

## BAB II

### LANDASAN TEORI

#### 2.1 Mesin *Filling*

PKIS Sekar Tanjung untuk pengemasan susu UHT menggunakan lima mesin *filling* milik Tetra Pack yang terdiri dari tiga mesin *Tetra Pack A3/CompactFlex*, dua mesin *Tetra Pack A3/Flex*. Di bawah ini terdapat sub-bagian yang akan menjelaskan tentang mesin-mesin yang telah disebutkan di atas.

##### 2.1.1 *Tetra Pack A3/CompactFlex*

Menurut Tetra Pack (2012), *Tetra Pack A3/CompactFlex* adalah sistem yang benar-benar masih baru yang dikhususkan untuk kemasan-kemasan porsi. Dengan 20 kemasan *Tetra Brik Aseptic* dan *Tetra Prisma Aseptic*, *Tetra Pack A3/CompactFlex* mempunyai variasi kemasan porsi yang paling banyak di pasaran saat ini. *Tetra Pack A3/CompactFlex* memiliki kapasitas 9.000 kemasan/jam.



Gambar 2.1 *Tetra Pack A3/CompactFlex*.

### 2.1.2 *Tetra Pack A3/Flex*

Menurut Tetra Pack (2012), *Tetra Pack A3/Flex* adalah mesin pengemasan untuk kemasan-kemasan *Tetra Brik Aseptic*, *Tetra Prisma Aseptic* dan *Tetra Gemina Aseptic*. Alat ini dirancang agar fleksibel dan dapat diganti dengan mudah antara *volume* dan bentuk kemasan yang berbeda-beda. *Tetra Pack A3/Flex* memproduksi kemasan-kemasan TBA, TGA, TPA mulai dari ukuran 100 sampai 2000 ml, dengan kapasitas 7.500 kemasan/jam.



Gambar 2.2 *Tetra Pack A3/Flex*.

## 2.2 *Capacity Utilization Time & Efficiency (CUTE)*

*Capacity Utilization Time & efficiency (CUTE)* adalah metode yang digunakan untuk mengukur seberapa bagus mesin memproduksi atau metode yang memberikan petunjuk untuk menyamakan persepsi tentang definisi dan kontrol terhadap utilisasi dan pemakaian mesin.

<b>C</b>	<b>Calendar Time</b>		
<b>A</b>	<b>Available time</b>		<b>Unavailable Time</b>
<b>D</b>	<b>Disposable Time</b>		<b>Available Unused Time</b>
<b>O</b>	<b>Operational Time</b>		<b>Planned non Operational Time</b>
<b>P</b>	<b>Production Time</b>		<b>Routine Production Activities</b>
<b>NP</b>	<b>Net Production Time</b>	<b>Unexpected Stoppages</b>	

Gambar 2.3 Klasifikasi Waktu CUTE.

Gambar 2.3 digunakan sebagai panduan untuk menghitung *Time Utilization* (TU), *Production Time Utilization* (PTU), dan *Total Capacity Utilization* (TCU). Keakuratan data adalah hal terpenting untuk memastikan indikator terukur dengan benar.

Penjelasan dari klasifikasi waktu CUTE, adalah sebagai berikut:

- Calendar Time* adalah maksimum waktu dalam periode pelaporan.
- Available Time* adalah waktu dimana mesin dapat berproduksi sesuai dengan peraturan industri dan kesepakatan kerja.
- Unavailable Time* adalah waktu dimana pabrik tidak beroperasi karena terbentur peraturan pemerintah atau tradisi.
- Disposable Time* adalah waktu ketika mesin digunakan untuk kepentingan produksi atau *engineering*, baik mesin dalam keadaan berproduksi atau tidak.

- e. *Available Unused Time* adalah waktu dimana mesin siap dan ada operator tetapi tidak ada jadwal produksi.
- f. *Operational Time* adalah waktu dimana mesin dijadwalkan untuk berproduksi.
- g. *Planned Non Operational Time* adalah waktu dimana mesin tidak produksi terkait dengan kegiatan yang sudah direncanakan.
- h. *Production Time* adalah maksimum waktu yang diharapkan mesin berproduksi secara efektif dan menghasilkan *finish product*.
- i. *Routine Production Activities* adalah waktu yang dibutuhkan untuk melakukan kegiatan operasional untuk memastikan mesin dapat berproduksi.
- j. *Net Production Time* adalah teoritikal waktu yang dibutuhkan mesin untuk menghasilkan *finish product*, jika mesin berjalan sesuai dengan nominal *speed*-nya.
- k. *Unexpected Stoppages* adalah waktu dimana mesin berhenti berproduksi karena *problem* yang terjadi.

