

BAB II

LANDASAN TEORI

Dalam bab ini akan dijelaskan mengenai definisi dan teori yang berkaitan dengan permasalahan yang ada. Berikut adalah teori–teori yang mendasari dari perancangan sistem informasi pengolahan data pemesanan dan produksi *furniture* pada CV. Annisa Prima Karya.

2.1 Sistem

Suatu sistem sangatlah dibutuhkan dalam suatu perusahaan atau instansi pemerintahan, karena sistem sangatlah menunjang terhadap kinerja perusahaan atau instansi pemerintah, baik yang berskala kecil maupun besar. Ada berbagai pendapat yang mendefinisikan pengertian sistem. Salah satunya menurut Jogiyanto (2005 : 1) sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu.

2.1.1. Konsep Dasar Sistem

Terdapat dua kelompok pendekatan di dalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya. Suatu sistem yang lebih menekankan pada prosedurnya menurut Jogiyanto (2005 : 1) didefinisikan suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu. Sedangkan definisi dari pendekatan sistem yang lebih menekankan pada komponen atau

elemennya menurut Hariyanto (2004 : 53) didefinisikan sebagai kumpulan objek atau elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

Dari definisi diatas, maka dapat disimpulkan bahwa sistem adalah sekumpulan komponen yang saling berintegrasi untuk dapat mencapai suatu tujuan dengan menerima input dan menghasilkan output dalam sebuah proses yang terorganisasi.

2.1.2. Karakteristik Sistem

Menurut Jogiyanto (2005 : 3) suatu sistem mempunyai karakteristik atau ciri-ciri tertentu yaitu sebagai berikut :

a. **Komponen Sistem (*component*)**

Pada suatu sistem terdiri dari beberapa atau sejumlah komponen yang saling berhubungan, atau berinteraksi antara komponen yang satu dengan komponen yang lainnya, atau bekerjasama untuk membentuk satu kesatuan dan dalam komponen sistem ini atau elemen-elemen sistem ini dapat berupa subsistem yang lebih kecil.

b. **Batasan Sistem (*boundary*)**

Batasan sistem merupakan garis besar abstraksi yang memisahkan antara sistem dan lingkungannya, batas sistem ini juga menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

c. **Lingkungan Luar Sistem (*environment*)**

Lingkungan luar dari suatu sistem adalah apapun yang terdapat diluar batas dari sistem yang mempengaruhi operasi sistem, lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga merugikan sistem tersebut,

lingkungan luar sistem yang menguntungkan merupakan energi dari sistem dan demikian harus tetap dijaga dan dipelihara.

d. Penghubung Sistem (*interface*)

Penghubung sistem merupakan suatu media antara satu sub sistem dengan yang lainnya, melalui penghubung ini memungkinkan sumber-sumber mengalir dari satu subsistem ke subsistem yang lainnya.

e. Masukan Sistem (*input*)

Masukan atau input merupakan energi yang dimasukkan kedalam sistem, masukan ini dapat berupa energi manusia, data model, bahan baku, layanan atau lainnya, input atau masukan dapat diklasifikasikan menjadi 3 kategori yaitu, *serial input*, *probable input*, dan *feedback input*.

f. Keluaran Sistem (*output*)

Keluaran sistem merupakan hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna, pembuangan keluaran sistem dapat merupakan masukan untuk susbsistem yang lain atau menuju kepada suatu sistem, definisi lain ialah *output* merupakan hasil dari porses yang merupakan tujuan dari keberadaan sistem.

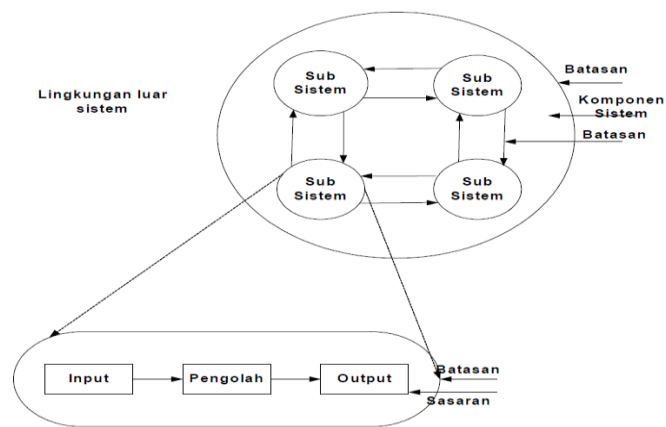
g. Pengolah Sistem (*proses*)

Pengolahan atau proses merupakan perubahan dari masukan atau input menjadi keluaran proses ini mungkin dilakukan oleh mesin, orang atau komputer.

h. Sasaran Sistem (*objective*)

Sasaran dari suatu sistem sangat menentukan sekali masukan yang dibutuhkan sistem serta keluaran yang akan dihasilkan suatu sistem. Serta dikatakan berhasil jika mengenai sasaran atau tujuannya, karena tujuan ini merupakan target atau sasaran akhir yang akan dicapai oleh suatu sistem.

Karakteristik suatu sistem dapat dilihat pada gambar 2.1 dibawah ini :



Gambar 2.1 Karakteristik Sistem

Sumber : Jogiyanto (2005 : 6) Analisis dan Desain Sistem Informasi Pendekatan Tersruktur

2.1.3. Klasifikasi Sistem

Menurut Al-Bahra bin Ladjamudin (2005 : 6) sistem dapat dikelompokkan atau diklasifikasikan menjadi beberapa sudut pandang diantaranya adalah sebagai berikut :

- Sistem abstrak (*abstract system*) dan sistem fisik (*physical system*), sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik.
- Sistem alamiah (*natural system*) dan sistem buatan manusia (*human made system*). Sistem alamiah adalah sistem yang terjadi melalui proses alam dan

tidak dibuat oleh manusia. Sistem buatan manusia adalah sistem yang dirancang oleh manusia dan melibatkan interaksi antara manusia dengan mesin.

