

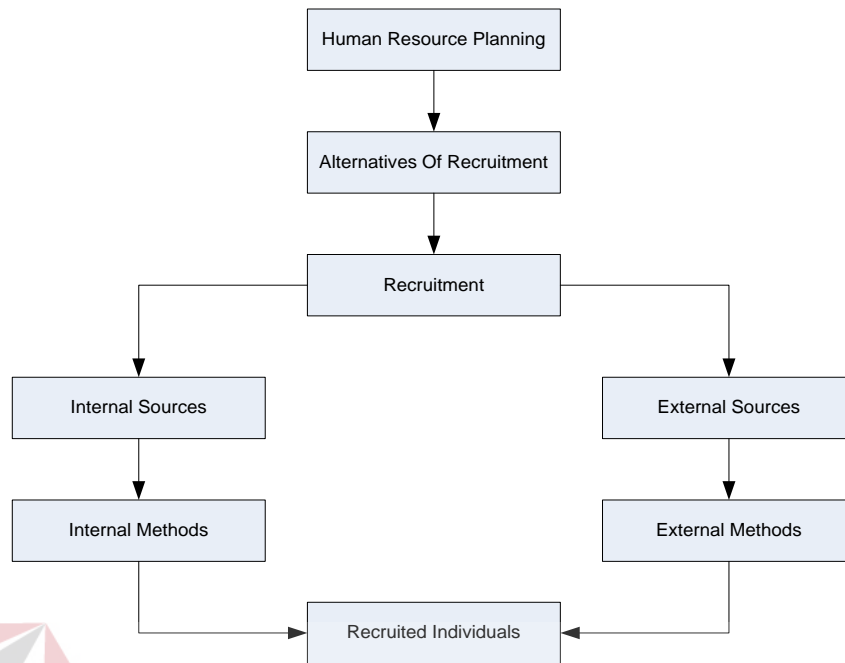
BAB II

LANDASAN TEORI

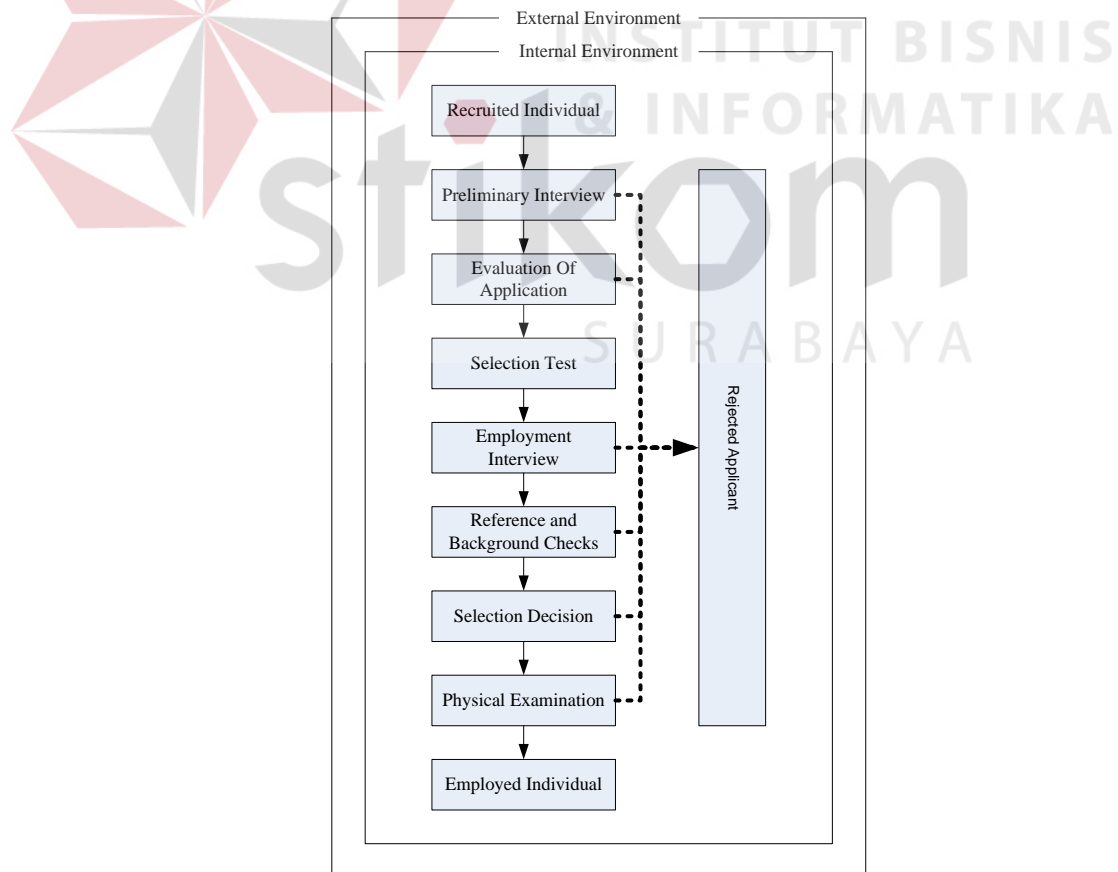
2.1 Rekrutmen dan Seleksi

Rekrutmen adalah sebuah proses menarik minat individu dalam sebuah jangka waktu tertentu, dalam jumlah yang sesuai dan dengan kualifikasi yang sesuai, dan meyakinkan mereka untuk melamar kerja pada sebuah organisasi. Proses rekrutmen dipicu ketika seorang manajer menyerahkan dokumen permintaan karyawan, yaitu sebuah dokumen yang menspesifikasikan nama pekerjaan, nama bagian, tanggal kapan karyawan dibutuhkan, dan informasi lain. Dengan informasi ini, manajer sumber daya manusia dapat mengacu deskripsi kerja yang sesuai untuk menentukan kualifikasi dari orang yang akan direkrut. Proses rekrutmen juga bisa dipicu setelah dilakukan perencanaan sumber daya manusia. Proses rekrutmen dimulai dengan menentukan sumber dari sumber daya manusia (SDM) yang akan direkrut, lalu menentukan metode