

## **BAB II**

### **LANDASAN TEORI**

Dalam merancang dan membangun aplikasi, sangatlah penting untuk mengetahui terlebih dahulu dasar-dasar teori yang digunakan. Dasar-dasar teori tersebut digunakan sebagai landasan berpikir dalam melakukan pembahasan lebih lanjut sehingga terbentuk suatu aplikasi yang sesuai dengan tujuan awal.

#### **2.1 Dokumen**

Dokumen merupakan sumber tertulis bagi informasi sejarah sebagai kebalikan daripada kesaksian lisan, artefak, peninggalan-peninggalan terlukis dan petilasan-petilasan arkeologis. Dokumen diperuntukan untuk surat-surat resmi dan surat-surat Negara seperti surat perjanjian, undang-undang, hibah dan konsesi. Dokumen dalam arti luas merupakan proses pembuktian yang didasarkan atas sumber jenis apapun, baik yang bersifat tulisan, lisan, gambaran atau arkeologis (Gottschalk, 1986).

Dokumen adalah Segala benda yang berbentuk barang, gambar, ataupun tulisan sebagai bukti dan dapat memberikan keterangan yang penting dan absah. Dokumen dalam arti luas yaitu meliputi semua sumber tertulis saja, baik tertulis maupun lisan. Dokumen dalam arti sempit yaitu yang meliputi semua sumber tertulis saja. Dokumen dalam arti spesifik yaitu hanya meliputi surat-surat resmi dan surat-surat Negara, seperti surat perjanjian, undang-undang, konsesi, hibah dan sebagainya (Renier, 1997).

## **2.2 Pengelolaan**

Pengelolaan adalah proses membantu merumuskan kebijakan dan tujuan organisasi atau proses yang memberikan pengawasan pada suatu hal yang terlibat dalam pelaksanaan kebijakan dan pencapaian tujuan. Pengelolaan sama dengan manajemen yaitu penggerakan, pengorganisasian dan pengarahan usaha manusia untuk memanfaatkan secara efektif material dan fasilitas untuk mencapai tujuan organisasi.

Pengelolaan atau manajemen adalah proses perencanaan, pengorganisasian, pengarahan, dan pengawasan usaha-usaha para anggota, organisasi dan penggunaan sumber daya. Sumber daya organisasi lainnya agar mencapai tujuan organisasi yang telah ditetapkan (Handoko, 2003).

### **2.2.1 Fungsi Pengelolaan**

Pengelolaan merupakan ilmu pengetahuan juga dalam artian bahwa manajemen memerlukan disiplin ilmu-ilmu fungsi pengelolaan sebagai pengetahuan lain dalam penerapannya secara universal (Handoko, 2003). fungsi-fungsi pengelolaan sebagai berikut: Perencanaan, Pengorganisasian, Penyusunan Personalia, Pengarahan, Pengawasan.

## **2.3 Bahan Baku**

Bahan baku merupakan bahan yang membentuk bagian menyeluruh (Mulyadi, 2005). Bahan baku merupakan bahan yang membentuk bagian besar produk jadi, bahan baku yang diolah dalam perusahaan manufaktur dapat diperoleh dari pembelian lokal, impor atau hasil pengolahan sendiri (Kholmi, 2003). Dari definisi tersebut dapat disimpulkan bahwa bahan baku merupakan

bahan yang utama di dalam melakukan proses produksi sampai menjadi barang jadi. Bahan baku meliputi semua barang dan bahan yang dimiliki perusahaan dan digunakan untuk proses produksi (Wibowo, 2007).

