

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Presensi dan Absensi Karyawan**

Menurut Dessler (2003), Manajemen Sumber Daya Manusia (MSDM) adalah suatu manajemen yang khusus mempelajari hubungan dan peranan manusia dalam organisasi perusahaan. Unsur MSDM adalah manusia yang merupakan tenaga kerja pada perusahaan, fokus yang dipelajari MSDM ini hanya masalah yang berhubungan dengan tenaga kerja manusia saja.

Manusia selalu berperan aktif dan dominan dalam setiap kegiatan organisasi, karena manusia menjadi perencanaan, perilaku, dan penentu terwujudnya tujuan organisasi. Tujuan tidak mungkin terwujud tanpa peran aktif karyawan meskipun alat-alat yang dimiliki perusahaan begitu canggihnya. Alat-alat canggih yang dimiliki perusahaan tidak ada manfaatnya bagi perusahaan jika, peran aktif karyawan tidak diikutsertakan. Mengatur karyawan adalah sulit dan kompleks, karena mereka mempunyai pikiran, perasaan status, keinginan, dan latar belakang yang *heterogen* yang dibawa ke dalam organisasi. Karyawan tidak dapat diatur dan dikuasai sepenuhnya seperti mengatur mesin, modal, atau gedung.

Menurut Bastian (2007), presensi karyawan adalah suatu kegiatan mendokumentasikan kehadiran karyawan di perusahaan, setiap hari kerja pegawai diharuskan melakukan presensi pada waktu datang dan pulang, dalam satu periode waktu. Jadi, presensi karyawan adalah suatu proses pencatatan data kehadiran karyawan mulai dari waktu masuk dan pulang karyawan pada periode waktu tertentu. Kantor perwakilan SKK Migas Surabaya dalam mencatat kehadiran

karyawan menggunakan *finger print*, dimana seorang karyawan harus melakukan *clock in* dan *clock out* saat hari kerja dan yang mengelola data kehadiran karyawan adalah bagian administrasi keuangan.

Nurachmad (2009), lembur digambarkan sebagai pekerjaan dengan tambahan waktu kerja reguler. Menurut undang-undang kebutuhan, waktu kerja reguler biasanya delapan jam sehari, dan 40 jam dalam seminggu. Waktu ini bervariasi disetiap negara-negara industri. Informasi ini dapat digunakan untuk mengkalkulasi tenaga kerja yang efisiensi dan lainnya, untuk mengkalkulasi tergantung pada pembayaran tambahan di luar tingkat tarif yang sesuai pada hukum

Berdasarkan KEP.102/MEN/VI/2004 tentang waktu lembur dan upah lembur, waktu lembur kerja adalah waktu kerja yang melebihi 7 (tujuh) jam sehari dan 40 (empat puluh) jam 1 (satu) minggu untuk 6 (enam) hari kerja dalam 1 (satu) minggu atau 8 (delapan) jam sehari, dan 40 (empat puluh) jam 1 (satu) minggu untuk 5 (lima) hari kerja dalam 1 (satu) minggu atau upah yang diterima pada hari istirahat mingguan.

Kondisi perusahaan saat ini terkait lembur adalah lembur dilihat dari waktu *clock out* dari karyawan, dapat dikatakan lembur jika karyawan melakukan *clock out* minimal 1 jam di atas jam kerja. Untuk karyawan OB dikatakan lembur jika datang satu jam lebih awal dari jam kerja kantor dan jika kerja di atas jam kerja kantor minimal satu jam karena lembur dihitung per 1 jam.