Indikator-indikator kinerja mesin yang digunakan oleh metode *Capacity Utilization Time & efficiency* (CUTE), sebagai berikut:

- a. *Time Utilization (TU)* adalah indikator yang digunakan untuk mengukur seberapa efisien mesin dioperasikan oleh produksi, *engineering* dan *quality* ketika mesin dijadwalkan untuk berproduksi.

$$TU = (\text{Net Production Time}) / (\text{Operasional Time}) \dots \dots \dots (2.1)$$

- b. *Production Time Utilization (PTU)* adalah indikator yang digunakan untuk mengukur seberapa efektif mesin dioperasikan oleh bagian *filling* dalam kurun waktu dibawah kontrol produksi.

$$PTU = (\text{Net Production Time})/(\text{Production Time}) \dots\dots\dots (2.2)$$

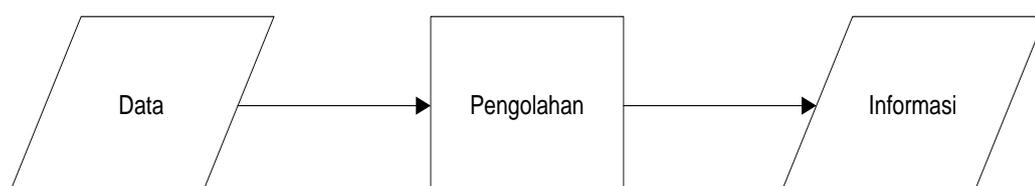
- c. *Total Capacity Utilization (TCU)* adalah indikator yang digunakan untuk mengukur seberapa baik mesin terutilisasi dalam periode *calendar time*.

$$TCU = (\text{Operasional Time})/(\text{Calendar Time}) \dots\dots\dots (2.3)$$

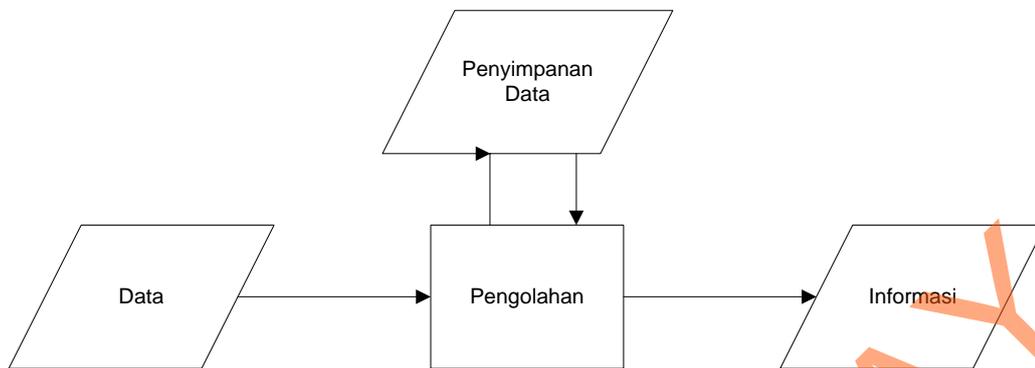
### 2.3 Sistem Informasi

Menurut Hartono (1999), sistem informasi (SI) 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-laporan yang diperlukan. Informasi merupakan hal yang penting bagi manajemen di dalam pengambilan keputusan.

Model dasar pengolahan sistem informasi berguna dalam memahami bukan hanya keseluruhan pengolahan informasi, tetapi juga penerapan pengolahan informasi secara tersendiri. Setiap penerapan dapat dianalisis menjadi masukan, penyimpanan, pengolahan, proses, dan keluaran.



Gambar 2.4 Model Sistem Informasi.



Gambar 2.5 Model SI Dengan Penyimpanan Data.

Sistem informasi yang dibuat oleh manusia, yang terdiri dari komponen-komponen dalam suatu organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi. Untuk dapat berguna, maka suatu sistem informasi harus didukung oleh empat pilar yaitu:

- Lengkap (*complete*)
- Tepat nilainya / akurat (*accurate*)
- Tepat orangnya / relevan (*relevance*)
- Tepat waktu (*timelines*)

Sistem informasi adalah suatu sistem yang memberikan informasi bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang. Kualitas informasi didasarkan pada kelengkapan, keakuratan, ketelitian, ketepatan makna, ketepatan waktu, kejelasan dan fleksibilitas. Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan, yaitu blok masukan, blok model, blok keluaran, blok teknologi, blok basis data dan blok kendali. Sebagai suatu sistem, keenam blok tersebut saling berinteraksi dengan yang lainnya membentuk suatu kesatuan untuk mencapai sasaran.

Alasan utama untuk mendapatkan informasi adalah mengurangi rasa ketidak-pastian. Oleh karena itu, penting sekali bagi kita untuk menentukan kegunaan suatu informasi sebelum membentuk suatu informasi dan menentukan metode pengolahan data untuk mendapatkan informasi yang dibutuhkan dimana informasi ini nantinya akan sangat berguna untuk proses pengambilan keputusan.