#### **2.4 Workflow Management System**

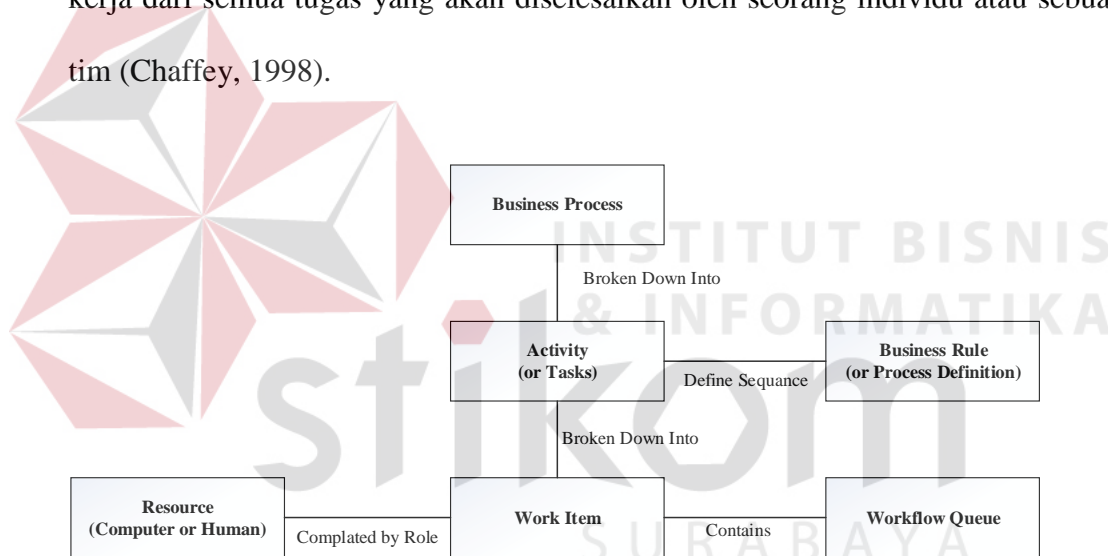
*Workflow Management System* (WfMS) sebagai tipe perangkat lunak khusus yang digunakan untuk membantu kerja kolaboratif dengan komputer dan sering disebut sebagai otomatisasi *Workflow*, karena WfMS bisa mengotomatisasi tugas atau aktifitas yang dilakukan manusia atau komputer dari sebuah organisasi (Chaffey, 1998).

*Workflow Management Coalition* (WfMC) menggambarkan *workflow* sebagai fasilitas komputerisasi atau otomatisasi proses bisnis secara keseluruhan atau sebagian, sedangkan WfMS digambarkan sebagai sebuah sistem yang mendefinisikan, menciptakan, dan mengelola pelaksanaan *workflow* melalui penggunaan perangkat lunak, yang berjalan pada satu atau lebih *workflow engine*, yang mampu menafsirkan definisi proses, berinteraksi dengan peserta alur kerja dan, jika diperlukan, meminta penggunaan alat dan aplikasi TI.

*Workflow* dapat memberikan perbedaan yang besar pada efisiensi operasional dari proses yang ada pada sebuah bisnis. *Workflow* dapat membantu manajer dalam mengoordinasikan tugas-tugas yang dilakukan oleh staf dan memberikan informasi kepada staf untuk membantu manajer melakukan tugas-tugasnya. Keuntungan bisnis utama dari penerapan sebuah sistem *workflow* adalah waktu penyelesaian dan biaya dari proses bisnis yang ada sekarang dapat dikurangi (Chaffey, 1998).

### 2.4.1 Elemen Kerja Kunci dalam *Workflow System*

Sebuah *workflow* dapat digambarkan sebagai suatu hal yang terdiri atas serangkaian kegiatan, yang bersama-sama, membentuk sebuah proses bisnis. Pada Gambar 2.1 akan dijelaskan bagaimana sebuah kegiatan dipecah menjadi *workitem* individu yang harus diselesaikan. Setiap *workitem* disini dilakukan oleh sebuah *resource*, baik perangkat lunak, perangkat keras, atau seorang personil yang memiliki tanggung jawab untuk melakukannya. *Work item* yang akan diselesaikan ditunjukkan pada sebuah *workflow queue*, yang adalah sebuah daftar kerja dari semua tugas yang akan diselesaikan oleh seorang individu atau sebuah tim (Chaffey, 1998).



