

## BAB II

### LANDASAN TEORI

#### 2.1 Workflow System

*Workflow system* merupakan pengembangan dari sebuah perangkat lunak yang mengotomasi proses bisnis dengan menyediakan sebuah rangka kerja terstruktur untuk membantu sebuah proses (Chaffey, 1996).

*Workflow system* membantu pengaturan sebuah proses bisnis dengan memastikan bahwa sebuah tugas yang paling prioritas harus didahulukan:

1. Sesegera mungkin
2. Oleh orang yang tepat
3. Dengan perintah yang tepat

*Workflow* biasanya mengikutsertakan rekan kerja yang mengerjakan beberapa bagian dari keseluruhan proses, jadi bisa disebut ini merupakan bagian penting dalam berkolaborasi. *Workflow* bukanlah sebuah metode dalam suatu kelompok kerja, tetapi lebih kepada pengurangan waktu dan biaya dalam pelaksanaan proses bisnis dan memastikan bahwa tugas-tugas tetap berjalan secara konsisten untuk meningkatkan kualitas

Fungsi utama *workflow system* sebagai pengingat kepada para pekerja untuk menjalankan perintah atau tugas-tugas yang tepat di saat yang tepat. Hal ini dapat dicapai melalui rancangan *workflow* yang mengatur tiap-tiap tugas mana sajakah yang perlu dilakukan berdasarkan prioritas tugas dan siapa yang dapat menyelesaikannya.

Dalam penggunaannya, *Workflow System* dibagi menjadi 4 tipe yang sesuai untuk penyelesaian masalahnya. Berikut adalah tipe-tipe *Workflow System*:

1. *Administrative Workflow Systems*

Ini adalah bentuk umum yang berdasar *workflow system* yang menggunakan perangkat elektronik tersambung ke *e-mail*. Aplikasi administratif melakukan tugas-tugas rutin seperti persetujuan cuti atau proses pembelian. Meskipun rutin, hal ini dapat menghemat biaya yang banyak bagi perusahaan.

2. *Production Workflow Systems*

Sistem produksi merupakan suatu struktur tertinggi dalam *workflow system* yang menjelaskan aturan bisnis secara jelas dan diprioritaskan. Ini diperlukan sebab resiko penjualan diikutsertakan apabila tahapan-tahapan yang telah ditentukan tidak diikuti dibawah dalam melakukan misi bisnis yang kritis.

3. *Push and Pull Workflows*

Dua istilah lain yang sering digunakan untuk mendeskripsikan *workflow* adalah “*push*” dan “*pull*” *workflow*. Dalam model *push* pengguna secara otomatis memberikan tugas-tugas dengan perangkat lunak biasanya dari barisan *workflow*. Dengan model *pull* tanggung jawab berada di pengguna paling akhir untuk memilih tugas-tugas yang tersedia dalam kolom kerja dan mengurutkannya dalam urutan yang dirasa paling pas. Hal ini berarti jelaslah bahwa model *pull* begitu dekat berhubungan dengan aturan-aturan *workflow* dan model *push* untuk *workflow* yang terstruktur.

#### 4. *Object Oriented Workflow Systems*

Sistem ini paling sering digunakan untuk memecahkan masalah-masalah dalam sebuah bisnis. Sedikit sekali contoh model sistem object oriented yang ada, tetapi suatu saat sistem ini akan berkembang pesat.

### **2.2 Beasiswa Direktorat Jendral Pendidikan Tinggi (DIKTI)**

Pemerintah melalui Direktorat Jendral Pendidikan Tinggi Kementerian Pendidikan dan Kebudayaan berupaya mengalokasikan dana untuk memberikan bantuan biaya pendidikan kepada mahasiswa yang orang tuanya tidak mampu untuk membiayai pendidikannya, dan memberikan beasiswa kepada mahasiswa yang mempunyai prestasi tinggi, baik kurikuler maupun ekstrakurikuler. Agar program bantuan biaya pendidikan dan beasiswa dapat dilaksanakan sesuai dengan prinsip 3T, yaitu: Tepat Sasaran, Tepat Jumlah, dan Tepat Waktu, maka Direktorat Jendral Pendidikan Tinggi menerbitkan pedoman.

Dijelaskan lebih lanjut di dalam penjelasan, yang dimaksud dengan “beasiswa” adalah dukungan biaya Pendidikan yang diberikan kepada Mahasiswa untuk mengikuti dan/atau menyelesaikan Pendidikan Tinggi berdasarkan pertimbangan utama prestasi dan/atau potensi akademik. Sedangkan “bantuan biaya pendidikan” adalah dukungan biaya Pendidikan yang diberikan kepada Mahasiswa untuk mengikuti dan/atau menyelesaikan Pendidikan Tinggi berdasarkan pertimbangan utama keterbatasan kemampuan ekonomi.