#### 2.4 *Monitoring*

Menurut Mercy Corps (2005), *Monitoring* adalah siklus kegiatan yang mencakup pengumpulan, peninjauan ulang, pelaporan, dan tindakan atas informasi suatu proses yang sedang diimplementasikan. Umumnya, *monitoring* digunakan dalam *checking* antara kinerja dan target yang telah ditentukan.

*Monitoring* ditinjau dari hubungan terhadap manajemen kinerja adalah proses terintegrasi untuk memastikan bahwa proses berjalan sesuai rencana (*on the track*). *Monitoring* dapat memberikan informasi keberlangsungan proses untuk menetapkan langkah menuju ke arah perbaikan yang berkesinambungan. Pada pelaksanaannya, *monitoring* dilakukan ketika suatu proses sedang berlangsung.

Menurut Mercy Corps (2005), *monitoring* memiliki 2 (dua) fungsi dasar yang berhubungan, yaitu *compliance monitoring* dan *performance monitoring*. *Compliance monitoring* berfungsi untuk memastikan proses sesuai dengan harapan/rencana. Sedangkan, *performance monitoring* berfungsi untuk mengetahui perkembangan organisasi dalam pencapaian target yang diharapkan.

Umumnya, *output monitoring* berupa *progress report* proses. *Output* tersebut diukur secara deskriptif maupun *non-deskriptif*. *Output monitoring*

bertujuan untuk mengetahui kesesuaian proses telah berjalan. *Output monitoring* berguna pada perbaikan mekanisme proses/kegiatan dimana *monitoring* dilakukan.

#### 2.4.1 Efektifitas *Monitoring*

Menurut Mercy Corps (2005), *monitoring* akan memberikan dampak yang baik bila dirancang dan dilakukan secara efektif. Berikut kriteria sistem *monitoring* yang efektif:

- a. Sederhana dan mudah dimengerti (*user friendly*). *Monitoring* harus dirancang dengan sederhana namun tepat sasaran. Konsep yang digunakan adalah singkat, jelas, dan padat. Singkat berarti sederhana, jelas berarti mudah dimengerti, dan padat berarti bermakna (berbobot).
- b. Fokus pada beberapa indikator utama. Indikator diartikan sebagai titik kritis dari suatu *scope* tertentu. Banyaknya indikator membuat pelaku dan obyek *monitoring* tidak fokus. Hal ini berdampak pada pelaksanaan sistem tidak terarah. Maka itu, fokus diarahkan pada indikator utama yang benar-benar mewakili bagian yang dipantau.
- c. Perencanaan matang terhadap aspek-aspek teknis. Tujuan perancangan sistem adalah aplikasi teknis yang terarah dan terstruktur. Maka itu, perencanaan aspek teknis terkait harus dipersiapkan secara matang. Aspek teknis dapat menggunakan pedoman 5W1H, meliputi apa, mengapa, siapa, kapan, di mana dan bagaimana pelaksanaan sistem *monitoring*.

- d. Prosedur pengumpulan dan penggalian data. Selain itu, data yang didapatkan dalam pelaksanaan *monitoring* pada *on-going process* harus memiliki prosedur tepat dan sesuai. Hal ini ditujukan untuk kemudahan pelaksanaan proses masuk dan keluarnya data. Prosedur yang tepat akan menghindari proses *input* dan *output* data yang salah (tidak akurat).

#### 2.4.2 Tujuan *Monitoring*

Menurut Amsler, dkk (2009), Tujuan *monitoring* dapat ditinjau dari beberapa segi, misalnya segi obyek dan subyek yang dipantau, serta hasil dari proses *monitoring* itu sendiri. Adapun beberapa tujuan dari *monitoring* yaitu:

- a. Memastikan suatu proses dilakukan sesuai prosedur yang berlaku. Sehingga, proses berjalan sesuai jalur yang disediakan (*on the track*).
- b. Menyediakan probabilitas tinggi akan keakuratan data bagi pelaku *monitoring*.
- c. Mengidentifikasi hasil yang tidak diinginkan pada suatu proses dengan cepat (tanpa menunggu proses selesai).
- d. Menumbuh kembangkan motivasi dan kebiasaan positif pekeja.

#### 2.4.3 Bentuk-Bentuk *Monitoring*

*Monitoring* dapat dilakukan dengan berbagai bentuk/metode implementasi. Bentuk implementasi *monitoring* tidak memiliki acuan baku, sehingga pelaksanaan sistem mengacu ke arah improvisasi individu dengan penggabungan

beberapa bentuk. Penggunaan bentuk *monitoring* disesuaikan dengan situasi dan kondisi organisasi. Situasi dan kondisi dapat berupa tujuan organisasi, ukuran dan sifat proses bisnis perusahaan, serta budaya/etos kerja.

