file inti dari framework CakePHP

II.1.5.2.2 Contoh Penggunaan CakePHP

Cara penggunaan cakePHP dimulai dengan membuat file (disini akan menggunakan nama post.php) yang diletakkan dalam folder /app/models dengan isi file sebagai berikut :

```
<?php
      Class Post extends AppModel{
      Var $name = 'Post';     }
?>
```

Penggunaan naming convension sangat penting pada CakePHP. Dengan memberi nama model dengan nama Post, CakePHP akan secara otomatis merujuk bahwa model ini digunakan pada PostController, dan akan ditujukan pada tabel database yang bernama posts.

setelah itu membuat file controller dimana semua business logic untuk beriteraksi dengan post model akan terjadi (disini akan menggunakan nama post_controller.php) yang diletakkan dalam folder /app/controller

```
<?php
class PostsController extends AppController {
      var $name = 'Posts';
      function index() {
             $this->set('posts', $this->Post->find('all'));
      }
      function view($id = null) {
            $this->Post->id = $id;
            $this->set('post', $this->Post->read());  }}?>
```

setelah itu membuat file view dimana hasil yang didapat dari controller akan ditampilkan ke browser user (disini akan menggunakan nama view.ctp) yang diletakkan dalam folder /app/view/post/

```
<!-- File: /app/views/posts/view.ctp -->
<h1><?php echo $post['Post']['title']?></h1>
```

```
<p><small>Created:<?php echo$post['Post']['created']?></small></p>

<p><?php echo $post['Post']['body']?></p>
```

jika browser mengakses url http://localhost/post/ maka browser akan menampilkan seluruh data yang ada pada tabel posts.

II.2 Pengertian Pemesanan

Menurut Kamus Umum Bahasa Indonesia Badudu-Zain, definisi pemesanan adalah :

"Pemesanan adalah hal, cara, hasil, atau proses kerja memesan". Badudu-Zain [2].

II.3 Pengertian Produk

Menurut Kamus Umum Bahasa Indonesia Badudu-Zain definisi produk adalah :

"Produk adalah hasil; yang dihasilkan berupa barang baik konkret maupun abstrak". Badudu-Zain [2].

II.4 Pengertian Penjualan

Menurut Kamus Umum Bahasa Indonesia Badudu-Zain definisi penjualan adalah :

"Penjualan adalah hal, cara, atau hasil pekerjaan menjual". Badudu-Zain [2].

II.5 Perangkat Lunak

Pada bagian ini akan dijelaskan mengenai definisi Perangkat Lunak dan definisi Rekayasa Perangkat Lunak.

II.5.1 Pengertian Perangkat Lunak

Pada masa sekarang pengembangan Perangkat Lunak sangat berkembang pesat seiring juga dengan berkembangnya dunia internet. Seiring dengan dibutuhkannya sebuah Perangkat Lunak yang canggih dan dapat mengatasi semua masalah yang timbul, maka terkadang Perangkat Lunak lebih mahal dari

Perangkat Kerasnya sendiri. Ada beberapa deskripsi Perangkat Lunak (Software), yaitu :

(1). Instruksi–instruksi (program– program komputer) yang ketika dieksekusi memberikan fungsi dan kinerja yang diinginkan.
(2). Struktur–struktur data yang memungkinkan program untuk memanipulasi informasi.
(3). Dokumen–dokumen yang menjelaskan operasi dan penggunaan program.

Perangkat Lunak mempunyai karakteristik yang berbeda dari perangkat keras. Beberapa karakteristik tersebut yaitu :

(1). Perangkat Lunak dikembangkan / direkayasa, jadi tidak dimanufaktur dalam pengertian klasik.
(2). Perangkat Lunak tidak akan rusak, maksudnya adalah bersamaan dengan perkembangan waktu, Perangkat Lunak tidak memiliki kecepatan rusak atau kegagalan seperti pada perangkat keras.
(3). Pada umumnya Perangkat Lunak dibuat sendiri, bukan dibentuk dari komponen yang sudah ada.

Perangkat Lunak adalah informasi yang tersimpan dalam 2 bentuk komponen dasar , yaitu komponen yang tidak bisa dieksekusi oleh mesin (*non machine executable*) dan yang dapat dieksekusi oleh mesin (*executable machine*).

Komponen Perangkat Lunak dibuat melalui translasi– translasi berurutan, yang memetakan kebutuhan pemakai dengan kode mesin. Model kebutuhan (*prototype*) diubah jadi desain, lalu diubah jadi bentuk bahasa yang sebelumnya telah didefinisikan struktur datanya, prosedur, fungsi, dan lain- lainnya.

Suatu Perangkat Lunak dapat dikategorikan sebagai Perangkat Lunak yang baik jika memenuhi kriteria sebagai berikut :

(1). *Maintanability* (memenuhi perubahan)
(2). *Dependability* (handal , aman)
(3). *Efisiensi* (penggunaan resource yang efisien)

(4). *Usability* (user interface yang baik, program dapat berjalan dengan benar dan dokumentasi tersedia)

II.5.2 Pengertian Rekayasa Perangkat Lunak

Dari beberapa pengajuan definisi Perangkat Lunak, hampir semuanya menekankan pada aspek disiplin dalam pengembangan Perangkat Lunak. Definisi Awal dari Rekayasa Perangkat Lunak sendiri dikemukakan / diusulkan oleh Fritz Baeur, yaitu mengenai pembentukan dan penggunaan prinsip–prinsip engineering yang baik untuk mendapatkan Perangkat Lunak yang ekonomis, yang handal dan dapat bekerja secara efisien pada mesin nyata.

