



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Pengertian Sumber Daya Manusia (*Human Resource*)**

*Human resource* adalah potensi yang merupakan asset dan berfungsi sebagai modal (non material/non finansial) di dalam organisasi bisnis, yang dapat diwujudkan menjadi potensi nyata (*real*) secara fisik non fisik dalam mewujudkan eksistensi organisasi (Nawawi, 2010).

#### **2.2 Manajemen Sumber Daya Manusia**

Manajemen sumber daya manusia adalah proses memperoleh, melatih, menilai, dan memberikan kompensasi kepada karyawan, memperhatikan hubungan kerja mereka, kesehatan, dan keamanan, serta masalah keadilan (Dessler, 2003).

Manajemen sumber daya manusia merupakan aktivitas-aktivitas yang didesain untuk menyediakan dan mengkoordinasikan sumber daya manusia dalam suatu organisasi (Byars, 2005).

Dapat disimpulkan bahwa manajemen sumber daya manusia adalah proses memperoleh, melatih, menilai, dan memberikan kompensasi kepada karyawan yang bertujuan untuk menyediakan dan mengkoordinasikan sumber daya manusia dalam suatu organisasi.

### 2.3 Pengertian *Enterprise Resource Planning* (ERP)

*Enterprise Resource Planning* adalah sistem yang mengadopsi sebuah proses bisnis terhadap keseluruhan organisasi untuk mengintegrasikan perencanaan, manajemen dan penggunaan terhadap sumber daya organisasi dan menggunakan *software platform* umum dan *database* (Rainer dan Cegielski, 2011).

### 2.4 Pengertian SAP

SAP ditemukan pada tahun 1972 dengan nama SAP (*Systems, Application and Products in Data Processing*) di Walldorf, Jerman. Pada mulanya SAP merupakan nama sebuah perusahaan sistem komputer. Sistem SAP terdiri dari modul-modul terintegrasi yang mencakup setiap aspek manajemen bisnis secara virtual (Leon, 2005).

SAP (*Systems, Applications, and Products in Data Processing*) merupakan sebuah perusahaan Jerman yang mengembangkan perangkat lunak bisnis yang berbasis ERP (*Enterprise Resource Planning*). ERP merupakan istilah yang digunakan untuk suatu perangkat lunak terintegrasi yang menggabungkan fungsi-fungsi bisnis utama dari sebuah perusahaan atau organisasi (SAP AG, 2006).

Dapat disimpulkan bahwa SAP merupakan sebuah aplikasi buatan perusahaan Jerman yang didirikan oleh 5 orang *system analysts* yang semuanya mantan pegawai IBM di Jerman, *software* ini berbasis ERP yang mengintegrasikan fungsi-fungsi bisnis utama dan SAP terdiri dari modul-modul yang mencakup setiap aspek manajemen bisnis.

Pada sekitar tahun 90an, SAP memiliki beberapa modul untuk sistem

ERPnya, yaitu:

1. FICO – *Finance and Controlling*
2. SD – *Sales & Distribution*
3. MM – *Materials Management*
4. PP – *Production Planning*
5. QM – *Quality Management*
6. SM – *Service Management*
7. WM – *Warehouse Management*
8. TM – *Transportation Management*
9. HR – *Human Resources*
10. PS – *Project Systems*

## **2.5 Pengertian *Human Resources* Modul**

*Human Resources* modul terdiri dari semua keperluan dalam proses bisnis untuk mengelola secara efisien semua kebutuhan dari sumber daya manusia yang ada di dalam perusahaan – dari penyaringan lamaran pekerjaan sampai pembayaran gaji dan pengembangan karyawan. Tujuan dari HR modul juga agar hanya sekali memasukkan data dan tersedia untuk aplikasi yang terkait (Hernandez, Keogh, dan Martinez, 2006).