- c. Sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*), sistem tertentu adalah sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi, interaksi diantara bagian-bagian dapat dideteksi dengan pasti sehingga keluaran/*output* dapat diramalkan. Sistem tak tentu adalah sistem yang kondisi masa depan atau yang akan datang tidak dapat diprediksi karena mengandung unsur probabilitas.
- d. Sistem tertutup (*closed system*) dan sistem terbuka (*open system*), sistem tertutup adalah sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luar. Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luar.

2.2 Konsep Dasar Data dan Informasi

Data adalah deskripsi tentang benda, kejadian, aktifitas, dan transaksi yang tidak mempunyai makna atau tidak berpengaruh secara langsung kepada pemakai. Dalam menganalisis dan merencanakan perancangan suatu sistem harus mengerti terlebih dahulu komponen-komponen yang ada dalam sistem tersebut. Darimana data dan informasi tersebut diperoleh dan kemana hasil pengolahan data dan informasi tersebut diperlukan.

Informasi bisa diibaratkan sebagai darah yang mengalir dalam tubuh suatu organisasi, sehingga informasi ini sangat penting. Jogyanto (2005 : 8)

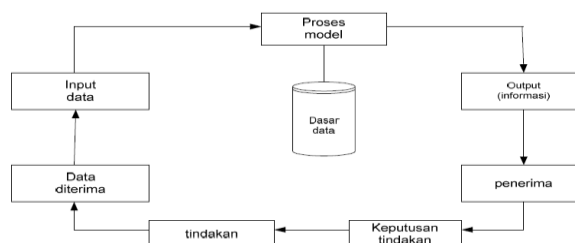
mendefinisikan Informasi sebagai data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Menurut Gordon. B. Davis dalam buku Al-bahra bin Ladjamudin mendefinisikan.

Informasi sebagai data yang telah diolah menjadi bentuk yang lebih berarti dan berguna bagi penerimanya untuk mengambil keputusan masa kini maupun masa yang akan datang. Informasi mempunyai ciri benar atau salah, baru, tambahan dan kolektif. Al-bahra (2005 : 8)

Sedangkan menurut Jogiyanto (2005 : 9) data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Berdasarkan pengertian-pengertian tersebut, dapat disimpulkan bahwa informasi adalah sekumpulan atau semua data yang telah diproses dan memiliki arti untuk dimengerti dan digunakan oleh pengguna informasi tertentu.

2.2.1 Siklus Informasi

Informasi merupakan suatu proses perubahan dunia menjadi informasi, data yang diolah menjadi suatu model menjadi informasi penerima, kemudian menerima informasi tersebut membuat suatu keputusan dan melakukan tindakan yang berarti menghasilkan suatu keputusan dan melakukan tindakan yang lain yang akan membuat sejumlah data kembali, untuk lebih jelasnya dapat dilihat pada gambar 2.2 siklus informasi berikut ini :



Gambar 2.2 Siklus Informasi

Sumber : Jogiyanto (2005 : 9) Analisis dan Desain Sistem Informasi Pendekatan Tersruktur.

Kegunaan informasi adalah untuk mengurangi ketidakpastian didalam proses pengambilan keputusan tentang suatu keadaan. Informasi yang digunakan di dalam suatu sistem informasi umumnya digunakan untuk beberapa kegunaan.

2.2.2 Kualitas Informasi

Menurut Jogiyanto (2005 : 10) kualitas dari sistem informasi tergantung dari 6 (enam) hal, antara lain :

a. Akurat (*accurate*)

Akurat berarti informasi harus bebas dari kesalahan yang tidak bisa atau menyesatkan.

b. Tepat Pada Waktunya (*Timeliness*)

Tepat pada waktunya berarti informasi yang datang pada penerima tidak boleh terlambat.

c. Relevan (*Relevance*)

Relevan berarti informasi tersebut mempunyai manfaat untuk pemakainya.

d. Ekonomis (*economy*)

Ekonomis berarti informasi harus mempunyai daya jual yang tinggi, biaya untuk menghasilkan informasi tersebut minimal.

e. Efisien (*efficiency*)

Efisien berarti informasi yang berkualitas memiliki sintak atau kalimat yang sederhana, namun mampu memberikan makna dan hasil yang mendalam.

f. Dapat dipercaya (*reliability*)

Dapat dipercaya berarti informasi tersebut berasal dari sumber yang dapat dipercaya.

2.2.3 Nilai Informasi

Jogiyanto (2005 : 11) mengemukakan nilai dari informasi ditentukan dari dua hal yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya sebagian besar informasi tidak dapat ditaksir keuntungannya dengan suatu nilai uang, tetapi dapat ditaksir nilai efektifitasnya.

2.3 Konsep Dasar Sistem Informasi

Sistem informasi adalah sekumpulan dari bagian/komponen baik fisik ataupun nonfisik yang saling berhubungan dan bekerja sama secara harmonis untuk mencapai satu tujuan yaitu mengolah data menjadi informasi.