Yukl dalam Williams (1998), mengemukakan tujuh bentuk aktifitas dari *monitoring*, yaitu:

- a. Observasi proses kerja, misalnya dengan melakukan *visit* pada fasilitas kerja, pemantauan kantor, lantai produksi, maupun karyawan yang sedang bekerja.
- b. Membaca dokumentasi laporan, berupa ringkasan kinerja dan *progress report*.
- c. Melihat *display* data kinerja lewat layar komputer.
- d. Melakukan inspeksi sampel kualitas dari suatu proses kerja.
- e. Melakukan rapat pembahasan perkembangan secara individual maupun *group*.
- f. Melakukan survei klien/konsumen untuk menilai kepuasan akan produk atau layanan jasa suatu organisasi.
- g. Melakukan survei pasar untuk menilai kebutuhan konsumen sebagai pedoman dalam tindak lanjut perbaikan.

## 2.5 Evaluasi

Menurut Mehrens & Lehmann (1991), evaluasi adalah penilaian yang sistematis tentang manfaat atau kegunaan suatu objek. Dalam melakukan evaluasi terdapat judgement untuk menentukan nilai suatu objek yang sedikit banyak mengandung unsur subjektif. Evaluasi memerlukan data hasil pengukuran dan

informasi hasil penilaian yang memiliki banyak dimensi, seperti kemampuan objek dan sebagainya.

### **2.5.1 Tujuan Evaluasi**

Sebagaimana diuraikan pada bagian terdahulu bahwa evaluasi dilaksanakan dengan berbagai tujuan. Khusus terkait dengan kinerja mesin, evaluasi dilaksanakan dengan tujuan:

- a. Mendeskripsikan kemampuan mesin.
- b. Menentukan tidak lanjut hasil penilaian kinerja mesin.
- c. Memberikan pertanggung jawaban (*accountability*).

### **2.5.2 Fungsi Evaluasi**

Sejalan dengan tujuan evaluasi di atas, evaluasi yang dilakukan juga memiliki banyak fungsi, diantaranya adalah fungsi:

- a. Selektif
- b. Diagnostik
- c. Penempatan
- d. Pengukuran keberhasilan

### **2.5.3 Manfaat Evaluasi**

Secara umum manfaat yang dapat diambil dari kegiatan evaluasi dalam kinerja mesin, yaitu:

- a. Membuat keputusan.
- b. Memahami sesuatu (kondisi mesin).
- c. Meningkatkan kualitas mesin.

## 2.6 Grafik

Menurut Kochhar (2008), grafik adalah gambar datar yang menggunakan titik, garis, atau gambar untuk menunjukkan angka dan data statistik, yang digunakan untuk memperlihatkan statistik atau hubungan. Grafik dibuat menurut spesifikasinya yang pasti, dan menggambarkan secara khusus data kuantitatif untuk analisis, interpretasi, atau perbandingan.

Grafik adalah alat bantu pembelajaran yang efektif untuk membuat perbandingan dan perbedaan. Penggunaan perbandingan secara visual untuk ide yang abstrak akan membantu klarifikasi dan ingatan.

Menurut Santoso (1994), ada beberapa tipe grafik atau diagram yang dapat digunakan untuk menampilkan gambaran informasi supaya lebih jelas, antara lain:

- a. Diagram garis, digunakan untuk menunjukkan perubahan nilai dari sederatan data relatif terhadap waktu, karena diagram garis biasanya digunakan untuk menunjukkan suatu kecenderungan atau *trend*.
- b. Diagram batang, digunakan untuk menyajikan nilai relatif terhadap data yang lain.
- c. Diagram roti (*pie*), digunakan untuk menggambarkan besarnya prosentase data.

## 2.7 Hypertext Preprocessor (PHP)

Menurut Welling (2001), *Hypertext Preprocessor* (PHP) adalah *Server side scripting environment* yang dapat digunakan untuk membuat dan menjalankan aplikasi-aplikasi di *web Server* agar lebih interaktif dan *programmable*. Dengan PHP aplikasi-aplikasi yang ada di *web Server* benar-benar di *web Server* tanpa mengharuskan adanya tambahan atau syarat tertentu untuk sisi *client* (*web browser*). PHP biasanya dijadikan sebagai *module* dalam suatu *web* agar bisa mengeksekusi *file-file* PHP yang tersedia di *web Server*. PHP dapat berjalan di hampir seluruh *platform*, *open source* dan berlisensi GNU *Public license* (GPL).

PHP pada mulanya di tulis sebagai sebuah kumpulan dari CGI dengan menggunakan bahasa pemrograman C oleh *programmer* bernama Rasmus Lerdorf. *Programmer* asal Greenland ini membuat PHP pada tahun 1994 untuk menggantikan sebagian kecil kumpulan *script* dengan *Perl* yang digunakan untuk *maintenance* halaman *web* miliknya. Lerdorf mengawali menciptakan PHP untuk menampilkan *resume* miliknya dan mengumpulkan beberapa data, seperti berapa banyak lalu lintas data yang diterima dalam halaman *web* miliknya.