Dalam *Human Resources* modul juga terdapat 4 grup modul, antara lain:

1. PA. *Personnel management* dan *personnel administration*, terdiri dari:

- PA-RC. *Recruitment*
- PA-BN. *Benefits administration*
- PA-CM. *Compensation management*
- PA-PM HR. *Funds and position management*
- PA-MA. *Managers desktop*
- *Personnel development (former PD)*

2. PT. *Personnel time management*, terdiri dari:

- *Work schedules*
- *Time evaluation*
- *Incentive wages*
- *Shift planning*
- *Time data recording*

3. PY. *Payroll*

4. PE. *Training and event management*

U  
M  
M  
N



**Gambar 2. 1 Proses *Human Capital Management***

**Sumber: SAP AG, 2006**

Elemen-elemen mySAP ERP HCM (SAP AG, 2006) , yaitu:

1. *Organizational Management Objects*

*Organizational Management* dapat dengan mudah dan secara efektif memetakan struktur organisasi (*organizational*) dan struktur laporan sesuai dengan objek organisasi yang relevan. Beberapa kunci yang digunakan dalam *Organizational Management*, yaitu:

- *Organizational Units*

*Organizational units* menggambarkan unit bisnis yang bervariasi dalam suatu organisasi.

- *Jobs*

*Jobs* merupakan deskripsi umum atau *template* yang berlaku untuk beberapa posisi dengan kebutuhan yang sesuai, tugas dan karakteristik.

- *Positions*

*Positions* merupakan unit organisasi terkecil dalam suatu struktur organisasi.

- *Persons*

*Persons* menempati posisi yang ada dalam struktur organisasi

## 2. *Applicant Data (Recruitment)*

Komponen *Applicant Data* dapat digunakan untuk menjalankan seluruh proses rekrutmen dari proses memasukkan data pelamar sampai penempatan posisi (SAP AG, 2006).

Proses dalam *recruitment* dapat melibatkan langkah-langkah sebagai berikut:

- Posisi yang kosong diumumkan dalam periklanan pekerjaan.
- Data pelamar dimasukkan ke dalam sistem (nama, alamat, kualifikasi, dan sebagainya)
- Dilakukan proses penyeleksian yang kompleks, menghasilkan pengambilan keputusan.
- Kontrak kerja dihasilkan atau dibuat untuk pelamar yang akan dipekerjakan
- Data pelamar dikirim ke HCM *Master Data*.

### 3. *Personnel Management*

Integrasi antara *Recruitment* dan *Personnel Administration* dapat memungkinkan data pelamar untuk dikirim sebagai data karyawan. Data karyawan disimpan dalam *mySAP* ERP HCM sebagai data *infotype*. Data tersebut dapat ditampilkan, diduplikasi, diperbaiki, dan dihapus (SAP AG, 2006).

*Infotype* yang ada dapat ditangani dengan beberapa cara, yaitu:

- *Single-screen Maintenance* (satu *infotype* untuk satu orang)
- *Personnel Action* (rangkaian *infotype* untuk satu orang)
- *Fast Entry* (satu *infotype* untuk satu orang)

Satu *infotype* yang merupakan pusat penting bagi data karyawan yaitu *Organizational Assignment infotype*. Dalam *infotype* ini, karyawan ditempatkan dalam 3 (tiga) struktur penting:

- *Enterprise Structure*
- *Personnel Structure*
- *Organization Plan*

### 4. *Personnel Development*

Dalam komponen *Personnel Development*, dapat direncanakan dan direalisasikan pengembangan personal dan memajukan atau meningkatkan pendidikan dan pelatihan untuk para karyawan, dengan mengintegrasikan *Personnel Development* dan *Training and Event Management* (SAP AG, 2006).



## 5. *Training and Event Management*

*Training and Event Management* merupakan komponen yang terintegrasi yang mendukung dalam perencanaan, pelaksanaan, dan penanganan rangkaian pelatihan dan kegiatan bisnis (SAP AG, 2006).