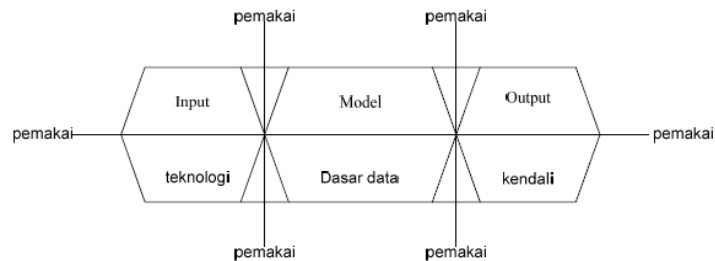
Dalam buku karangan Jogiyanto sistem informasi didefinisikan oleh Robert dan K. Roscoe adalah sebagai berikut.

Sistem informasi adalah suatu sistem didalam 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-laporan yang diperlukan. Jogiyanto (2005 : 11)

2.3.1 Komponen Sistem Informasi

Menurut John Burch dan Gary Grudnitski yang dikutip dari buku karangan Jogiyanto mengemukakan bahwa sistem informasi terdiri dari komponen-komponen yang disebutnya dengan istilah blok bangunan (*building block*), yaitu blok masukan (*input blok*), blok model (*model blok*), blok keluaran (*output blok*), blok teknologi (*technology blok*), blok basis data (*database blok*) dan blok kendali (*controls blok*). Sebagai suatu sistem keenam blok tersebut masing-

masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarnya.



Gambar 2.3 Blok Sistem Informasi yang Berinteraksi
Sumber : Jogiyanto (2005 : 12) Analisis dan Desain Sistem Informasi Pendekatan Tersruktur

a. Blok Masukan

Input mewakili datang masuk ke dalam sistem informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukan, yang dapat berupa dokumen-dokumen dasar.

b. Blok Model

Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara tertentu untuk menghasilkan keluaran yang diinginkan.

c. Blok Keluaran

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

d. Blok Teknologi

Teknologi merupakan “kotak alat” (*tool box*) dalam sistem informasi.

e. Blok Basis Data

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

f. Blok Kendali

Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan dapat langsung cepat diatasi.

2.3.2 Kegiatan Sistem Informasi

Dalam buku Al-Bahra bin Ladjamudin (2005 : 22) kegiatan-kegiatan yang dilakukan dalam sistem informasi, yaitu :

- a. *Input*, menggambarkan suatu kegiatan untuk menyediakan data untuk diproses.
- b. Proses, menggambarkan bagaimana suatu data diproses untuk menghasilkan suatu informasi yang bernilai tambah.
- c. *Output*, suatu kegiatan untuk menghasilkan laporan dari proses diatas tersebut.
- d. Penyimpanan, suatu kegiatan untuk memelihara dan menyimpan data.
- e. Kontrol, suatu aktivitas untuk menjamin bahwa sistem informasi tersebut berjalan sesuai dengan yang diharapkan.

2.4 Sekilas Tentang *Software* Yang digunakan

Berikut ini adalah sekilas tentang *software* yang digunakan dalam perancangan sistem informasi pemesanan dan produksi furniture pada CV. Annisa Prima Karya

2.4.1 PHP (*Hypertext Preprocessor*)

PHP merupakan singkatan dari PHP *Hypertext Preprocessor* yang digunakan sebagai bahasa *script server-side* dalam pengembangan *web* yang disisipkan pada dokumen HTML. PHP merupakan *software open source* yang disebar dan dilisensikan secara gratis serta dapat didownload secara bebas dari situs resminya <http://www.php.net>.

Kelebihan dari PHP, yaitu :

- 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 *apache, IIS, Lightpd, nginx*, hingga *Xitami* dengan konfigurasi lebih mudah.
- c. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan *developer* yang siap membantu pengembangan.
- d. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
- e. PHP adalah bahasa *open source* yang dapat digunakan di beberapa mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah system.

2.4.2 MySQL

Menurut David M. Kroenke (2005 : 220) MySQL adalah produk DBMS open source yang berjalan pada UNIX, Linux, dan Windows. Sumber dan kode biner MySQL dapat didownload dari situs Web MySQL (<http://www.mysql.com>). Keterbatasan MySQL tidak mendukung *View*, prosedur tersimpan, maupun *trigger*. Akan tetapi, semua hal tersebut ada pada *to-do-list* MySQL, sehingga periksa dokumentasi terakhir untuk menentukan apakah beberapa fitur-fitur tersebut telah ditambahkan ke produk tersebut pada *realese-realese* yang terbaru.

2.4.3 PHP Storm 2.0

PhpStorm adalah PHP IDE ringan dan cerdas berfokus pada produktivitas pengembang yang sangat memahami kode Anda, menyediakan penyelesaian kode cerdas, navigasi cepat dan memeriksa *on-the-fly* kesalahan. Itu selalu siap untuk membantu Anda membentuk kode Anda, menjalankan unit-tes atau memberikan debugging visual. PhpStorm mencakup semua fungsi WebStorm (HTML / CSS Editor, Javascript Editor) dan menambahkan dukungan penuh untuk PHP. <http://www.jetbrains.com/phpstorm/>

2.4.4. CSS (*Cascading Style Sheet*)

Cascading Style Sheets (CSS) adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa *markup*. Penggunaan yang paling umum dari CSS adalah untuk memformat halaman *web* yang ditulis dengan HTML dan XHTML. Walaupun demikian, bahasanya sendiri dapat dipergunakan untuk semua jenis dokumen XML termasuk SVG dan XUL. Spesifikasi CSS diatur oleh *World Wide Web Consortium* (W3C).