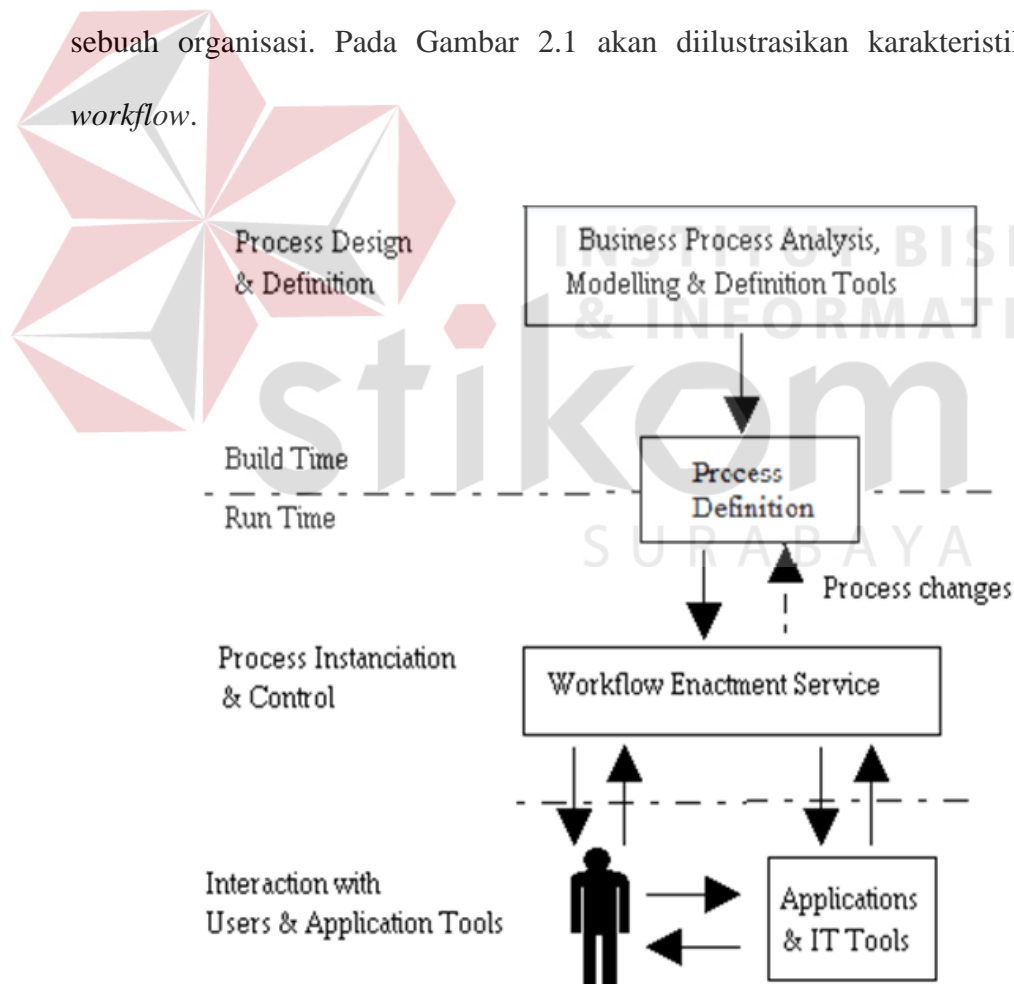
Menurut Hasibuan (2001), absensi adalah tidak bekerjanya seorang karyawan pada saat hari kerja, karena sakit, ijin, alpa, dan cuti. Jadi, absensi adalah ketidakhadiran seorang karyawan pada hari kerja dan pada suatu periode waktu

tertentu dikarenakan adanya suatu hal seperti sakit, ijin, alpa, cuti. Angka absensi karyawan memiliki pengaruh atas penilaian kinerja karyawan.

## 2.2 Workflow

Menurut Hollingsworth (1994), *workflow* adalah memfasilitasi secara terkomputerisasi atau membuat proses bisnis berjalan otomatis baik secara keseluruhan maupun setiap bagian. *Workflow* sering dihubungkan dengan perancangan kembali proses bisnis, dimana berkaitan dengan peninjauan, analisis pemodelan, pendefinisian dan implementasi pelaksanaan atas inti proses bisnis dari

sebuah organisasi. Pada Gambar 2.1 akan diilustrasikan karakteristik sistem *workflow*.



Gambar 2.1 Karakteristik Sistem *Workflow*

### 2.2.1 Fungsional Sistem *Workflow*

Sistem *workflow* mendukung 3 area fungsional, yaitu:

#### 1. *Build-time Functions*

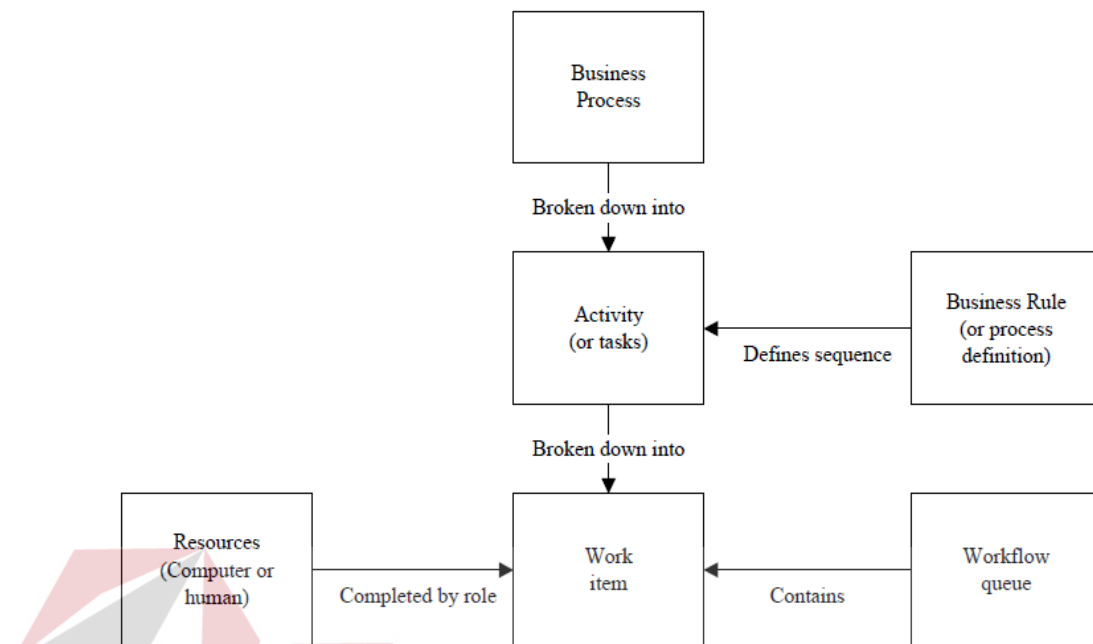
Fungsi ini menghasilkan sebuah pendefinisian proses bisnis yang terkomputerisasi. Selama fase ini, sebuah proses bisnis diterjemahkan dari aktivitas yang terjadi saat ini kedalam pendefinisian yang dapat diproses komputer dengan analisa, pemodelan, dan teknik sistem pendefinisian dengan hasil berupa model proses, pola proses, *metadata*, atau sebuah pendefinisian proses.

#### 2. *The Run-time Process Control Functions*

Menafsirkan melalui perangkat lunak dimana bertanggungjawab atas pembuatan dan kontrol contoh proses operasional, penjadwalan berbagai macam tahapan aktivitas dalam proses-proses dan melibatkan sumber daya manusia dan aplikasi IT secara tepat.

#### 3. *The Run-time Activity Interactions*

Berkaitan dengan kegiatan manusia, lebih sering dimunculkan dengan penggunaan alat IT tertentu seperti pengisian *form*, atau dengan kegiatan pemrosesan informasi yang membutuhkan program aplikasi tertentu untuk digunakan pada beberapa informasi yang telah ditentukan, seperti memperbarui *database* lama dengan data baru.

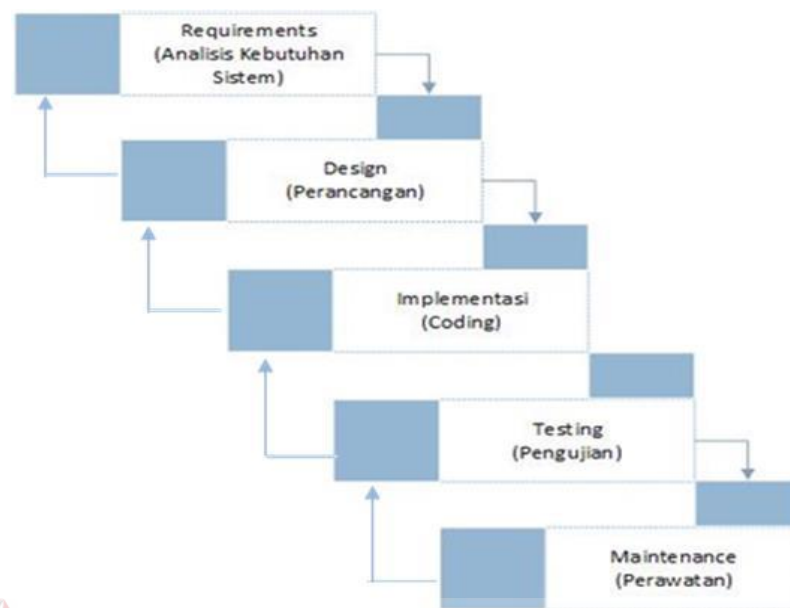


Gambar 2.2 Elemen Kunci Utama dalam *Workflow Systems* (Chaffey, 1998)

### 2.3 *System Development Life Cycle (SDLC)*

Menurut Pressman (2001), model *System Development Life Cycle (SDLC)* ini biasa disebut juga dengan model *waterfall* atau disebut juga *classic life cycle*. Adapun pengertian dari SDLC ini adalah suatu pendekatan yang sistematis dan berurutan. Tahapan-tahapannya adalah *Requirements* (analisis sistem), *Analysis* (analisis kebutuhan sistem), *Design* (perancangan), *Coding* (implementasi), *Testing* (pengujian) dan *Maintenance* (perawatan).

Model eksplisit pertama dari proses pengembangan perangkat lunak, berasal dari proses-proses rekayasa yang lain. Model ini memungkinkan proses pengembangan lebih terlihat. Hal ini dikarenakan bentuknya yang bertingkat ke bawah dari satu fase ke fase lainnya, model ini dikenal dengan model *waterfall*, dapat dilihat pada gambar berikut.