Terdapat 4 (empat) proses utama dalam *Training and Event Management*:

- *Business event preparation* mencakup semua data yang diperlukan ketika membuat *business event catalog*, seperti lokasi *training* yang akan diadakan.
- *Business event catalog* dihasilkan dan jadwal untuk kegiatan direncanakan dan dibuat dalam sistem.
- Ketika jadwal telah tersedia, kegiatan pemesanan *day-to-day*, pemesanan ulang, atau pembatalan kegiatan *business event* untuk peserta internal dan eksternal dapat dilaksanakan.
- Proses utama yang keempat meliputi kegiatan/aktivitas berulang (*recurring activities*) yang berhubungan dengan aktivitas *day-to-day*.

Ketika pelatihan telah dilakukan, peserta dapat menulis penilaian untuk kegiatan tersebut.

## 6. *Time Management*

Komponen *Time Management* mendukung semua proses yang berhubungan dengan perencanaan, pencatatan, dan pengevaluasian kehadiran dan ketidakhadiran karyawan (SAP AG, 2006).

Terdapat beberapa pilihan yang tersedia untuk pencatatan *time data* seperti jam kerja, liburan, *off-site work*, dan substitusi:

- Secara *online* oleh *time administrator*
- Menggunakan *front-end time recording system*.
- Menggunakan mySAP ERP *cross-application time sheet* (CATS).
- Menggunakan aplikasi *employee self-service* (ESS).

#### 7. *Appraisal*

Fungsi *Appraisal* dalam *Personnel Development* mendukung proses penilaian yang fleksibel yang dapat digunakan untuk melakukan standarisasi penilaian dengan level tujuan yang tinggi (SAP AG, 2006).

Tahapan dalam *Appraisal* dapat mencakup semua langkah umum dalam proses penilaian, seperti perencanaan, peninjauan, dan pelaksanaan, serta pengevaluasian penilaian yang diajukan.

#### 8. *Payroll*

mySAP ERP HCM mendukung fungsi *Payroll* dan kebutuhan pengaturan untuk lebih dari 50 negara (SAP AG, 2006).

Secara umum, *Payroll* berhubungan dengan perhitungan upah karyawan untuk pekerjaan yang karyawan selesaikan. Secara spesifik, *Payroll* mencakup beberapa proses seperti pembuatan hasil daftar gaji/upah dan pernyataan gaji, bank *transfer*, dan pembayaran cek.

#### 9. *Personnel Cost Planning*

*Personnel Cost Planning* mendukung proses bisnis di luar batasan departemen dan aplikasi. Informasi mengenai pembayaran dapat diperoleh

dari berbagai sumber untuk membuat *personnel cost plans*. Sebagai contoh, hasil dari *Payroll* dapat digunakan sebagai sumber data untuk *Personnel Cost Planning*. Rencana data pembayaran untuk posisi dan pekerjaan dalam *Organizational Management* juga dapat digunakan sebagai data bisnis (SAP AG, 2006).

## 2.6 Pengertian *Rapid Application Development* (RAD)

*Rapid Application Development* (RAD) adalah strategi siklus hidup yang ditujukan untuk menyediakan pengembangan yang jauh lebih cepat dan mendapatkan hasil dengan kualitas yang lebih baik dibandingkan dengan hasil yang dicapai melalui siklus tradisional (McLeod, 2002).

RAD merupakan gabungan dari bermacam-macam teknik terstruktur dengan teknik *prototyping* dan teknik pengembangan *joint application* untuk mempercepat pengembangan sistem/aplikasi (Bentley dan Whitten, 2004).

Dari definisi yang telah dikemukakan maka dapat disimpulkan bahwa dengan menggunakan RAD ini proses pengembangan aplikasi dapat dilakukan dengan waktu yang relatif lebih cepat.

Terdapat tiga fase dalam RAD (Kendall, 2010), yaitu:

### 1. *Requirement Planning Phase*

Dalam tahap ini, *users* dan *analyst* bertemu untuk mengidentifikasi tujuan-tujuan aplikasi atau sistem serta untuk mengidentifikasi syarat-syarat informasi yang ditimbulkan dari tujuan-tujuan tersebut.

## 2. RAD *Design Workshop*

Tahap ini adalah tahap dimana untuk merancang dan memperbaiki yang bisa digambarkan sebagai *workshop*. Selama *workshop* desain RAD, *users* merespon *prototype* yang ada dan *analysts* memperbaiki modul-modul yang dirancang berdasarkan respon dari *users*.