Gambar 2.1 Kunci Kerja dalam *Workflow System* (Chaffey, 1998)

#### 1. *Process Elements (Work Activities and Tasks)*

Aktifitas kerja atau tugas adalah unit kerja individu yang membentuk *workflow*. Aktifitas-aktifitas ini biasanya bisa diuraikan menjadi sub tugas yang membentuk sebuah hirarki tugas. Pada saat sebuah aktifitas kerja diselesaikan, perubahan status sebuah obyek akan terjadi dan perlu dicatat oleh sistem.

## 2. *Resources and Their Roles*

*Resources* adalah sumber daya manusia atau komputer yang melakukan aktifitas kerja yang membangun proses bisnis. *User* atau *computer resource*, yang dikenal sebagai *workflow participant* diberikan satu atau beberapa peran yang akan menentukan apakah mereka dapat melakukan tugas tertentu. Penggunaan peran daripada individu lebih penting karena akan memudahkan untuk memindahkan tanggung jawab seseorang ke orang lain dengan peran yang sama. Pada situasi tertentu, adalah penting untuk menentukan bahwa sebuah tugas ditingkatkan pada sebuah peran yang berbeda.

## 3. *Dependencies and Business Rules*

*Dependencies* (dependensi) menjelaskan bagaimana aktifitas yang berbeda berhubungan satu sama lain. Dependensi didefinisikan oleh *business rules* yang membangun *workflow*. Urutan dari aktifitas dapat diatur berdasarkan *pre-condition* atau *post-condition* yang harus dipenuhi sebelum mulai atau selesainya sebuah aktifitas.

## 4. *Workflow Queue*

Sistem *workflow* biasanya menerapkan sebuah antrian *workflow* yang digunakan untuk menugaskan sebuah tugas ke individu. Sebuah urutan *workflow* akan menampung sebuah daftar tugas atau aktifitas yang harus dikerjakan dalam sebuah urutan prioritas.

## 5. *Case Management*

Penggunaan dari sebuah *case* atau tiruan dari *folder* adalah sebuah hal yang umum pada sistem *workflow*. Sebuah *case* akan terdiri atas sebuah *instance* tunggal dari subyek dan obyek yang utama dari *workflow*, yaitu *customer*.

Setiap *case* dapat digambarkan sebagai sebuah berkas dari sebuah lemari arsip yang menyimpan semua informasi yang berhubungan dengan *customer*.

#### 6. *Messaging*

Pesan tambahan dapat dikirim antara teman sekerja ketika terjadi kejadian yang tidak biasa, yang mengganggu lancarnya jalan dari sistem. Sistem mungkin menggunakan *standard company mail system*, atau sistem *workflow* akan mengijinkan sebuah notifikasi untuk dikeluarkan atau dapat mengijinkan perubahan jalur sebuah tugas atau pencabutan sebuah tugas.

#### 2.4.2 *Administrative Workflow System*

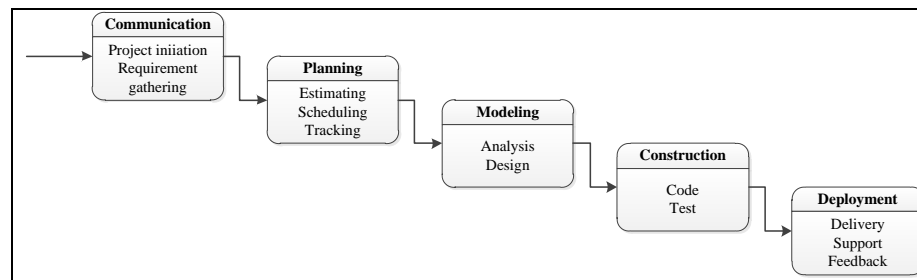
*Administrative Workflow System* adalah sebuah sistem *workflow* umum digunakan, yang memanfaatkan penggunaan form elektronik yang terhubung dengan surat elektronik. Sistem ini biasa diaplikasikan ke dalam tugas-tugas administrasi rutin seperti persetujuan pengajuan liburan, pemrosesan pemesanan pembelian, dll. The Gartner Group memperkirakan bahwa 83% dari semua dokumen bisnis di US adalah dokumen formulir dengan biaya pembelian tahunan sebesar 6-8 milyar USD dan biaya pemrosesan mencapai 360 milyar USD. Formulir-formulir kertas ini menjadi target dari 1995 *Paper Reduction Act* (Chaffey, 1998)