Gambar 2.3 *System Development Life Cycle (SDLC) Model Waterfall*

Sumber: Pressman (2001)

Penjelasan-penjelasan SDLC Model *Waterfall*, adalah sebagai berikut.

a. *Requirement* (Analisis Kebutuhan Sistem)

Pada tahap awal ini dilakukan analisa guna menggali kebutuhan yang akan dibutuhkan. Kebutuhan ada bermacam-macam seperti kebutuhan data dan kebutuhan *user* itu sendiri. Kebutuhan itu sendiri sebenarnya dibedakan menjadi tiga jenis kebutuhan. Pertama tentang kebutuhan teknologi. Dari hal ini dilakukan analisa mengenai kebutuhan teknologi yang diperlukan dalam pengembangan suatu sistem seperti data penyimpanan informasi atau *database*. Kedua kebutuhan informasi, contohnya seperti informasi mengenai visi dan misi perusahaan, sejarah perusahaan, latar belakang perusahaan. Dan yang ketiga yaitu kebutuhan *user*. Dalam hal ini dilakukan analisa terkait kebutuhan *user* dan kategori *user*. Dari analisa yang telah disebutkan di atas, terdapat satu hal yang tidak kalah penting dalam tahap analisa di metode SDLC, yaitu analisa

biaya dan resiko. Dalam tahap ini diperhitungkan biaya yang akan dikeluarkan seperti biaya implementasi, *testing* dan *maintenance*.

b. *Design* (Perancangan)

Selanjutnya, hasil analisa kebutuhan sistem tersebut akan dibuat sebuah *design database*, DFD, ERD, antarmuka pengguna atau *Graphical User Interface* dan jaringan yang dibutuhkan untuk sistem. Selain itu juga perlu merancang struktur data, arsitektur perangkat lunak, detil prosedur dan karakteristik tampilan yang akan disajikan. Proses ini menterjemahkan kebutuhan sistem ke dalam sebuah model perangkat lunak yang dapat diperkirakan kualitasnya sebelum memulai tahap implementasi.

c. *Implementation* (*Coding*)

Rancangan yang telah dibuat dalam tahap sebelumnya akan diterjemahkan ke dalam suatu bentuk atau bahasa yang dapat dibaca dan diterjemahkan oleh komputer untuk diolah. Tahap ini juga dapat disebut dengan tahap implementasi, yaitu tahap yang mengkonversi hasil perancangan sebelumnya ke dalam sebuah bahasa pemrograman yang dimengerti oleh komputer. Kemudian komputer akan menjalankan fungsi-fungsi yang telah didefinisikan sehingga mampu memberikan layanan-layanan kepada penggunanya.

d. *Testing* (Pengujian)

Pengujian program dilakukan untuk mengetahui kesesuaian sistem berjalan sesuai prosedur atau tidak dan memastikan sistem terhindar dari *error* yang terjadi. *Testing* juga dapat digunakan untuk memastikan kevalidan dalam proses *input*, sehingga dapat menghasilkan *output* yang sesuai. Pada tahap ini terdapat dua metode pengujian perangkat yang dapat digunakan, yaitu: metode

*black-box* dan *white-box*. Pengujian dengan metode *black-box* merupakan pengujian yang menekankan pada fungsionalitas dari sebuah perangkat lunak tanpa harus mengetahui bagaimana struktur di dalam perangkat lunak tersebut. Sebuah perangkat lunak yang diuji menggunakan metode *black-box* dikatakan berhasil jika fungsi-fungsi yang ada telah memenuhi spesifikasi kebutuhan yang telah dibuat sebelumnya. Pengujian dengan menggunakan metode *white-box* yaitu menguji struktur internal perangkat lunak dengan melakukan pengujian pada algoritma yang digunakan oleh perangkat lunak.

e. *Maintenance* (Perawatan)

Tahap terakhir dari metode SDLC ini adalah *maintenance*. Pada tahap ini jika sistem sudah sesuai dengan tujuan yang ditentukan dan dapat menyelesaikan masalah pada perusahaan, maka akan diberikan kepada pengguna. Setelah digunakan dalam periode tertentu pasti terdapat penyesuaian atau perubahan sesuai dengan keadaan yang diinginkan, sehingga membutuhkan perubahan terhadap sistem tersebut. Tahap ini dapat pula diartikan sebagai tahap penggunaan perangkat lunak yang disertai dengan perawatan dan perbaikan. Perawatan dan perbaikan suatu perangkat lunak diperlukan, termasuk didalamnya adalah pengembangan, karena dalam prakteknya ketika perangkat lunak digunakan terkadang masih terdapat kekurangan atau penambahan fitur-fitur baru yang dirasa perlu.