Mengacu kepada Undang-undang dan Peraturan Pemerintah tersebut, maka Pemerintah melalui Direktorat Jenderal Pembelajaran dan Kemahasiswaan

(Direktorat Jenderal Pendidikan Tinggi), mengupayakan pemberian beasiswa bagi yang berprestasi dan bantuan biaya pendidikan bagi mahasiswa yang memiliki keterbatasan kemampuan ekonomi.

Dan berikut adalah peraturan atau undang-undang yang mendasari dibentuknya beasiswa bagi para peserta didik:

1. Undang-Undang Republik Indonesia Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional.
2. Undang-undang Republik Indonesia Nomor 12 Tahun 2012 tentang Pendidikan Tinggi.
3. Peraturan Pemerintah Nomor 48 Tahun 2008 tentang Pendanaan Pendidikan.
4. Peraturan Pemerintah Nomor 4 Tahun 2014 tentang Penyelenggaraan Pendidikan Tinggi dan Pengelolaan Perguruan Tinggi.
5. Peraturan Menteri Nomor 30 Tahun 2010 tentang pemberian bantuan biaya pendidikan kepada peserta didik yang orang tua atau walinya tidak mampu membiayai pendidikan.

### 2.3 Web dan PHP

*World Wide Web* (W3) atau yang dikenal juga dengan istilah *web* adalah suatu sistem yang berkaitan dengan dokumen digunakan sebagai media untuk menampilkan teks, gambar, multimedia dan lainnya pada jaringan tersebut (Alexander, 2014). Proses kerja saat pemanggilan halaman *web* terbagi menjadi dua bagian, yaitu proses pada sisi klien atau dikenal dengan istilah *Client Side* dan proses pada sisi *server* atau dikenal dengan istilah *Server Side*.

### 2.3.1 Client Side Scripting

Pada sisi klien, proses menjalankan suatu *script* seperti Javascript, VBScript, dll dapat berjalan dengan baik. *Script* yang dijalankan umumnya adalah pengolahan tampilan halaman *web* serta validasi *control input*. Melihat dari sisi kecepatan, proses sisi klien sangat membantu mengurangi lalu lintas data pada *server*, hal ini dikarenakan pada sisi klien proses menampilkan halaman *web* diolah menggunakan sumber daya dari komputer yang menjalankan halaman *web* tersebut.

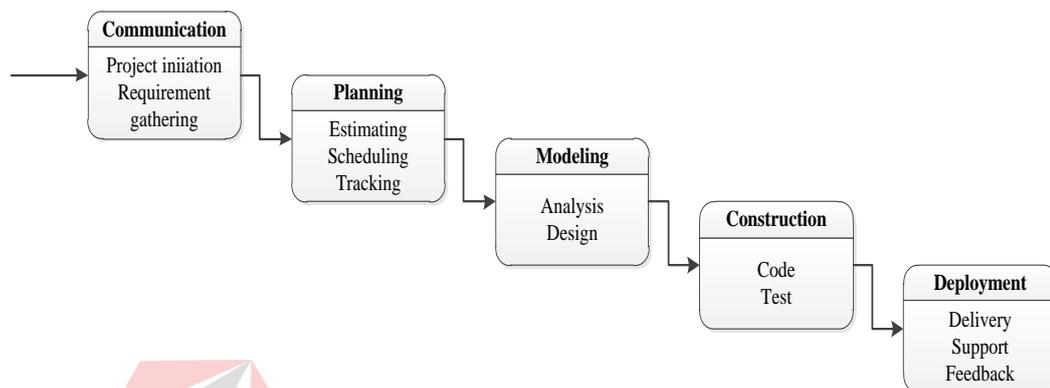
### 2.3.2 Server Side Scripting

Proses yang dijalankan pada sisi *server* memiliki tingkat pengolahan yang lebih kompleks seperti pengolahan data, hak akses dan pengolahan lainnya yang memerlukan ketelitian dan sumber daya yang besar. Beberapa bahasa pemrograman yang berjalan pada sisi *server* antara lain PHP, ASP, Java Servlet, JSP, ColdFusion, dll.

## 2.4 Metode SDLC Model Waterfall

Menurut Pressman (2015), *System Develoment Life Cycle* (SDLC) ini biasanya disebut juga dengan model *waterfall*. Menurut Pressman (2015), nama lain dari Model *Waterfall* adalah Model Air Terjun kadang dinamakan siklus hidup klasik (*classic life cyle*), 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.



Gambar 2.1 Model pengembangan *Waterfall* (Pressman, 2015)

Gambar 2.1 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 (2015) 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 ke tahap *Communication*, *Planning*, *Modeling*, *Construction*, dan *Deployment*.

Berikut ini adalah penjelasan dari tahap-tahap yang dilakukan di dalam Model *Waterfall* menurut Pressman (2015):

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 ke 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.