Manfaat yang besar dapat terjadi melalui mengotomatisasikan proses berbasis formulir. proses dapat berbalik lebih cepat menggunakan formulir elektronik dan mengurangi biaya melalui pengurangan biaya pembelian formulir dan waktu siklus yang lebih pendek. salah satu penghematan biaya terbesar adalah dalam koordinasi pengolahan formulir yang sekarang ditangani oleh logika bisnis yang dibangun ke dalam aplikasi (Chaffey, 1998)

## 2.5 Siklus Hidup Pengembangan Sistem

Siklus Hidup Pengembangan Sistem, nama lain dari *System Development Life Cycle* (SDLC) ini merupakan suatu proses pengembangan atau perubahan suatu sistem perangkat lunak. Pengembangan atau perubahan tersebut dilakukan dengan menggunakan model-model dan metodologi yang digunakan oleh banyak orang, yang telah mengembangkan sistem-sistem perangkat lunak sebelumnya. Hal tersebut tentu berdasarkan *best practice* atau cara-cara yang telah teruji dengan baik oleh banyak orang yang menggunakannya. SDLC memiliki beberapa model dalam penerapan tahapan prosesnya. Beberapa model SDLC tersebut antara lain yaitu Model *Waterfall*, *Spiral*, *Rapid Application Development*, *Agile* dan *Prototype*. Masing-masing model memiliki kelemahan dan kelebihan, sehingga hal yang terpenting adalah mengenali tipe pelanggan dan memilih menggunakan model SDLC yang sesuai dengan karakter pelanggan dan sesuai dengan karakter pengembang perangkat lunak (Kendall dan Kendall, 2008).

*System Development Life Cycle* (SDLC) ini biasanya disebut juga dengan model *waterfall*. Menurut Pressman (2012), nama lain dari Model *Waterfall* adalah Model Air Terjun, kadang dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menyiratkan pendekatan yang sistematis dan berurutan (sekuensial) pada pengembangan perangkat lunak. Pengembangan perangkat lunak dimulai dari spesifikasi kebutuhan pengguna dan berlanjut melalui tahapan-tahapan perencanaan (*planning*), pemodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem perangkat lunak ke para pelanggan/pengguna (*deployment*), yang diakhiri dengan dukungan berkelanjutan pada perangkat lunak yang dihasilkan (Pressman, 2012).



Gambar 2.2 Pengembangan menggunakan Model *Waterfall* (Pressman, 2012)

Gambar 2.2 menunjukkan tahapan umum dari model proses *waterfall*.

Model ini disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Akan tetapi, Pressman (2012) memecah model ini meskipun secara garis besar sama dengan tahapan-tahapan model *waterfall* pada umumnya.

Model ini merupakan model yang paling banyak dipakai dalam *Software Engineering*. Model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ketahap *Communication*, *Planning*, *Modeling*, *Construction*, dan *Deployment*.