Rekayasa Perangkat Lunak mempunyai / terdiri dari 4 elemen, yaitu :

(1). *Tool* , merupakan alat Bantu dalam melaksakan proses Rekayasa Perangkat Lunak.
(2). Metode , merupakan teknik tentang bagaimana untuk membangun sebuah Perangkat Lunak.
(3). *Process* , merupakan kerangka kerja bagi key process area, yang harus dibentuk untuk mencapai efektifitas dari Rekayasa Perangkat Lunak.
(4). *Quality Flows* , merupakan kualitas produk sebagai tujuan akhir dari Rekayasa Perangkat Lunak.

II.6 Konsep Dasar Sistem

Terdapat dua kelompok pendekatan dalam mendefinisikan suatu sistem, yaitu pendekatan yang menekankan pada komponen atau elemen sistem. Menurut Jerry Fitz Gerald definisi sistem berdasarkan pendekatan yang lebih menekankan pada prosedur adalah :

> *“Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.”* JOG [4].

Pendekatan sistem yang merupakan jaringan kerja dari prosedur lebih menekankan urutan-urutan operasi di dalam sistem. Menurut Jerry Fitz Gerald, Ardra F. Fitz Gerald dan Warren D. Stallings,Jr., definisi prosedur adalah :

> *"Suatu prosedur adalah suatu urutan-urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan apa (what) yang harus dikerjakan, siapa (who) yang akan mengerjakannya, kapan (when) dikerjakan dan bagaimana (how) mengerjakannya."* JOG [4].

Menurut Jogiyanto. HM, definisi sistem berdasarkan pendekatan sistem yang lebih menekankan pada elemen atau komponennya adalah :

> *"Sistem merupakan kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu".* JOG [4].

Secara garis besar, sistem merupakan kumpulan komponen-komponen dan elemen-elemen yang saling berkaitan atau berhubungan dan saling berinteraksi membentuk suatu kesatuan secara keseluruhan demi mencapai suatu tujuan tertentu yang sudah dirancang sebelum sistem tersebut dibangun.

II.7 Karakteristik Sistem

Menurut Jogiyanto HM, suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu :

(1). Komponen – komponen (*Components*)

Komponen atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

(2). Batas sistem (*boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luar sistem dan memungkinkan suatu sistem dipandang sebagai suatu kesatuan, juga menunjukan ruang lingkup dari sistem tersebut.

(3). Lingkungan luar sistem (*environtments*)

Lingkungan luar sistem (*environtments*) dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem.

(4). Penghubung (*interface*)

Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem yang lain.

(5). Masukan (*input*)

Masukan (*input*) adalah energi yang dimasukkan ke dalam sistem, dapat berupa masukkan perawatan (*maintenance input*) dan masukkan sinyal (*signal input*).

(6). Keluaran (*output*)

Keluaran (*output*) adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna.

(7). Pengolah (*process*)

Pengolah (*process*) adalah suatu kegiatan sesuai dengan prosedur yang dimasukkan, untuk mengubah suatu masukkan menjadi keluaran yang bermanfaat.

(8). Sasaran (*objectives*) atau tujuan (*goals*)

Sasaran (*objectives*) atau tujuan (*goals*) adalah suatu keadaan yang diharapkan dan ingin dicapai dari suatu sistem.

II.8 Konsep Dasar Informasi

Suatu informasi dapat ditransformasikan pada saat informai itu mengalir dalam sebuah sistem baik itu manual maupun berbasis komputerisasi. Sistem tersebut menerima input dalam berbagai cara mengaplikasikan perngakat keras dan elemen manusia untuk mentransformasikan input menjadi output dalam berbagai bentuk.

Informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan berarti bagi penerimanya, yang

menggambarkan suatu kejadian nyata dan digunakan untuk pengambilan keputusan.

Menurut Jogiyanto HM definisi informasi adalah :

*"Informasi adalah data yang telah diolah menjadi suatu bentuk yang berarti bagi penerima dan bermanfaat dalam mengambil keputusan saat ini atau yang akan datang".*JOG [4].

Kualitas informasi ditentukan oleh beberapa faktor, yaitu

(1). Keakuratan dan teruji kebenaran informasi

Informasi harus bebas dari kesalahan-kesalahan, tidak bias, dan tidak meyesatkan.

(2). Kesempurnaan informasi

Untuk mendukung faktor pertama diatas, maka kesempurnaan informasi menjadi faktor penting, dimana informasi disajikan lengkap tanpa pengurangan, penambahan atau pengubahan.

(3). Tepat waktu

Informasi harus disajikan secara tepat waktu, mengingat informasi akan menjadi dasar dalam pengambilan keputusan. Keterlambatan informasi akan mengakibatkan kekeliruan dalam pengambilan keputusan.

(4). Relevansi

Informasi akan memiliki nilai manfaat yang tinggi jika informasi tersebut diterima oleh mereka yang membutuhkan, dan menjadi tidak berguna jika diberikan kepada mereka yang tidak membutuhkan.

(5). Mudah dan murah

Sulit atau tidaknya cara dan mahal atau tidaknya biaya untuk memperoleh informasi juga menjadi bahan pertimbangan. Jika cara dan biaya untuk memperoleh informasi sulit dan mahal, maka orang akan menjadi tidak berminat untuk memperolehnya, atau mencari alternatif substitusinya.