2.4.5. JavaScript

Javascript diperkenalkan pertama kali oleh Netscape pada tahun 1995. Pada awalnya bahasa ini dinamakan “*LiveScript*” yang berfungsi sebagai bahasa sederhana untuk *browser* Netscape Navigator 2. Javascript adalah bahasa yang berbentuk kumpulan skrip yang pada fungsinya berjalan pada suatu dokumen HTML, sepanjang sejarah internet bahasa ini adalah bahasa skrip pertama untuk web. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap bahasa HTML dengan mengizinkan pengeksekusi perintah di sisi user, yang artinya di sisi *browser* bukan di sisi *server* web. Javascript bergantung kepada *browser (navigator)* yang memanggil halaman web yang berisi skrip-skrip dari Javascript dan tentu saja terselip di dalam dokumen HTML.

2.4.6. JQuery

JQuery merupakan suatu *framework* (library) Javascript yang menekankan bagaimana interaksi antara Javascript dan HTML. *JQuery* pertama kali dirilis pada tahun 2006 oleh John Resig. Fitur utama dari *JQuery* diantaranya :

- a. Dapat mengakses elemen dalam dokumen

Javascript khusus, untuk mengakses suatu bagian tertentu dari halaman, harus mengikuti aturan *Document Object Model* dan pengaksesan harus secara spesifik menyesuaikan dengan struktur HTML.

- b. Mengubah tampilan halaman website

CSS (*Cascading Style Sheet*) menawarkan metode yang cukup handal dalam mengatur dan mempercantik halaman web.

c. Mengubah isi dari dokumen

Tidak hanya memberikan “kosmetik” pada halaman web, *JQuery* juga memberikan fasilitas untuk mengubah isi dari dokumen hanya dengan beberapa baris perintah.

d. Merespon interaksi *user*

Javascript sendiri memiliki beberapa *event-handling* seperti *onclick* untuk menangani *event* saat terjadi *click*.

e. Animasi pada dokumen

Animasi seringkali disertakan dalam suatu halaman web untuk menambah kecantikannya.

f. Mengambil informasi dari *server* tanpa harus me-*refresh* halaman

Mengambil informasi dari *server* tanpa *refresh* halaman merupakan salah satu konsep dasar yang dikenal dengan nama AJAX (*Asynchronous Javascript and XML*).

g. Menyederhanakan penulisan *sintaks* Javascript

Semboyan *JQuery* adalah “*Write less, do more*” atau dengan kata lain kesederhanaan dalam penulisan *code*, tetapi menghasilkan tampilan yang lebih.

2.5. Basis Data (*Database*)

Menurut Connolly dan Begg (2002 : 14), pengertian basis data yaitu kumpulan koleksi data-data yang saling berhubungan secara logika yang isinya didesain untuk memenuhi kebutuhan informasi dari suatu perusahaan.

Ada beberapa istilah umum yang sering dipakai pada *database*, yaitu sebagai berikut :

- a. *Field*, yaitu sekumpulan kecil dari kata atau sebuah deretan angka-angka.
- b. *Record*, yaitu kumpulan dari *field* yang berelasi secara logis.
- c. *File*, yaitu kumpulan dari *record* yang berelasi secara logis.
- d. *Entity*, yaitu orang, tempat, benda, atau kejadian yang berkaitan dengan informasi yang disimpan.
- e. *Attribute*, yaitu setiap karakteristik yang menjelaskan suatu *entity*.
- f. *Primary key*, yaitu sebuah *field* yang nilainya unik yang tidak sama antara satu *record* dengan *record* yang lain.
- g. *Foreign key*, yaitu sebuah *field* yang nilainya berguna untuk menghubungkan *primary key* yang berada pada *table* yang berbeda.

2.6. CodeIgniter

Menurut Hakim (2010:8) CodeIgniter adalah sebuah *framework* PHP yang dapat membantu mempercepat developer dalam pengembangan aplikasi web berbasis PHP dibanding jika menulis semua kode program dari awal.



Gambar 2.4 Logo CodeIgniter

Sumber : Hakim (2010 : 9) Membangun Web Berbasis PHP dengan *Framework* CodeIgniter

CodeIgniter pertama kali dibuat oleh Rick Ellis, CEO Ellislab, Inc. (<http://ellislab.com>), sebuah perusahaan yang memproduksi CMS (*Content*

Management System) yang cukup handal, yaitu *Expression Engine* (<http://www.expressionengine.com>). Saat ini, CodeIgniter dikembangkan dan dimaintain oleh *Expression Engine Development Team*.

Adapun beberapa keuntungan menggunakan CodeIgniter, diantaranya:

1. Gratis

CodeIgniter berlisensi dibawah Apache/BSD opensource.

2. Ditulis Menggunakan PHP 4

Meskipun CodeIgniter dapat berjalan di PHP 5, namun sampai saat ini kode program CodeIgniter masih dibuat dengan menggunakan PHP 4.

3. Berukuran Kecil

Ukuran CodeIgniter yang kecil merupakan keunggulan tersendiri. Dibanding dengan *framework* lain yang berukuran besar.

4. Menggunakan Konsep MVC

CodeIgniter menggunakan konsep MVC yang memungkinkan pemisahan *layer application-logic* dan *presentation*.

5. URL yang Sederhana

Secara default, URL yang dihasilkan CodeIgniter sangat bersih dan *Search Engine Friendly* (SEF).

6. Memiliki Paket *Library* yang Lengkap

CodeIgniter mempunyai *library* yang lengkap untuk mengerjakan operasi-operasi yang umum dibutuhkan oleh sebuah aplikasi berbasis web, misalnya mengakses *database*, mengirim email, memvalidasi form, menangani *session* dan sebagainya.

7. *Extensible*

Sistem dapat dikembangkan dengan mudah menggunakan *plugin* dan *helper*, atau dengan menggunakan *hooks*.