Berikut ini adalah penjelasan dari tahap-tahap yang dilakukan di dalam Model *Waterfall* (Pressman, 2012):

a. *Communication*

Langkah pertama diawali dengan komunikasi kepada konsumen/pengguna. Langkah awal ini merupakan langkah penting karena menyangkut pengumpulan informasi tentang kebutuhan konsumen/pengguna.

b. *Planning*

Setelah proses *communication* ini, kemudian menetapkan rencana untuk pengerjaan *software* yang meliputi tugas-tugas teknis yang akan dilakukan,



risiko yang mungkin terjadi, sumber yang dibutuhkan, hasil yang akan dibuat, dan jadwal pengerjaan.

c. *Modeling*

Pada proses *modeling* ini menerjemahkan syarat kebutuhan sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat *coding*. Proses ini berfokus pada rancangan struktur data, arsitektur *software*, representasi *interface*, dan detail (algoritma) prosedural.

d. *Construction*

*Construction* merupakan proses membuat kode (*code generation*). *Coding* atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. *Programmer* akan menerjemahkan transaksi yang diminta oleh *user*. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu *software*, artinya penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut untuk kemudian bisa diperbaiki.

e. *Deployment*

Tahapan ini bisa dikatakan final dalam pembuatan sebuah *software* atau sistem. Setelah melakukan analisis, desain dan pengkodean maka sistem yang sudah jadi akan digunakan *user*. Kemudian *software* yang telah dibuat harus dilakukan pemeliharaan secara berkala.

## 2.6 Testing

*Testing* adalah proses pemantapan kepercayaan akan kinerja program atau sistem sebagaimana yang diharapkan. *Testing software* adalah proses

mengoperasikan *software* dalam suatu kondisi yang dikendalikan untuk verifikasi, mendeteksi *error* dan validasi. Verifikasi adalah pengecekan atau pengetesan entitas-entitas, termasuk *software*, untuk pemenuhan dan konsistensi dengan melakukan evaluasi hasil terhadap kebutuhan yang telah ditetapkan. Validasi adalah melihat kebenaran sistem apakah proses yang telah dituliskan sudah sesuai dengan apa yang dibutuhkan oleh pengguna. Deteksi *error* adalah testing yang berorientasi untuk membuat kesalahan secara intensif, untuk menentukan apakah suatu hal tersebut terjadi bilamana tidak seharusnya terjadi atau suatu hal tersebut tidak terjadi. *Test case* merupakan suatu tes yang dilakukan berdasarkan pada suatu inisialisasi, masukan, kondisi ataupun hasil yang telah ditentukan sebelumnya (Romeo, 2003). Adapun kegunaan dari *test case* ini, adalah sebagai berikut:

1. Untuk melakukan testing kesesuaian suatu komponen terhadap desain *White Box Testing*.
2. Untuk melakukan testing kesesuaian suatu komponen terhadap spesifikasi *Black Box Testing*.

### **2.6.1 White Box Testing**

*White box testing* adalah suatu metode desain *test case* yang menggunakan struktur kendali dari desain prosedural. Seringkali *white box testing* diasosiasikan dengan pengukuran cakupan tes, yang mengukur persentase jalur-jalur dari tipe yang dipilih untuk dieksekusi oleh *test cases*. *White box testing* dapat menjamin semua struktur *internal* data dapat dites untuk memastikan validitasnya (Romeo, 2003).

Cakupan pernyataan, cabang dan jalur adalah suatu teknik *white box testing* yang menggunakan alur logika dari program untuk membuat *test cases* alur logika adalah cara dimana suatu bagian dari program tertentu dieksekusi saat menjalankan program. Alur logika suatu program dapat direpresentasikan dengan *flow graph*.

### 2.6.2 *Black Box Testing*

*Black box testing* dilakukan tanpa adanya suatu pengetahuan tentang detail struktur internal dari sistem atau komponen yang dites, juga disebut sebagai *functional testing*. *Black box testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software* (Romeo, 2003).

Dengan adanya *black box testing*, perancang *software* dapat menggunakan kebutuhan fungsional pada suatu program. *Black box testing* dilakukan untuk melakukan pengecekan apakah sebuah *software* telah bebas dari *error* dan fungsi-fungsi yang diperlukan telah berjalan sesuai dengan yang diharapkan.