Nilai suatu informasi ditentukan oleh dua hal, yaitu manfaat dan biaya mendapatkan informasi. Suatu informasi dikatakan bernilai jika memiliki manfaat yang lebih efektif dibandingkan dengan biaya mendapatkan informasi tersebut, sebagian besar informasi tidak dapat diperkirakan dengan satuan nilai uang, tetapi dapat ditaksir nilai efektivitas informasi tersebut.

II.9 Konsep Dasar Sistem Informasi

Menurut Jogiyanto HM, definisi sistem informasi adalah :

"Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan yang diperlukan". JOG [4].

Suatu sistem informasi harus mampu mendukung para pengelola untuk menganalisis permasalahan, menyajikan kesimpulan dari hasil analisa melalui grafik-grafik dan tabel-tabel. Sistem informasi yang baik memiliki sistematika yang jelas, mulai dari tahap penginputan data, pengolahan data, dan penyajian output yang telah didapat kemudian mendistribusikan, menyimpan data dan informasi.

II.10 Tahap Perancangan Sistem Informasi

Tahap perancangan disebut juga tahap pemecahan masalah, yaitu dengan menyusun suatu algoritma, alur sistem, masukan, prosedur proses, keluaran, dan database. Proses perancangan diperlukan untuk menghasilkan suatu rancangan sistem yang baik, karena dengan rancangan yang tepat akan menghasilkan sistem yang stabil dan mudah dikembangkan di masa mendatang. Berikut ini akan dijelaskan rangkaian atau ruang lingkup sistem yang akan dirancang dengan memanfaatkan alat bantu seperti :

(1). *Unified Modeling Language*

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML tidak berdasarkan pada bahasa pemrograman tertentu. Standar

spesifikasi UML dijadikan standar defacto oleh OMG (Object Management Group) pada tahun 1997.UML yang berorientasikan object mempunyai beberapa notasi standar. Spesifikasi ini menjadi populer dan standar karena sebelum adanya UML, telah ada berbagai macam spesifikasi yang berbeda. Hal ini menyulitkan komunikasi antar pengembang perangkat lunak. Untuk itu beberapa pengembang spesifikasi yang sangat berpengaruh berkumpul untuk membuat standar baru. Dalam UML terdapat beberapa diagram yaitu :

a. *Use case diagram.*

Use Case dibentuk untuk mendeskripsikan apa yang sistem harus lakukan, bukan bagaimana cara sistem melakukannya. Setiap use case berisi minimal: nama yang mencerminkan gunanya dan skenario. Selain itu use case juga dapat berisi:

(1). ID unik guna memudahkan traceability

(2). Tujuan, yaitu kalimat sederhana yang menyatakan guna dari use case tersebut.

(3). Aktor, yaitu pihak-pihak yang bersangkutan untuk mengembangkan use case tersebut.

(4). prioritas, yaitu tingkat kepentingan dari use case

(5). resiko, yaitu tingkat resiko perubahan atau kegagalan saat pengembangan use case

(6). asumsi, yaitu asumsi-asumsi yang dipakai

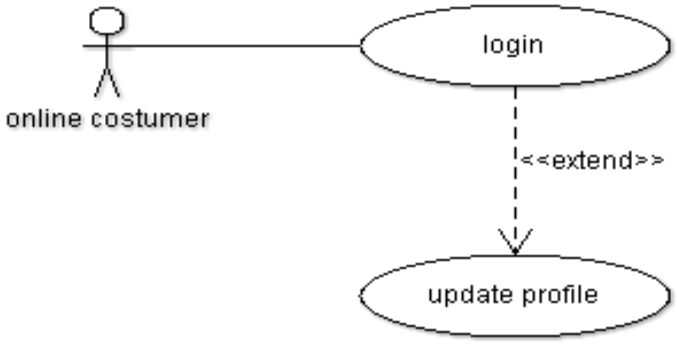
(7). preconditions, yaitu: kondisi-kondisi yang harus dipenuhi sebelum mengembangkan use case

(8). postconditions, yaitu: kondisi-kondisi yang harus dipenuhi sebelum use case diselesaikan

(9). outstanding issues, yaitu: daftar item-item yang harus ada sebelum use case dapat dianalisis dan didesain

(10). requirements satisfied, yaitu: hal-hal yang harus dipenuhi oleh usecase tersebut.

Use case sebetulnya hanyalah sebuah teknik untuk mendeskripsikan kebutuhan. Use case harus bisa dimengerti oleh kedua belah pihak, yaitu *stakeholders* atau *client* dan staff pengembang, oleh karena itu use case dibuat se-fleksibel mungkin. Sebuah use case dapat berisi beberapa skenario. Skenario utama, disebut juga happy path , adalah skenario dari hal yang umumnya terjadi. Sedangkan ada skenario lain yang mendeskripsikan jika terjadi hal-hal yang berbeda. Skenario diharapkan deskriptif, yaitu mendeskripsikan dialog antara aktor dan sistem. Hubungan antar use cases digambarkan dalam suatu use case diagram. Kumpulan lengkap antara use cases , aktor, dan diagram form disebut use case model. ini adalah sebuah contoh diagram usecase.