8. Tidak Memerlukan *Template Engine*

Meskipun CodeIgniter dilengkapi dengan *template* parser sederhana yang dapat digunakan, tetapi hal ini tidak mengharuskan kita untuk menggunakannya.

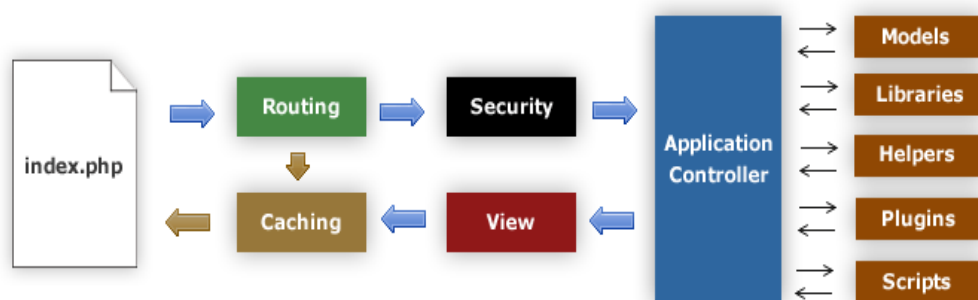
9. Dokumentasi Lengkap dan Jelas

Dari sekian banyak *framework*, CodeIgniter adalah satu-satunya *framework* dengan dokumentasi yang lengkap dan jelas.

10. Komunitas

Komunitas CodeIgniter saat ini berkembang pesat. Salah satu komunitasnya bisa dilihat di (<http://codeigniter.com/forum/>).

Proses aliran data aplikasi pada sistem dapat diilustrasikan seperti terlihat pada gambar 2.5.



Gambar 2.5 *Application Flowchart*

Sumber : Hakim (2010 : 12) Membangun Web Berbasis PHP dengan *Framework* CodeIgniter

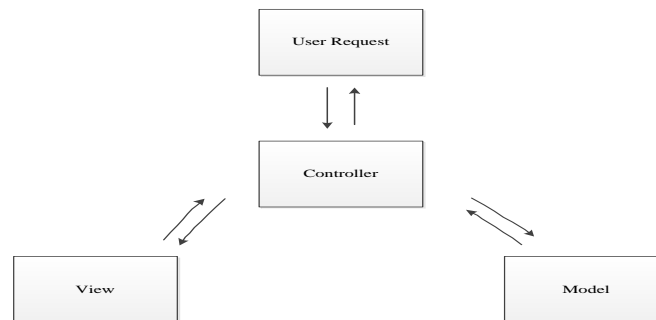
Keterangan :

1. *Index.php* berfungsi sebagai *front controller*, menginisialisasi *base resource* untuk menjalankan *CodeIgniter*.
2. Router memeriksa *HTTP request* untuk menentukan apa yang harus dilakukan dengannya.
3. Jika *Cache* aktif, maka hasilnya akan langsung dikirimkan ke *browser* dengan mengabaikan aliran data normal.
4. *Security*. Sebelum *Controller* dimuat, *HTTP request* dan data yang dikirimkan *user* akan difilter untuk keamanan.
5. *Controller* memuat *model*, *core libraries*, *plugins*, *helpers* dan semua *resource* yang diperlukan untuk memproses *request*.
6. Akhirnya *View* yang dihasilkan akan dikirimkan ke *browser*. Jika *Cache* aktif, maka *View* akan disimpan sebagai *Cache* dahulu, sehingga pada *request* berikutnya langsung dapat ditampilkan.

2.7. MVC (*Model-View-Controller*)

CodeIgniter adalah *framework* PHP yang dibuat berdasarkan kaidah *model-View-controller*. Dengan MVC, maka memungkinkan pemisahan antara *layer application-logic* dan *presentation*. Sehingga, dalam sebuah pengembangan web, seorang *programmer* bisa berkonsentrasi pada *core-system*, sedangkan web *designer* bisa berkonsentrasi pada tampilan web. Menariknya, skrip PHP, *query* MySQL, Javascript dan CSS bisa saling terpisah, tidak dibuat dalam satu skrip berukuran besar yang membutuhkan *resource* besar pula untuk mengesekusinya.

Adapun alur program aplikasi berbasis *framework* Codeigniter dapat dilihat pada gambar 2.6.



Gambar 2.6 *Model-View-Controller*

Sumber : Hakim (2010 : 4) Membangun Web Berbasis PHP dengan *Framework* CodeIgniter

Gambar diatas menerangkan bahwa ketika datang sebuah *user request*, maka akan ditangani oleh *controller*, kemudian *controller* akan memanggil *model* jika memang diperlukan operasi *database*. Hasil dari *query* oleh *model* kemudian akan dikembalikan ke *controller*. Selanjutnya *controller* akan memanggil *view* yang tepat dan mengkombinasikannya dengan hasil *query model*. Hasil akhir dari operasi ini akan ditampilkan di *browser*,

Dalam konteks CodeIgniter dan aplikasi berbasis web, maka penerapan konsep MVC mengakibatkan kode program dapat dibagi menjadi tiga kategori, yaitu :

1. *Model*

Kode program (berupa OOP *class*) yang digunakan untuk memanipulasi *database*.

2. *View*

Berupa *template* html/xml atau php untuk menampilkan data pada *browser*

3. *Controller*

Kode program (berupa OOP *class*) yang digunakan untuk mengontrol aliran aplikasi (sebagai pengontrol *model* dan *View*)

2.8. Intranet (*Internal Network*)

Menurut O'Brien (2003:182) intranet adalah suatu jaringan didalam suatu organisasi yang menggunakan internet. Intranet selain digunakan untuk hubungan antar perusahaan, juga digunakan untuk dalam perusahaan itu sendiri, baik dalam satu gedung maupun dengan cabang perusahaan yang berada diluar kota atau diluar negeri.