Setelah mengalami perkembangan oleh suatu kelompok *open source* (termasuk Rasmus) maka mulai versi 3 php menampilkan keunggulan sebagai salah satu bahasa *Server* yang handal. Melalui perkembangan yang pesat ini banyak fasilitas yang ditambahkan oleh kelompok ini, maka jadilah PHP disebut sebagai *Hypertext Preprocessor*. Sintak yang digunakan berasal dari bahasa C, *java* maupun *Perl*.

Aplikasi yang dibangun dengan PHP memiliki kelebihan tersendiri.

Beberapa kelebihan yang dimiliki PHP antara lain:

- a. *Software* ini disebar dan dilisensikan sebagai perangkat lunak yang *open source*, maksudnya pendistribusian kode programnya disertakan juga kode programnya dan biasanya secara gratis.
- b. Dengan menggunakan PHP *script* maka *maintenance* suatu situs *web* menjadi lebih mudah. Proses *update* data dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan menggunakan *script* PHP.
- c. Penulisan *script* PHP dapat menyatu dengan dokumen HTML, sehingga memudahkan pembuatannya. Untuk membedakan dengan sintak HTML dan PHP maka dibuatlah kesepakatan *tag* yang digunakan oleh PHP.
- d. Kemampuan PHP yang paling diandalkan dan signifikan adalah dukungan kepada banyak *database*. Membuat halaman *web* yang digunakan data dari *database* dapat sangat mudah untuk dilakukan. *Database* yang didukung oleh PHP antara lain: *D, dBase, Empress, IBM DB2, Infomix, Ingers, Interbase, Frontbase, file pro (read only), SQL Server, MySQL, Oracle, ODBC, PostgreSQL, Solid, Sysbase, Velocis, dan unix DBM.*

## 2.8 Apache

Menurut Apache dalam Wibowo (2010), *Apache* adalah *web-Server* yang mendukung bahasa PHP sehingga dapat dipakai untuk implementasi aplikasi berbasis PHP. *Web-Server* akan menerjemahkan bahasa PHP yang dipakai pada aplikasi *score online* untuk ditampilkan secara visual pada *browser*.

## 2.9 *My Structure Query Language (MySQL)*

Menurut Anhar (2010), *My Structure Query Language (MySQL)* adalah sebuah perangkat lunak sistem, manajemen basis data SQL atau DBMS dari sekian banyak DBMS, seperti Oracle, MS SQL, Postagre SQL, dan lain-lain. MySQL merupakan DBMS yang *Multithread, multi-user* yang bersifat gratis di bawah lisensi GNU General Public Licence (GPL).

Menurut Anhar (2010), beberapa kelebihan MySQL, antara lain:

- a. MySQL dapat berjalan dengan stabil pada berbagai sistem operasi, seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, dan banyak lagi.
- b. Bersifat *open source*, MySQL didistribusikan secara *open source* (gratis), di bawah lisensi GNU *Gerenal Public Licence* (GPL).
- c. Bersifat *Multiuser*, MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah.
- d. MySQL memiliki kecepatan yang baik dalam menangani *query* (perintah SQL). Dengan kata lain, dapat memproses lebih banyak SQL per satuan waktu.
- e. Dari segi *security* atau keamanan data, MySQL memiliki beberapa lapisan *security*, seperti level *subnet mask*, nama *host*, dan izin akses *user* dengan sistem perijinan yang mendetail serta *password* yang terenkripsi.

## 2.10 *Framework*

Menurut Lukmanul Hakim (2010), *framework* adalah koleksi atau kumpulan potongan-potongan program yang disusun atau diorganisasikan

sedemikian rupa, sehingga dapat digunakan untuk membantu membuat aplikasi utuh tanpa harus membuat semua kodenya dari awal.

Menurut Daqiqil (2011), keuntungan yang didapat dalam penggunaan *framework* adalah:

- a. Menghemat waktu pengembangan, dengan struktur dan *library* yang telah disediakan oleh *framework* maka tidak perlu lagi memikirkan hal-hal tersebut.
- b. *Reuse of code*, dengan menggunakan *framework* maka pekerjaan kita akan memiliki struktur yang baku.
- c. Bantuan Komunitas, ada komunitas-komunitas yang siap membantu jika ada permasalahan.
- d. Kumpulan *best practice*, sebuah *framework* merupakan kumpulan *best practice* yang sudah teruji.

### 2.11 Codeigniter

Menurut Lukmanul Hakim (2010), *Codeigniter* adalah sebuah *framework* PHP yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi *web* berbasis PHP dibandingkan jika menulis semua kode program dari awal. *Codeigniter* menyediakan banyak *library* untuk mengerjakan tugas-tugas yang umumnya ada pada sebuah aplikasi yang anda buat menjadi semakin teratur dan rapi.