## 3. *Implementation*

Pada fase implementasi ini, penganalisis bekerja dengan para pengguna secara intens selama *workshop* dan merancang aspek-aspek bisnis dan nonteknis perusahaan. Segera setelah aspek-aspek ini disetujui dan sistem-sistem dibangun dan disaring, sistem-sistem baru atau bagian dari sistem diujicoba dan kemudian diperkenalkan kepada organisasi

## 2.7 **Pengertian *Waterfall***

*Waterfall* terbagi menjadi beberapa fase, masing-masing terstruktur dalam serangkaian kegiatan yang dapat dilakukan secara bersamaan, dan *input* dari sebuah fase merupakan hasil dari fase sebelumnya (Elisio, 2011). Terdapat 6 fase pada *waterfall* yaitu:

- *Feasibility Study*
- *Requirements Analysis and Specification*
- *Design and Specification*
- *Coding and Module Testing*
- *Integration and System Testing*
- *Delivery and Maintenance*

Fase pertama dimulai saat sebuah masalah muncul, langkah pertama adalah dengan memahami masalah dengan menganalisisnya dan menemukan solusi alternatif. Oleh karena itu, fase ini harus masuk ke dalam pengetahuan yang mendalam sehingga memakan banyak waktu (dan uang).

Fase kedua dilakukan setelah memahami masalah pada fase pertama. Pusat dari fase ini bukanlah cara yang harus dilakukan, apa yang harus diselesaikan. *Output* dari fase ini adalah dokumen *requirement specification* dimana semua *requirement* dinyatakan.

Fase ketiga dilakukan berdasarkan dari *requirement* yang sudah didefinisikan pada fase sebelumnya.

Fase keempat merupakan fase dimana *software* sebenarnya dibuat. Fase ini menerjemahkan hasil dari dokumen *design and specification* dari fase sebelumnya ke dalam bahasa pemrograman. Modul-modul yang telah dibuat harus dilakukan *test* secara individu sebelum diintegrasikan ke fase sebelumnya dan semua test harus didokumentasikan.

Modul yang dibuat pada fase sebelumnya dirakit dan terkadang termasuk ke dalam fase *coding*. Pada fase kelima, sistem harus di *test* pada *environment* yang seharusnya bekerja dan juga diintegrasikan dengan sistem dimana bekerja (*hardware, operative system, users, dll.*). *Output* dari fase ini harus sudah dapat digunakan.

Fase keenam, aplikasi harus siap digunakan sebelum fase ini dimulai, dan *delivery* biasanya dilakukan secara bertahap.

*Waterfall* menuntut disiplin, perencanaan, dan pengelolaan dimana *coding* dan *fixing* merupakan strategi utama.

## 2.8 Pengertian *Spiral*

*Spiral* dapat dilihat sebagai turunan dari metodologi evolusioner, dimana siklus perulangan didasarkan pada hasil dari hasil sebelumnya (Elisio, 2011).

Metodologi ini lebih baik dimengerti ketika di dalam diagram.

- *Determine objectives*
- *Identify and resolve risks*
- *Development and Test*
- *Plan the next iteration*

Fase pertama mengidentifikasi objektif dari masing-masing perulangan dan untuk mendefinisikan *requirements plan* untuk membangun sebuah *project*.

Fase kedua ditandai dengan implementasi dan analisis *prototype*. Setiap perulangan harus memberikan *prototype* yang dipimpin oleh analisis risiko yang telah dilakukan.

Fase ketiga bertanggung jawab untuk mengirimkan artefak dan untuk verifikasi dan validasinya. Konsep dari *requirement* di definisikan pada fase ini.

Fase keempat menghasilkan konsep operasi dari fase-fase sebelumnya. Dengan verifikasi dan validasi dari *requirement* maka *development plan* juga selesai pada fase ini.