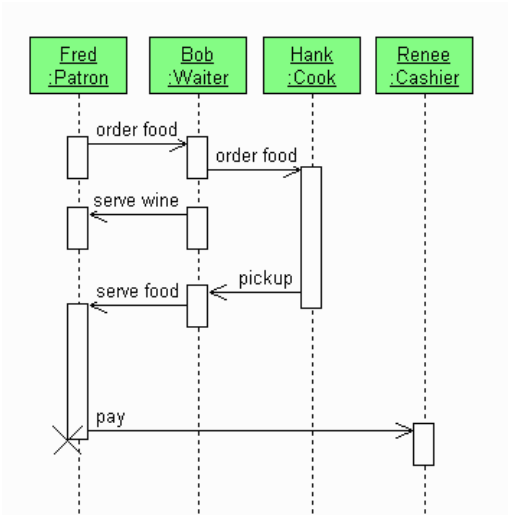


Gambar II .4 diagram usecase

Dalam diagram di atas, sang aktor, yang disebut online costumer, merepresentasikan peran yang dapat dimainkan oleh pengguna dalam sistem tersebut. Hal ini berarti aktor dapat memanggil use case ini dan berpartisipasi dalam skenario tersebut. Sebuah use case tambahan yang disebut update-profile terhubung dengan use case log-on. Di sebelah panah tersebut terdapat kata “<<extends>>”. Hal ini dapat dibaca: update profile extends log-on use case. Hal ini berarti aktor berpartisipasi dalam use case log-on dan dapat juga berpindah ke use case update profile.

b. Sequence diagram

Sequence diagram untuk use case adalah diagram interaksi yang mengekspresikan interaksi antara aktor dan sistem dengan penekanan waktu. Setiap skenario pada use case harus dibuat sequence diagramnya. Yang dideskripsikan adalah keinginan aktor, bukan mekanisme implementasi. Sequence diagram menunjukkan hubungan dinamis antara objek. Setiap sequence diagram hanya mendeskripsikan satu skenario. Contoh dari sequence diagram adalah sebagai berikut:



Gambar II .5 contoh sequence diagram

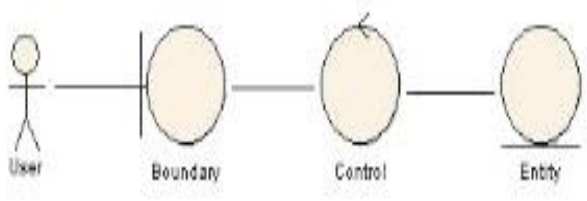
c. Use case analysis

Use case analysis dimulai dengan mengelaborasikan sequence diagram dari use case dasar dengan analysis-object model.Analysis objects adalah instansiasi dari analysis classes yang merepresentasikan hal-hal umum dari sistem yang memiliki sifat dan tugas. Nama-nama class pada tahap ini harus memiliki hubungan yang erat dengan domain. Ada tiga macam tipe analysis classes, yaitu:

(1). Boundary objects, merepresentasikan antar muka antara aktor dan sistem. Dalam sebuah aplikasi web, biasanya berbentuk halaman web.

(2). Entity objects, adalah sesuatu yang tetap ada setelah kejadian-kejadian pada use case. Contoh instansiasinya:customer,order, dll.

(3). Control objects, hal ini merepresentasikan proses.



Gambar II .6 usecase analisis

Hal untuk dicatat: dalam tahap pencarian kebutuhan, diagram yang harus ada hanya diagram usecase. Dua diagram terakhir adalah opsional.

(2). Kamus Data

Kamus data adalah peralatan yang ikut berperan dalam perancangan dan pembangunan sistem informasi karena berfungsi untuk menjelaskan arti aliran data dan penyimpanan dalam penggambaran pada data flow diagram, mendeskripsikan komposisi paket data yang bergerak melalui aliran, dan menjelaskan spesifikasi nilai dan satuan yang relevan terhadap data yang mengalir dalam sistem tersebut.

II.11 Metode Pengembangan Sistem

Metode pengembangan sistem adalah metode-metode, prosedur-prosedur, konsep-konsep pekerjaan, aturan-aturan yang akan digunakan sebagai pedoman bagaimana dan apa yang harus dikerjakan selama pengembangan ini. Metode adalah suatu cara atau teknik sistematis untuk mengerjakan sesuatu. Urutan prosedur untuk penyelesaian masalahnya dikenal dengan istilah algoritma. Metode pengembangan sistem yang akan digunakan dalam pembangunan sistem informasi pemesanan dan penjualan ini adalah pendekatan terstruktur.

Pendekatan terstruktur mengenalkan penggunaan alat-alat dan teknik-teknik untuk mengembangkan sistem yang terstruktur. Tujuan pendekatan terstruktur adalah agar pada akhir pengembangan perangkat lunak dapat

memenuhi kebutuhan user, dilakukan tepat waktu, tidak melampaui anggaran biaya, mudah dipergunakan, mudah dipahami dan mudah dirawat.

Menurut Roger S Pressman [6], dalam melakukan perancangan sistem yang menggunakan metode *prototype* dimulai dengan pengumpulan kebutuhan *user* dan mendefinisikan objektif keseluruhan dari perangkat lunak, mengidentifikasi gejala kebutuhan yang diketahui dan area garis besar dimana definisi lebih jauh merupakan keharusan kemudian dilakukan "perancangan kilat", perancangan kilat membawa kepada konstruksi sebuah *prototype*, *prototype* tersebut dievaluasi oleh *user* dan dipakai untuk menyaring kebutuhan pengembangan perangkat lunak.