Menurut Lukmanul Hakim (2010), Adapun beberapa keuntungan menggunakan *codeigniter*, diantaranya:

a. Gratis

*Codeigniter* berlisensi dibawah Apache/BSD *open source*, jadi anda bisa menggunakan secara bebas.

b. Ditulis menggunakan PHP 4

Meskipun *codeigniter* dapat berjalan pada PHP 5, namun samapai saat ini *kode program codeigniter* masih dibuat dengan menggunakan PHP 4.

c. Berukuran kecil

Ukuran *codeigniter* yang kecil merupakan keunggulan tersendiri dibanding *framework* lain yang berukuran besar. Pada *codeigniter*, bisa diatur agar *system me-load library* yang dibutuhkan saja sehingga sistem dapat berjalan ringan dan cepat.

d. Menggunakan konsep M-V-C

*Codeigniter* menggunakan konsep M-V-C (*Model-View-Controller*) yang memungkinkan pemisahan antara *layer application-logic* dan *presentation*.

e. URL yang sederhana

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

f. Memiliki paket *library* yang lengkap

*Codeigniter* memiliki *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.

g. *Extensible*

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

h. Tidak memerlukan *template engine*

Meskipun *codeigniter* dilengkapi dengan *template parser* sederhana yang dapat anda gunakan, tetapi hal ini mengharuskan anda untuk menggunakannya. Penggunaan *Template Engine* dapat mengurangi *performance* dari sistem.

i. Dokumentasi lengkap dan jelas

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

## 2.12 Testing dan Implementasi Sistem

Menurut Standard ANSI/IEEE 1059 dalam Wibowo (2010), *testing* adalah proses menganalisa suatu entitas *software* untuk mendeteksi perbedaan antara kondisi yang ada dengan kondisi yang diinginkan (*defects/error/bugs*) dan mengevaluasi fitur-fitur dari entitas *software*.

Menurut Romeo (2003), *testing software* adalah proses mengoperasikan *software* dalam suatu kondisi yang dikendalikan untuk *verifikasi*, mendeteksi *error*, dan validasi.

Menurut Romeo (2003), *test case* merupakan tes yang dilakukan berdasarkan pada suatu inisialisasi, masukan, kondisi ataupun hasil yang telah

ditentukan sebelumnya. Metode *testing* ini dibagi menjadi dua, yaitu *white box testing* dan *black box testing*.

### 2.12.1 *White Box Testing*

*White box testing* atau *glass box testing* atau *clear box testing* adalah suatu metode desain *test case* yang menggunakan struktur kendali dari desain prosedural. Metode desain *test case* ini dapat menjamin:

- a. Semua jalur (*path*) yang independen/terpisah dapat dites setidaknya sekali tes
- b. Semua logika keputusan dapat dites dengan jalur yang salah atau jalur yang benar.
- c. Semua *loop* dapat dites terhadap batasannya dan ikatan operasionalnya.
- d. Semua struktur internal data dapat dites untuk memastikan validasinya.

### 2.12.2 *Black Box Testing*

*Black box testing* atau *behavioral testing* atau *specification-based testing*, *input/output testing* atau *functional testing* dilakukan tanpa sepengetahuan detail struktur internal dari sistem atau komponen yang dites. *Black box testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan spesifikasi kebutuhan dari *software*.

Menggunakan *black box testing*, perencana *software* dapat menggunakan sekumpulan kondisi masukan yang dapat secara penuh memeriksa keseluruhan

kebutuhan fungsional pada suatu program. Kategori *error* dapat diketahui melalui *black box testing*, antara lain:

- a. Fungsi yang hilang atau tidak benar.
- b. *Error* dari antar-muka.
- c. *Error* dari struktur data atau akses *eksternal database*.
- d. *Error* dari kinerja atau tingkah laku.
- e. *Error* dari inisialisasi dan terminasi.

### 2.12.3 *Testing Aplikasi Berbasis Web*

Menurut Romeo (2003), dengan adanya perkembangan teknologi internet, berkembanglah kebutuhan aplikasi berbasis *web*, baik untuk keperluan internet organisasi. Terdapat beberapa hal yang berkaitan dengan kualitas aplikasi berbasis *web*, antara lain:

- a. **Komplesitas Aplikasi**

*Web* merupakan aplikasi yang paling berkembang saat ini, baik dari segi kompleksitas, manajemen *query* pada *database* yang sangat besar, atau metode *searching* yang ada. *Website* lebih kompleks dari yang terlihat, karena *website* menggunakan teknologi GUI, *Network Connectivity* dan *Database Acces*. Beberapa pengamat menyatakan bahwa teknologi *client/server* akan digantikan oleh internet, tapi kenyataan yang berkembang adalah teknologi gabungan dari keduanya. Inilah alasan mengapa *client/server testing* yang dibahas sebelumnya juga berkaitan dengan *sub-bab* ini.

b. Keterbatasan Alat Bantu

Hala yang tidak dapat dibantah adalah alat bantu pengembangan aplikasi berbasis *web* saat ini masih memiliki keterbatasan yang sangat mengganggu.