2.8.1. Aplikasi Intranet

Intranet menyediakan portal informasi perusahaan yang menunjang *information sharing, communication, collaboration* dan dukungan terhadap operasi bisnis dan manajemen. Berikut ini adalah contoh dari pemanfaatan Intranet:

1. Komunikasi dan Kolaborasi

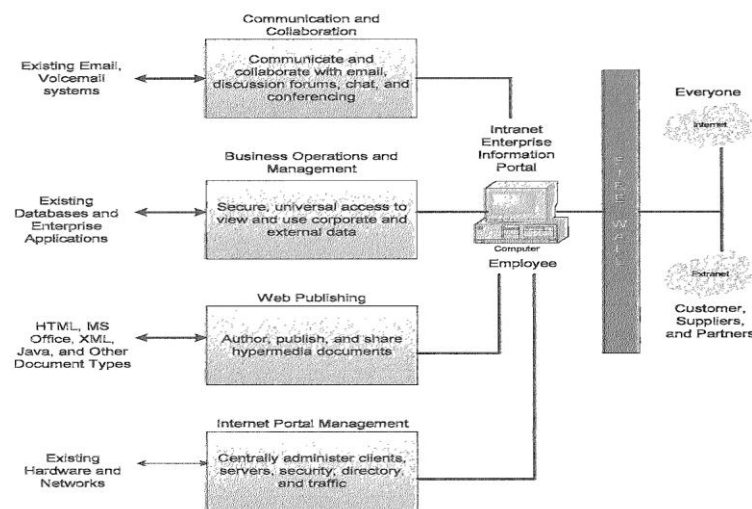
Intranet memungkinkan pekerja untuk saling mengirim dan menerima *email, voice mail, paging, dan fax*.

2. *Web Publishing*

Intranet membuat promosi suatu perusahaan menjadi lebih menarik dan mempunyai biaya lebih rendah.

3. Operasi dan Manajemen Bisnis

Intranet memungkinkan adanya akses umum terhadap data perusahaan dan membuat operasi-operasi bisnis seperti *order processing*, *inventory control*, *sales management* tersedia di Web server, sedangkan para pekerja dan partner bisnis dapat mengakses dan menjalankan aplikasi ini menggunakan Web browser dimanapun dan kapanpun.



Gambar 2.7 Aplikasi Intranet

Sumber : O'Brien (2003 : 183) Pengantar Sistem Informasi : Perspektif Bisnis dan Manajerial.

Berdasarkan gambar 2.7, intranet dapat menyediakan sebuah pintu gerbang informasi perusahaan untuk aplikasi-aplikasi dalam komunikasi dan kolaborasi, operasi dan manajemen bisnis, *Web Publishing*, dan manajemen pintu gerbang intranet.

2.8.2. Keterbatasan Intranet

1. Kurangnya fitur sekuriti karena *computer* dalam jaringan lebih rawan daripada *computer* yang berdiri sendiri.

2. Perlunya *upgrade* jaringan. Komputer *client* harus selalu mengikuti *computer server*, misalnya jika *computer server* berubah sistem operasi, maka *computer client* juga perlu berubah menyesuaikan dengan *computer server*.

2.9. Pemodelan

Pada bab ini penulis akan menjelaskan tentang berbagai konsep dari pengertian pemodelan yang penulis gunakan dalam dari perancangan sistem informasi pengolahan data pemesanan dan produksi *furniture* pada CV. Annisa Prima Karya.

2.9.1. UML (*Unified Modeling Language*)

Dalam bukunya Hariyanto (2004 : 259). Pengembangan UML dimulai dari kerjasama Grady Booch dan James Rumbaugh pada tahun 1994 untuk mengkombinasi dua metodologi terkenal-Booch dan OMT. Kemudian Ivar Jacobson, pencipta metode OOSE (*Object Oriented Software Engineering*) bergabung.

Usulan UML diberikan ke OMG (*Object Management Group*-konsorium standarisasi teknologi objek) agar UML dijadikan bahasa dan notasi pemodelan dilakukan pada 1997. OMG menerima UML, UML telah menjadi *de-facto* karena pencipta-penciptanya sangat populer. Banyak pengembang perangkat lunak yang mengadopsi UML. OMG adalah konsorium yang beranggotakan lebih dari 850 perusahaan untuk mendefinisikan standar-standar teknologi objek termasuk CORBA (*Common Object Request Broker Architecture*). UML menyediakan diagram-diagram yang sangat kaya dan dapat diperluas sesuai kebutuhan kita.

2.9.2. Pengertian UML (*Unified Modelling Language*)

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

1. Didalam *system intensive process*, metode diterapkan sebagai proses untuk menurunkan atau mengevolusikan sistem.
2. Sebagai bahasa, UML digunakan untuk komunikasi yaitu alat untuk menangkap pengetahuan (*semantics*) mengenai satu subyek dan mengekspresikan pengetahuan (*sintaks*) yang mempedulikan subyek untuk maksud berkomunikasi. Subyek adalah sistem yang dibahas.
3. Sebagai bahasa pemodelan, UML *focus* pada pemahaman subyek melalui formulasi model dari subyek (dan *konteks* yang terhubung).
4. Berkaitan dengan unifikasi. UML memadukan praktek rekayasa terbaik sistem informasi dan industri, meliputi beragam tipe sistem (perangkat lunak dan non perangkat lunak), domain (bisnis, perangkat lunak) dan proses siklus hidup.
5. Begitu diterapkan untuk menspesifikasi sistem, UML dapat digunakan untuk mengkomunikasi “apa” yang diperlukan dari sistem dan “bagaimana” sistem dapat direalisasikan.
6. Begitu diterapkan untuk memvisualisasikan sistem, UML dapat digunakan untuk menjelaskan sistem secara visual sebelum direalisasikan.
7. Begitu diterapkan untuk membangun sistem, UML dapat digunakan untuk memandu realisasi sistem serupa dengan “*blueprint*”.