## **2.9 Pengertian Sub Modul *Training and Event Management***

*Training and Event Management* memiliki berbagai macam fungsi yang kuat untuk memungkinkan merencanakan dan mengelola semua *business events* dari *training*, konvensi, sederhana dan efisiensi. Pelaporan dan penilaian fleksibel yang dapat membantu untuk memberikan keputusan penting agar dapat mendukung *feedback* untuk memastikan bahwa *business events* yang diselenggarakan memiliki kualitas dan efektivitas yang tinggi (SAP AG, 2002).

## **2.10 Pengertian Proses Bisnis**

Proses bisnis mengarah kedalam kegiatan yang saling berkaitan secara logis, sehingga perilaku organisasi tersebut dapat terkoordinasi dengan baik. Mengembangkan suatu produk yang baru, memenuhi pemesanan konsumen, membuat rencana pemasaran, dan pengangkatan karyawan baru adalah contoh kecil dalam proses proses bisnis (Laudon, 2012).

Proses bisnis terdiri dari banyak aktivitas yang akan mengkoordinasikan tehnik pengorganisasian sehingga dapat mencapai suatu tujuan perusahaan, yang mana dalam setiap prosesnya memiliki aturan-aturan tersendiri yang sudah tersusun, tetapi tidak menutup kemungkinan bahwa proses bisnis tersebut akan berpengaruh kepada proses bisnis yang lainnya juga (Weske, 2010).

Dapat disimpulkan bahwa pengertian proses bisnis adalah kegiatan atau proses yang terdiri dari banyak aktivitas logis yang terkoordinasi dengan baik sehingga mencapai suatu tujuan perusahaan.

## 2.11 Delapan Aturan Emas Perancangan *User Interface*

Ada delapan aturan emas dalam merancang *user interface* (Shneiderman, 2016), yaitu:

a. *Strive for consistency*

Urutan aksi yang konsisten akan dibutuhkan oleh situasi-situasi yang sama; istilah yang sama harus digunakan dalam *prompt*, *menu*, dan *help screen*; warna, *layout*, huruf capital, jenis huruf yang sama dan sebagainya harus digunakan secara menyeluruh. Perkecualian, pengulangan *password* atau konfirmasi perintah *delete* harus mudah dimengerti dan terbatas dalam jumlah.

b. *Seek universal usability*

Semenjak frekuensi penggunaan meningkat, demikian juga keinginan pengguna untuk mengurangi jumlah interaksi. Penyingkatan, *special key*, perintah tersembunyi, dan fasilitas makro dihargai oleh banyak *frequent user*.

c. *Offer informative feedback*

Untuk setiap aksi pengguna, seharusnya ada sistem untuk umpan balik. Untuk aksi yang kecil dan sering dilakukan, respon yang didapat bisa sederhana, sebaliknya untuk aksi yang besar dan jarang, respon seharusnya juga lebih banyak dan mendukung. Presentasi visual dari obyek menyediakan lingkungan yang nyaman untuk menunjukkan perubahan secara eksplisit.



d. *Design dialogs to yield closure*

Urutan tindakan harus diatur dalam kelompok yang dimulai dari awal, tengah, dan akhir. Umpan balik yang *informative* pada penyelesaian sekelompok tindakan memberikan kepuasan, rasa lega, dan sinyal untuk mempersiapkan ke kelompok tindakan berikutnya.

e. *Prevent errors*

Sistem yang kita rancang seharusnya merupakan sistem yang tidak memungkinkan pengguna untuk melakukan kesalahan yang serius. Jika pengguna melakukan kesalahan, seharusnya ada mekanisme yang menanganinya, seperti pemberian instruksi yang sederhana untuk pengembalian.

f. *Permit easy reversal of actions*

Tindakan dalam sistem seharusnya *reversible* (bisa kembali ke sebelumnya). Fungsi ini akan mengurangi rasa kecemasan karena pengguna mengetahui bahwa kesalahan bisa dibatalkan sehingga bisa mendorong user untuk mencoba menu-menu yang asing.

g. *Keep users in control*

Operator yang berpengalaman sangat menginginkan bahwa mereka mengendalikan sistem dan sistem tersebut memberikan respon terhadap aksi mereka. Aksi mereka yang tak terduga, rangkaian pengisian data yang membosankan, ketidakmampuan atau kesulitan dalam mendapatkan informasi yang dibutuhkan, dan ketidakmampuan untuk menghasilkan aksi yang diinginkan, semuanya menumbuhkan kegelisahan dan ketidakpuasan.