Aplikasi *web* dibangun dengan alat bantu standar yang menghasilkan *pages* statis, sehingga pengguna tidak dapat dengan mudah men-*download* data ke *desktop analysis tool* seperti *excel spreadsheet*, *pdf*, dan lain-lainnya.

Produk *web* merupakan aplikasi yang paling cocok mengalami penambahan versi oleh karena itu manajemen tes yang diperlukan juga harus handal, karena hal ini berhubungan dengan kualitas dari aplikasi itu sendiri.

c. Kompatibilitas

*Web pages* akan terlihat berbeda jika dilihat dari *web browser* yang berbeda, karena perbedaan implementasi dari *HTML standard*.

*Web pages* dapat diakses dari beberapa *platform* yang berbeda, seperti Win NT, Win 95, OS/2, Mac dan lain-lain. Ini artinya testing perlu dilakukan pada berbagai *platform* dan konfigurasi yang berbeda.

d. Performansi

Hal yang paling sulit untuk dites adalah pengukuran kecepatan akses. *Response time* dari *web*, karena hal ini bukan hal yang mudah untuk dipecahkan dengan biaya yang murah.

Banyak faktor yang menjadi penyebab seperti *loads* yang tidak dapat diprediksi, *web* yang menjadi favorit bisa menerima ribuan pengunjung per-hari dibandingkan dengan *web* biasa yang pengunjungnya hanya ratusan.

e. Kegunaan

Beberapa pengguna mungkin punya harapan sendiri-sendiri tentang bagaimana *website* yang menarik. Seperti contohnya *web pages* harus dapat mudah untuk disimpan. Oleh karena itu *web pages* harus terlihat atraktif agar menarik perhatian dari pengguna. Ada beberapa pengguna yang sangat sensitif dan terganggu jika keluar kata masuk dari suatu *web pages* tanpa suatu *permission* atau *awareness*.

f. Keamanan

Sistem keamanan merupakan hal yang sangat penting dalam aplikasi berbasis *web*, karena aplikasi ini dibangun untuk dapat diakses oleh pengguna atau aplikasi yang baik itu dalam suatu intranet ataupun extranet dengan sama baiknya. Hak akses eksternal memang dibatasi tapi tidak menutup kemungkinan terjadinya *hacking* terhadap aplikasi.

g. Organisasional

Telah dijelaskan diatas bahwa teknologi ini merupakan inovasi yang sangat fenomenal. Oleh karena itu, mungkin dalam perkembangannya yang kurang diperhatikan adalah kendali kualitas dan standar *testing* yang baik. Yang terjadi pada pengembangan intranet yang mengambil alih semua proses pembangunan dari suatu aplikasi *web* mulai dari desain hingga proses testing. Dalam beberapa organisasi intranet membuat kekacauan karena kurangnya koordinasi. Setiap orang mempunyai *web* internal pribadi. Setiap orang punya ide sendiri-sendiri bagaimana membuat *web*-nya, apa isinya, dan bagaimana harus berjalan. Sehingga terjadi kekacauan pada kepemilikan dan hak akses

informasi juga pertanyaan siapa yang bertanggung jawab atas kualitas dari informasi dan *maintenance* dari aplikasi itu sendiri.

Tipe-tipe *testing* pada aplikasi berbasis *web*, antara lain:

- a. *Content* dan *funcionality testing*. *Testing* terhadap isi dan fitur seperti yang terdapat pada *website* umumnya, pastikan sudah lengkap dan berjalan sesuai dengan yang diinginkan.
- b. *Feature interaction testing*. Banyak pengguna yang secara simultan mengakses satu *site* yang sama dan tidak boleh terjadi interfrensi antara mereka.
- c. *Usability testing*. Melakukan *testing* apakah *website* sudah *user friendly*.
- d. *Database testing*. Memastikan *database* dapat diakses dari *website* yang mempunyai kendali integritas dan kecukupan data.
- e. *Security* dan *control testing*. Memastikan *site* ini aman, termasuk *account setup*, *billing*, dan dari *unauthorized acces*.
- f. *Connectivity testing*. Pastikan *website* dapat melakukan *connection* atau *disconnection*.
- g. *Interoperability testing*. Pastikan semua *web browser* dari semua versi ada jenis komputer yang berbeda dapat berjalan dengan baik pada aplikasi ini.
- h. *Cross platform* dan *configuration testing*. Pastikan perilaku dari sistem kompatibel dalam *platform* dan konfigurasi yang berbeda.
- i. *Performance* dan *stress testing*. Ukur kemampuan, *response time* dan semua proses yang terjadi dalam keadaan *workloads* diatas rata-rata atau dibawah rata-rata.

- j. *Internazionalization testing*. Pastikan *site* tidak membingungkan atau menyerang pengguna.
- k. *Beta testing*. Undang beberapa pengguna terpilih untuk melakukan eksperimen pada *site* anda dan mintalah *feedback* pada mereka sebelum *website* itu diluncurkan.
- l. *Standard compliance testing*. Pastikan *website* itu kompetibel dengan internet standar. Apakah terlihat sama meskipun menggunakan *browser* atau *search engines* yang berbeda.