8. Begitu diterapkan untuk mendokumentasikan sistem, UML dapat digunakan untuk menangkap pengetahuan mengenai sistem pada seluruh siklus hidup.

2.9.3. Tujuan UML (*Unified Modelling Language*)

Tujuan utama UML adalah :

1. Menyediakan bahasa pemodelan visual yang ekspresif dan siap pakai untuk mengembangkan dan pertukaran model-model yang berarti.
2. Menyediakan mekanisme perluasan dan spesialisasi untuk memperluas konsep-konsep inti.
3. Mendukung spesifikasi independen bahasa pemograman dan proses pengembangan tertentu.
4. Menyediakan basis formal untuk pemahaman bahasa pemodelan
5. Mendorong pertumbuhan pasar kakas berorientasi objek
6. Mendukung konsep-konsep pengembangan level lebih tinggi seperti komponen, kolaborasi, *framework* dan *pattern*.

2.9.4. Diagram dan Teknik Pemodelan

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

1. Diagram kelas (*Class diagram*)

Diagram kelas merupakan diagram paling umum dipakai disemua pemodelan beorientasi objek. Elemen-elemen esensi di diagram kelas adalah sebagai berikut :

a. Kelas

Kelas merupakan elemen terpenting di sistem berorientasi objek.

b. Antarmuka (*interface*)

Antarmuka (*interface*) merupakan koleksi operasi yang menspesifikasikan layanan dari suatu kelas atau komponen.

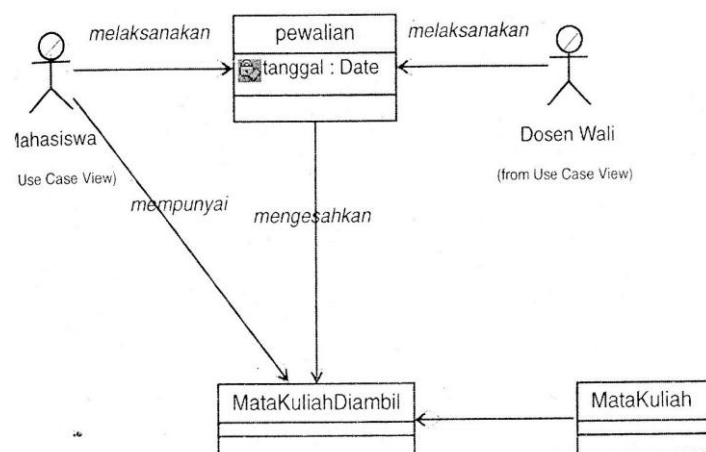
c. Kolaborasi

Kolaborasi merupakan pendefinisian suatu interaksi dan sekelompok peran dan elemen-elemen lain yang bekerja bersama untuk menyediakan suatu perilaku kooperatif yang lebih besar dari penjumlahan seluruh elemen.

d. Hubungan (*relationship*) seperti kebergantungan, generalisasi dan asosiasi.

Hubungan antar kelas di diagram kelas beraneka ragam, yaitu : asosiasi, generalisasi, *dependency*.

Contoh diagram kelas untuk perwalian :



Gambar 2.8 Contoh Diagram kelas untuk perwalian

Sumber : Hariyanto (2004 : 289) Rekayasa Sistem Berorientasi Objek

2. Diagram komponen (*Component diagram*)

Diagram komponen menunjukkan organisasi dan kebergantungan di antara sekumpulan komponen. Elemen esensi diagram komponen berisi :

a. Komponen

Komponen adalah bagian fisik dan dapat diganti sistem yang memenuhi dan menyediakan realisasi sekumpulan antarmuka.

b. Antar muka

c. *Dependency*, generalisasi, asosiasi dan realisasi

d. Paket, untuk mengelompokkan elemen-elemen model menjadi potongan-potongan besar.

3. Diagram *deployment* (*Deployment diagram*)

Diagram *deployment* digunakan untuk memodelkan aspek fisik dari sistem berorientasi objek, yaitu memodelkan konfigurasi *node-node* pengolahan waktu jalan dan komponen-komponen yang tinggal di *node-node* itu. Elemen Esensi diagram *deployment* adalah:

a. *Node*

Node adalah elemen fisik yang ada saat jalan dan merepresentasikan sumberdaya komputasi.

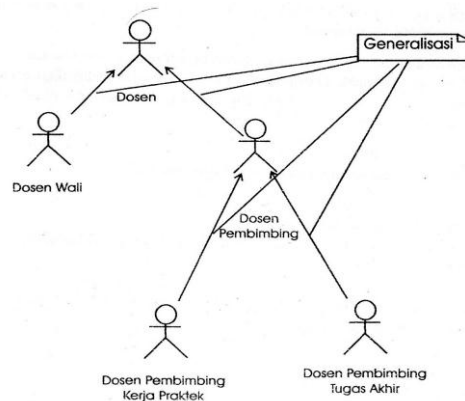
b. Hubungan *dependency* dan asosiasi

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

Use-case merupakan salah satu diagram untuk memodelkan aspek perilaku sistem. Yang termasuk kedalam elemen diagram *use-case* adalah:

a. Aktor

Aktor adalah pemakai sistem, dapat berupa manusia atau sistem terotomatisasi lain . Aktor berkomunikasi dengan sistem lewat pengiriman dan penerimaan pesan.