II.11.1 Tiga bentuk model *Prototype*

(1). Diatas kertas berbasis komputer menggambarkan interaksi manusia.
(2). *Working prototype* : mengimplementasikan sebagian fungsi perangkat lunak.
(3). Program jadi : melakukan sebagian atau keseluruhan fungsi yang akan dilakukan, ada *feature* yang belum dikembangkan.

II.11.2 *Prototyping* (Tahapan)

(1). Pengumpulan kebutuhan dan perbaikan (Awal): analisa terhadap kebutuhan calon user.
(2). *Quick Design* (Desain Cepat): Pembuatan desain global untuk membentuk s/w contoh.
(3). Pembentukan Prototipe: Pembuatan s/w prototipe termasuk pengujian dan penyempurnaan.
(4). Evaluasi Pelanggan terhadap prototipe: mengevaluasi prototipe dan memperhalus analis kebutuhan calon pemakai.
(5). Perbaikan Prototipe: pembuatan sebenarnya termasuk *design, coding*, dan *testing*.
(6). Produk S/W (Akhir).

Secara ideal *prototype* berfungsi sebagai sebuah mekanisme untuk mendefinisikan kebutuhan perangkat lunak. *Prototype* bisa berfungsi sebagai "sistem yang pertama", hal ini merupakan pandangan yang ideal. Memang benar pelanggan maupun pengembang menyukai paradigma *prototype* para pemakai merasa enak dengan sistem aktual, sedangkan pengembangan bisa membangunnya dengan segera. *Prototype* bisa juga menjadi masalah karena alasan sebagai berikut:

(1). Pelanggan melihat apa yang tampak sebagai versi perangkat lunak yang bekerja tanpa melihat itu dijalin bersam-sama tanpa melihat bahwa di dalam permintaan untuk membuatnya bekerja, pengembangan belum mencantumkan kualitas perangkat lunak secara keseluruhan atau kemampuan pemeliharaan jangka panjang.
(2). Pengembang sering membuat kompromi-kompromi implementasi untuk membuat *prototype* bekerja secara tepat. Sistem operasi atau bahasa pemrograman yang tidak sesuai bisa dipakai secara sederhana karena mungkin diperoleh dan dikenal, algoritma yang tidak efisien secara sederhana dapat diimplementasikan untuk mendemonstrasikan kemampuan.

II.12 Analisis dan Perancangan Terstruktur

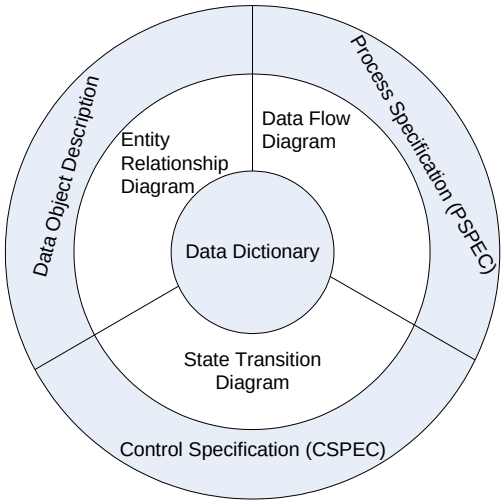
Analisis terstruktur merupakan metode pemodelan klasik yang telah digunakan secara luas sebagai sebuah notasi untuk definisi sistem, representasi, analisis persyaratan perangkat lunak dan desain sistem atau perangkat lunak.

II.12.1 Elemen Model Analisis

Menurut Roger S Pressman [6], elemen model analisis harus dapat mencapai tiga sasaran utama, yaitu :

(1). Untuk menggambarkan apa yang dibutuhkan oleh pelanggan,
(2). Untuk membangun dasar bagi pembuatan desain perangkat lunak , dan
(3). Untuk membatasi serangkaian persyaratan yang dapat divalidasi begitu perangkat lunak dibangun.

Untuk mencapai sasaran tersebut, model analisis yang ditarik selama analisis terstruktur berlangsung ditunjukkan pada gambar berikut.



Gambar II .7 Struktur Model Analisis
(Sumber: Pressman, Roger S, Rekayasa perangkat lunak pendekatan praktisi, McGraw-Hill Companies, Inc. 1997)

Pada inti model ada kamus data *(data dictionary)*, penyimpanan yang berisi deskripsi dari semua objek data yang dikonsumsi atau diproduksi oleh perangkat lunak. Ada tiga diagram yang mengelilingi inti. *Entity-relationship diagram (ERD*) mnegganbarkan hubungan antar objek data. ERD adalah notasi yang digunakan untuk melakukan aktifitas pemodelan data. Atribut dari masing-masing objek data yang ditulis pada ERD dapat digambarkan dengan menggunakan *deskripsi objek data*.

II.13 Pengujian Sistem

Menurut Roger S Pressman [6], mengungkapkan bahwa, Pengujian perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan mempresentasikan kajian pokok dari spesifikasi, desain, dan pengkodean.

Pengujian juga dapat diartikan sebagai sebuah proses eksekusi suatu program dengan maksud menemukan kesalahan.

II.13.1 Prinsip-prinsip pengujian

Sebelum mengaplikasikan metode untuk mendesain test case yang efektif, perekayasa perangkat lunak harus memahami prinsip dasar yang menuntun pengujian perangkat lunak.