### 2.13 System Development Life Cycle (SDLC)

Menurut Kendall & Kendall (2006), *System Development Life Cycle* (SDLC) adalah pendekatan bertahap untuk melakukan analisa dan membangun rancangan sistem dengan menggunakan siklus yang spesifik terhadap kegiatan pengguna dalam membangun sistem informasi. Dalam sebuah siklus SDLC, terdapat enam langkah yaitu identifikasi masalah, menentukan kebutuhan informasi, analisis kebutuhan sistem, perancangan sistem, implementasi sistem, perawatan dan pengembangan sistem.

#### 2.13.1 Identifikasi Masalah

Pada kegiatan pertama ini, Seluruh kebutuhan *software* harus bisa didapatkan analis sistem, termasuk di dalamnya kegunaan *software* yang diharapkan pengguna dan batasan *software*. Mengidentifikasi tujuan merupakan

salah satu komponen yang penting karena analisis sistem harus mengetahui apa yang ingin dicapai oleh perusahaan tersebut. Selain itu, analisis sistem juga harus mengetahui bila ada aspek-aspek aplikasi sistem informasi yang dapat digunakan untuk membantu mencapai tujuan perusahaan dengan merumuskan masalah yang spesifik. Keluaran dari tahap ini adalah informasi kelayakan yang berisikan definisi masalah dan rangkuman dari tujuan.

### **2.13.2 Menentukan Kebutuhan Informasi**

Pada tahap selanjutnya menentukan informasi apa saja yang dibutuhkan. Cara-cara yang digunakan untuk mendapatkan informasi yang dibutuhkan adalah seperti wawancara, membuat sampel dan menginvestigasi *hard* data, dan kuesioner. Orang-orang yang ikut serta dalam tahap ini adalah analisis sistem dan pengguna (manajer operasi dan pegawai). Analisis sistem harus mengetahui detail dari sistem yang sudah berjalan sekarang, siapa saja orang yang terlibat, apa jenis aktivitas bisnisnya, dimana pekerjaan ini akan dilakukan, waktunya, dan bagaimana prosedur yang sekarang dijalankan. Inti dalam fase ini adalah analisis sistem harus bisa mengetahui bagaimana fungsi bisnis yang sudah berjalan dan mempunyai informasi yang lengkap atas orang, tujuan, data, dan prosedur yang bersangkutan.

### 2.13.3 Analisis Sistem

Analisis sistem dapat didefinisikan sebagai pengguna dari suatu sistem informasi yang utuh kedalam bagian-bagian komponen dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan, sehingga dapat diusulkan perbaikan-perbaikan. Tahap ini merupakan tahap yang kritis dan penting karena kesalahan pada tahap ini akan menyebabkan kesalahan pada tahap berikutnya. Langkah-langkah dasar yang harus dilaksanakan oleh analisis sistem yaitu:

1. Memahami sistem dan membatasinya.
2. Alternatif-alternatif apa saja yang ada untuk mencapai sasaran dan untuk memodifikasi atau mengubah sistem.
  - a. Pilih satu dari alternatif yang telah diidentifikasi pada tahap sebelumnya.
  - b. Implementasikan alternatif yang dipilih.
  - c. Evaluasi masalah dari perubahan yang kita buat dalam sistem.

### 2.13.4 Perancangan Sistem

Alternatif yang telah dipilih dalam langkah analisa sistem merupakan dasar dari perancangan sistem. Rancangan sistem menentukan bagaimana suatu sistem akan menyelesaikan apa yang harus diselesaikan. Tahap ini menyangkut konfigurasi dari komponen-komponen perangkat keras dan perangkat lunak sistem sehingga setelah menginstalasi sistem akan benar-benar akan memuaskan spesifikasi sistem yang telah ditetapkan pada akhir analisa sistem.

### 2.13.5 Implementasi Sistem

Aplikasi yang telah selesai dibuat oleh *programmer* harus diuji dan siap diimplementasikan kedalam sistem pengguna. Tahap dari implementasi sistem adalah sebagai berikut:

1. Membangun dan menguji jaringan *database*
2. Membangun dan menguji program
3. Instalasi dan menguji sistem yang baru
4. Penyerahan sistem yang telah dibuat

### 2.13.6 Perawatan dan Pengembangan Sistem

Diperlukan adanya kegiatan tambahan setelah sistem yang baru dijalankan, seperti merawat dan menjaga agar sistem tetap berjalan sesuai dengan apa yang dikehendaki. Perlu juga diperhatikan akibat adanya kebijaksanaan yang baru yaitu perubahan-perubahan prosedur, agar sistem tetap menjalankan fungsinya sehingga pengembangan sistem diperlukan.