Gambar 2.9 Aktor-aktor

Sumber : Hariyanto (2004 : 269) Rekayasa Sistem Berorientasi Objek

b. *Use-Case*

Use-Case adalah cara spesifik penggunaan sistem oleh *actor* (Hariyanto (2004 : 270)).



Gambar 2.10 aktor dan *use-case*

Sumber : Hariyanto (2004 : 270) Rekayasa Sistem Berorientasi Objek

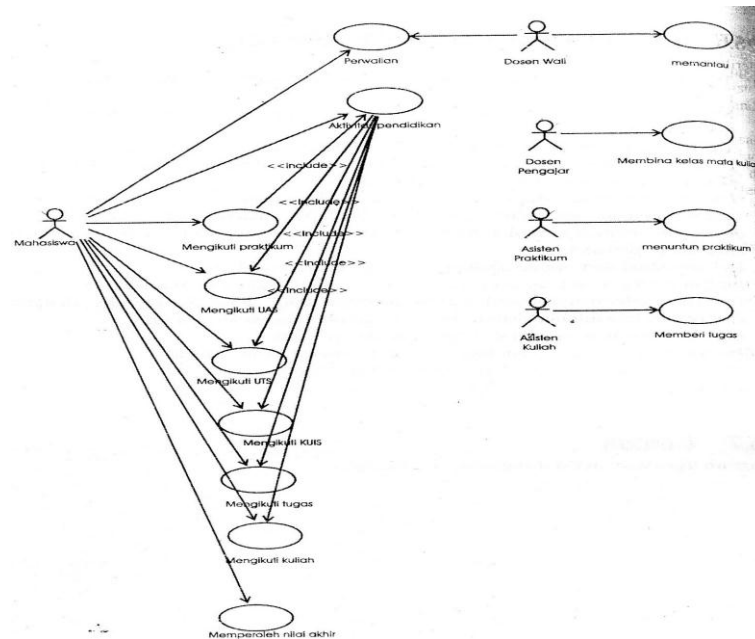
c. Hubungan ketergantungan, generalisasi dan asosiasi.

Keterhubungan antar *use-case* dengan *use-case* lain berupa generalisasi antara *use-case*, yaitu :

a) Include, perilaku *use-case* merupakan bagian dari *use-case* yang lain.

b) Extend, perilaku *use-case* memperluas perilaku *use-case* yang lain.

Contoh *use-case* diagram untuk fungsionalitas sistem :



Gambar 2.11 Diagram *Use-case* Untuk Fungsionalitas Sistem
Sumber : Hariyanto (2004 : 274) Rekayasa Sistem Berorientasi Objek

5. Diagram sekuen (*Sequence diagram*)

Diagram sekuen mendeskripsikan komunikasi di antara objek-objek, meliputi pesan-pesan yang ada dan urutan pesan tersebut muncul. Diagram sekuen digunakan untuk :

- OverView* perilaku sistem
- Menunjukkan objek-objek yang diperlukan
- Mendokumentasikan *scenario* dari suatu diagram *use-case*
- Memeriksa jalur-jalur pengaksesan

6. Diagram kolaborasi (*Collaboration diagram*)

Diagram kolaborasi mendefinisikan peran-peran yang dimainkan ketika satu tugas dilakukan. Diagram kolaborasi digunakan untuk :

- a. Pandangan dalam dari perilaku sistem, berfokus pada *link-link* di antara objek-objek.
- b. Ilustrasi dari suatu diagram *use-case*
- c. Menyatakan objek-objek yang diperlukan untuk merealisasikan suatu layanan
- d. Memeriksa jalur-jalur pengaksesan.

7. Diagram aktivitas (*activity diagram*)

Pada dasarnya, diagram aktivitas adalah diagram *flowchart* yang diperluas yang menunjukkan aliran kendali satu aktivitas ke aktivitas lain. Diagram aktivitas dapat digunakan untuk :

- a. Pandangan dalam yang dilakukan di operasi
- b. Pandangan dalam bagaimana objek-objek bekerja
- c. Pandangan dalam di aksi-aksi pengaruhnya pada objek-objek
- d. Logik dari suatu bisnis

Elemen-elemen diagram aktivitas adalah

- a. *Activity state dan action state*
- b. Transisi
- c. Objek

2.10. Terminologi dalam Proses Bisnis Perusahaan

Berikut ini adalah terminologi dalam proses bisnis perusahaan yang mendasari dari perancangan sistem informasi pengolahan data pemesanan dan produksi.

2.10.1. Pemesanan

Dalam dunia usaha pemesanan adalah hal yang penting dan tidak bisa dihilangkan . menurut Dr. Dendry Sugono Kepala Pusat Bahasa, pengertian pesan adalah suruhan (perintah, nasehat, permintaan, amanat) yang harus dilakukan atau disampaikan kepada orang lain. Pesanan adalah permintaan hendak membeli supaya dikirimkan, dibuatkan dan sebagainya. (<http://pusatbahasa.diknas.go.id/kbbi/index.php>).

2.10.2. Produksi

Menurut Nasution (2003:3), proses produksi merupakan cara, metode, dan teknik untuk menciptakan atau menambah kegunaan suatu produk dengan mengoptimalkan sumber daya produksi (tenaga kerja, mesin, bahan baku, dana).

Menurut Nasution (2003:2) sistem produksi merupakan kumpulan dari subsistem yang saling berinteraksi dengan tujuan menginformasi *input* produksi menjadi *output* produksi.