(1). Semua pengujian harus dapat ditelusuri sampai ke persyaratan pelanggan.
(2). Pengujian itu harus direncanakan lama sebelum pengujian itu mulai.
(3). Untuk menjadi paling efektif, pengujian harus dilakukan oleh pihak ketiga yang indefenden.

II.13.2 Pengujian *Black-Box*

Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black-box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black-box* bukan merupakan alternative dari teknik *white-box*, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan dari pada metode *white-box*.

I.14 Konsep Dasar Basis Data

Menurut FAT [3], Basis data terdiri dari 2 kata, yaitu basis dan data. Basis dapat diartikan sebagai markas, gudang, tempat berkumpul atau tempat penyimpanan. Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, konep keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya. Basis data merupakan himpunan kelompok data yang saling berkaitan dan tempat untuk menampung dan mengorganisasikan seluruh data yang ada dalam sistem, sehingga dapat dieksplorasi untuk menyusun informasi – informasi dalam berbagai bentuk.

Prinsip utama basis data adalah pengaturan data atau arsip dengan tujuan utama untuk kemudahan dan kecepatan dalam pengambilan kembali data atau arsip tersebut. Dapat disimpulkan bahwa basis data memiliki kesamaan fungsi dan

prinsip kerja dengan lemari arsip, perbedaan antara basis data dengan lemari arsip hanya terletak pada media penyimpanan yang digunakan.

I.14.1 Sistem Hierarki Basis Data

Pembangunan sistem informasi sangat bertumpu pada kualitas basis data yang disusun dan dibentuk diharapkan memiliki sifat efektif dan efisien dalam pengorganisasiannya, bebas redudansi, fleksibel, dan sistem database yang dapat diakses secara bersamaan dalam lingkungan jaringan. Data dalam basis data disusun berdasarkan sistem hierarki SUT [7], yaitu :

(1). Database, merupakan kumpulan file yang saling terkait satu sama lain.
(2). File, yaitu kumpulan dari record yang saling berkaitan dan memiliki format field yang sama dan sejenis.
(3). Record, yaitu kumpulan field yang menggambarkan suatu unit data atau individu tertentu.
(4). Field, yaitu atribut dari record yang menunjukan suatu item dari data seperti nama, alamat, dan sebagainya.
(5). Byte, adalah atribut dari field yang berupa huruf yang membentuk nilai dari sebuah field.
(6). Bit, yaitu bagian terkecil dari data secara keseluruhan berupa karakter ASCII nol atau satu yang merupakan komponen pembentuk byte.

I.14.2 Perancangan Basis Data

Perancangan basis data merupakan salah satu bagian terpenting dalam proses pengembangan sistem. Perancangan basis data mempunyai beberapa tujuan yaitu :

(1). Menghilangkan redundansi data.
(2). Meminimumkan jumlah relasi di dalam basis data.
(3). Membuat relasi berada dalam bentuk normal, sehingga dapat meminimumkan permasalahan berkenaan dengan penambahan, pembaharuan dan penghapusan.

Basis data adalah satu kelompok organisasi data yang terpusat. Basis data secara umum dianalogikan sebagai lemari dokumen atau sekumpulan lemari dokumen. Pada tahap ini digunakan beberapa peralatan untuk mendukung proses pembentukan database tersebut, peralatan–peralatan yang digunakan untuk mendukung pembentukan basis data adalah :

(1). Normalisasi

Menurut SUT [7], normalisasi adalah peralatan yang dipergunakan untuk melakukan proses pengelompokan data yang menjadi menjadi tabel – tabel yang menunjukan entitas dan relasinya. Normalisasi merupakan cara pendekatan lain yang dalam membangun desain lojik basis data relasional dengan menerapkan sejumlah aturan dan kriteria standar untuk menghasilkan struktur tabel yang normal. Secara umum proses normalisasi terdiri dari dalam tahap, yaitu :

a. Tahap tidak normal.

   Pada tahap ini, semua data yang ada direkam tanpa format tertentu. Hal tersebut dapat menyebabkan data mengalami duplikasi.

b. Normalisasi tahap 1.

   Normalisasi tahap 1 menghilangkan duplikasi data yang terjadi pada tahap tidak normal dengan cara menghapus data-data yang sama.

c. Normalisasi tahap 2.

   Tahap normalisasi 3 adalah menentukan kunci dari normalisasi 1 yang akan digunakan sebagai *primary key* pada tabel, membentuk tabel berdasarkan *primary key* dan mengelompokkan data pada tabel-tabel yang sudah dibentuk

d. Normalisasi tahap 3.

   Pada tahap 3 dilakukan penentuan relasi antar tabel sehingga memungkinkan adanya field kunci sekunder.

(2). Skema Relasi

Menurut Budi Sutedjo, Skema Relasi adalah peralatan yang berfungsi untuk menggambarkan relasi atau hubungan dari dua file atau dua tabel yang dapat digolongkan dalam tiga macam bentuk relasi, yaitu satu – satu (1 – 1), satu – banyak (1 – N), banyak – banyak (N – N). Model entity relationship yang berisi komponen – komponen himpunan entitas dan himpunan relasi yang masing–masing dilengkapi dengan atribut–atribut yang merepresentasikan seluruh fakta yang kita tinjau digambarkan lebih sistematis dengan menggunakan Diagram E–R. Pembuatan Diagram E–R menurut SUT [7] meliputi tahap – tahap berikut:

a. Mengidentifikasi dan menetapkan seluruh himpunan entitas yang akan terlihat.
b. Menentukan atribut – atribut *key* dari masing – masing himpunan entitas.
c. Mengidentifikasi dan menetapkan seluruh himpunan relasi di antara himpunan entitas–himpunan entitas yang ada beserta *foreign key* yang terdapat pada relasi tersebut.
d. Menentukan derajat atau kardinalitas relasi untuk setiap himpunan relasi.
e. Melengkapi himpunan entitas dan himpunan relasi dengan atribut – atribut deskriptif (non *key*).