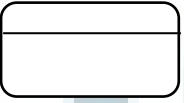
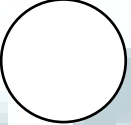
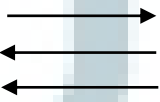
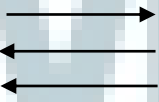

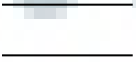
h. *Reduce short-term memory load*

Keterbatasan manusia memproses informasi dalam ingatan jangka pendek membutuhkan tampilan yang sederhana, banyak halaman digabungkan, urutan tindakan, dan penyediaan informasi yang diperlukan.

**2.12 Pengertian Data Flow Diagram (DFD)**

*Data flow diagram* merupakan suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data tersimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data tersimpan dan proses yang dikenakan pada data tersebut (Kristianto, 2003).

**Tabel 2. 1 Tabel Data Flow Diagram (DFD)**

Notasi Gane & Sarson	Notasi Yourdon / De Marco	Keterangan
		Entitas eksternal, dapat berupa orang / unit terkait yang berinteraksi dengan sistem tetapi di luar sistem.
		Orang, unit yang mempergunakan atau melakukan transformasi data. Komponen fisik tidak diidentifikasi.
		Aliran data dengan arah khusus dari sumber ke tujuan.
		Penyimpanan data atau tempat data direfer oleh proses.

*Data flow diagram* (DFD) terbagi menjadi 3 level (Muslihudin, 2006),

antara lain:

1. Diagram Konteks: menggambarkan satu lingkaran besar yang dapat mewakili keseluruhan proses yang terdapat dalam suatu sistem. Diagram ini merupakan tingkat tertinggi dan biasanya di beri nomor 0.
2. Diagram nol (Diagram Level-1): merupakan satu lingkaran besar yang mewakili lingkaran kecil yang ada di dalamnya. Merupakan pemecahan dari diagram konteks ke diagram nol dan di diagram ini memuat penyimpanan data.
3. Diagram rinci: merupakan diagram yang menguraikan proses apa yang ada di diagram nol.

### **2.13 UAT (*User Acceptance Testing*)**

UAT (*User Acceptance Testing*) merupakan pengujian yang dilakukan oleh *end user* dimana *user* tersebut adalah *staff* karyawan perusahaan yang langsung berinteraksi dengan sistem dan dilakukan verifikasi apakah fungsi yang ada telah berjalan sesuai dengan kebutuhan atau fungsinya (Perry, 2006).

### **2.14 Teknik Pengumpulan Data**

#### **2.14.1 Pengertian Wawancara**

Wawancara digunakan sebagai teknik pengumpulan data, apabila peneliti ingin melakukan studi pendahuluan untuk menemukan permasalahan yang harus diteliti, dan juga apabila

peneliti ingin mengetahui hal-hal dari responden yang lebih mendalam dan jumlah respondennya sedikit/kecil (Sugiyono, 2004).

Wawancara dapat dilakukan secara terstruktur maupun tidak terstruktur, dan dapat dilakukan melalui tatap muka (*face to face*) maupun dengan menggunakan telepon. Wawancara terdiri dari:

#### 1. Wawancara Terstruktur

Wawancara terstruktur digunakan sebagai teknik pengumpulan data, bagi peneliti atau pengumpul data telah mengetahui dengan pasti tentang informasi apa yang akan diperoleh.

#### 2. Wawancara Tidak Terstruktur

Wawancara tidak terstruktur, adalah wawancara yang bebas dimana peneliti tidak menggunakan pedoman wawancara yang telah tersusun secara sistematis dan lengkap untuk pengumpulan datanya. Pedoman wawancara yang digunakan hanya berupa garis-garis besar permasalahan yang akan dinyatakan.