(3). Tabel Relasi

Tabel relasi adalah suatu perangkat yang digunakan untuk menggambarkan representasi struktur dan data dari hubungan antar tabel secara fisik atau nyata yang terjadi pada sistem informasi.

(4). Struktur File

Perangkat yang digunakan untuk memaparkan identitas setiap file mulai dari nama file, jenis, fungsi, dan isi dari file tersebut dalam bentuk tabel kamus

data yaitu tabel yang berisi nama field, type field tersebut, ukuran field, dan keterangan dari field.

I.14.3 Konsep Sistem Basis Data

Basis data adalah sebuah objek pasif atau mati yang ada karena ada yang membuat dan hanya berguna jika ada program aplikasi yang mengelola atau menggerakkannya. Program aplikasi dan basis data yang tergabung akan menghasilkan sebuah sistem. Secara umum Sistem Basis Data merupakan sistem yang terdiri atas kumpulan file (tabel) yang saling berhubungan dalam sebuah basis data di sebuah sistem komputer dan sekumpulan program DBMS (*database management system*) yang memungkinkan beberapa pemakai atau program lain untuk mengakses dan memanipulasi file – file tersebut.

Secara lengkap, dalam sebuah sistem basis data akan terdapat komponen – komponen utama sebagai berikut (FAT [3]) :

(1). Perangkat Keras (*hardware*)
(2). Sistem Operasi (*Operating System*)
(3). Basis Data (*Database*)
(4). Sistem (Aplikasi/Perangkat lunak) Pengelola Basis Data (DBMS)
(5). Pemakai (*user*)
(6). Aplikasi (Perangkat Lunak) lain (bersifat opsional)

I.14.4 Tujuan Basis Data

Basis Data (Database) pada prinsipnya ditujukan untuk pengaturan data

agar terdapat kemudahan dalam pengambilan kembali data tersebut. Berikut ini terdapat beberapa tujuan dari Basis Data diantaranya yaitu :

(1). Kecepatan dan kemudahan (Speed)
(2). Efisiensi ruang penyimpanan (Space)
(3). Keakuratan (Accuracy)
(4). Ketersediaan (Avaibility)
(5). Kelengkapan (Completeness)
(6). Keamanan (Security)

(7). Kebersamaan (Sharability)

I.14.5 Pengguna Basis Data

Pengguna basis data yang dibedakan berdasarkan cara mereka berinteraksi terhadap sistem diantaranya yaitu :

(1). Programmer Aplikasi (Application Programmer)Pemakai yang berinteraksi dengan basis data dengan menggunakan DataManipulation Language (DML) untuk membuat aplikasi dengan menggunakan bahasa pemrograman.

(2). User Mahir (Casual User)

Pemakai yang berinteraksi dengan sistem tanpa membuat modul program. User Umum (Native User) Pemakai berinteraksi dengan aplikasi basis data yang telah dibuat atau disediakan oleh sistem.

(3). User Khusus (Specialized User)

Pemakai yang membuat program aplikasi basis data untuk keperluan tertentu atau khusus.

I.14.6 Database Management System (DBMS)

Diperlukan suatu sistem untuk diintegrasikan data file ke dalam suatu file sehingga bisa melayani berbagai user yang berbeda. Perangkat keras dan lunak serta prosedur yang mengelola database manajemen sistem.

DBMS memungkinkan untuk membentuk dan meremajakan file-file, memilih, mendatakan dan menyortir data, dan untuk menghasilkan laporanlaporan.

Fungsi yang penting dari DBMS adalah sebagai berikut :

(1). Menyediakan sistem akses cepat.

(2). Mengurangi kerangkapan data atau redundancy data.

(3). Memungkinkan adanya updating secara bersamaan.

(4). Menyediakan sistem yang memungkinkan dilakukannya pengembangan database.

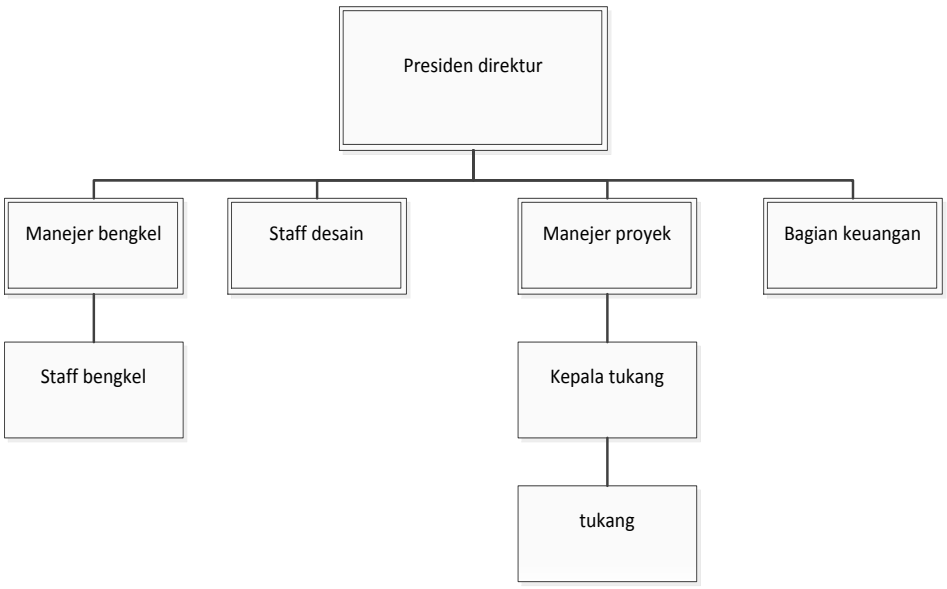
(5). Memberikan perlindungan dari pihak pemakai tidak berhak.

I.15 Sejarah singkat CC. Cipta Rupa Pratama

Cipta Rupa Pratama merupakan biro konsultan arsitektur dan interior, yang berlokasi di Jl.Awiligar Raya No.55 RT.03 RW.10 Kel.Cibeunying, Kecamatan Cimenyan dengan surat ijin usaha perdagangan (SIUP) Nomor : 1178/10-12/PK/XII/2008 oleh notaris Ferry Indra Buchari, SH, S.T,M.Kn. dengan surat keterangan domisili Nomor : 500/151/-pemb, Surat Ijin Tempat Usaha Nomor: 5 17/91 19/SITU/BPMP dan Nomor Pokok Wajib Pajak : 02.735.592.4-444.000, yang mengintegrasikan totalitas servis, yakni dengan mengharmonisasikan sistem metode desain dan pengembangan teknologi produksi yang optimal dalam mengaplikasikan keinginan klien untuk mewujudkan citra yang bernilai tinggi.Keberhasilan dalam mentransformasikan keinginan klien kedalam sebuah konsep desain yang kuat serta berkarakter dan menghadirkamiya sebagai produk interior yang sempurna sehingga tercapai sebuah kepuasan, merupakan tujuan utama kami dalam berkarya. Kami merupakan pilihan solusi yang tepat bagi kebutuhan anda akan sebuah proses desain dan hasil produksi yang inovatif, kreatif dan ekslusif.

I.16 Struktur Organisasi CC.Cipta Rupa Pratama

Untuk dapat bekerja lebih efektif, suatu perusahaan perlu memiliki pemahaman yang jelas tentang struktur organisasi. Dengan adanya struktur organisasi yang baik maka akan jelas fungsi, tugas, wewenang, dan tanggung jawab. Struktur organisasi CC.Cipta Rupa Pratama sebagai berikut:



Gambar II .8 Struktur organisasi

I.17 Uraian kerja masing — masing Bagian di CC.Cipta Rupa Pratama

(1). Presiden direktur

Bertugas memimpin, bertanggung jawab dan menetapkan kebijakankebijakan perusahaan dengan mengkoordinasikan, mengarahkan dan mengenda1ikan kegiatan-kegiatan perusahaan. Selain itu Director bertugas dalam proses memasukkan tender, bertemu owner/client dan mengontrol keuangan perusahaan.

(2). Staff desain

Bertugas menerjemahkan keinginan client yang disampaikan oleh director kedalam bentuk design arsitektur atau design interior.

(3). Manejer proyek

Manejer proyek yang bertugas memimpin dan bertanggung jawab terhadap jalannya proyek perusahaan secara keseluruhan pada lokasi proyek.

(4). Manejer Bengkel

Menyusun, merumuskan, dan pelaksana desain interior dan pengolahan bahan baku menjadi barang jadi dengan menyusun, mengatur dan mengendalikan produksi mulai dari rencana dan pelaksanaan produksi untuk mencapai target produksi yang telah ditetapkan oleh Design staff.

(5). Bagian keuangan

Melaksanakan dan mengendalikan pengelolaan keuangan dan akutansi sesuai dengan sistem prosedur dan peraturan perundang-undangan yang berlaku.

(6). Kepala Tukang

Mengatur, mengawasi dan juga bertindak sebagai pelaksana kegiatan proyek dilokasi proyek

(7). Workshop Staff

Pelaksana pengolahan bahan baku menjadi barang produksi

(8). Craftman (Tukang)

Pelaksana kegiatan proyek dilokasi project

II.18 Produk CC.Cipta Rupa Pratama

Produk yang dihasilkan cukup banyak, dan semua produk yang ada termasuk produk design interior unggulan dengan berbagai macam design, diantaranya:

(1). Grand Mahkota Hotel di kota Pontianak.
(2). Rumah Sakit Hasan Sadikin (bagian Penyakit dalam) dikota Bandung.
(3). Rumah Sakit Hasan Sadikin (Kamar Jaga PD) dikota Bandung.
(4). Hotel Papandayan dikota Bandung.
(5). Paradise House Dikota Yogyakarta.
(6). Provider Shop — BIP Mall Dikota Bandung.
(7). Fiesta Shop — BJPM all Dikota Bandung.
(8). Jatipadang House Dikota Jakarta.
(9). Sembara Restaurant Dikota Bandung.
(10). Housing Dikota Bandung dan Makassar.
(11). Exhibition Stand.