perekrutan SDM, yang akan menghasilkan pelamar pekerjaan pada perusahaan. Untuk gambaran proses rekrutmen dapat dilihat pada Gambar 2.1 (Mondy dan Noe, 1990).

Seleksi adalah sebuah proses untuk memilih orang terbaik dari sejumlah pelamar, yang cocok untuk sebuah posisi. Proses seleksi dimulai ketika pelamar pekerjaan sudah terkumpul melalui proses rekrutmen. Pelamar akan melakukan serangkaian tes dan wawancara sebelum akhirnya perusahaan menentukan pelamar mana yang akan diterima bekerja sebagai karyawan. Gambaran proses seleksi dapat dilihat pada Gambar 2.2 (Mondy dan Noe, 1990).



Gambar 2.1 Proses Rekrutmen (Mondy dan Noe, 1990)



Gambar 2.2 Proses Seleksi (Mondy dan Noe, 1990)

2.2 *Workflow Management System (WfMS)*

Menurut Chaffey (1963), *Workflow Management System* adalah 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.

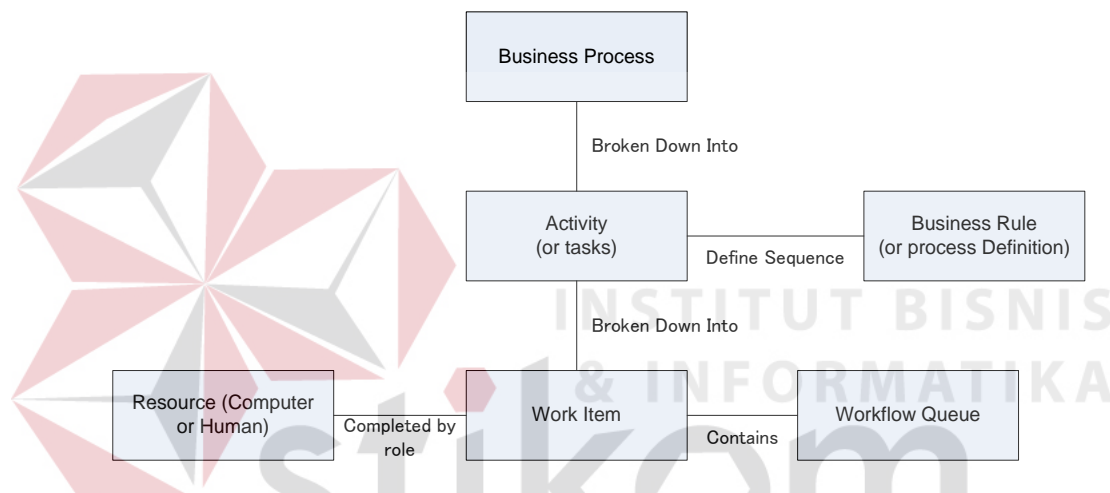
Workflow Management Coalition (WfMC) menggambarkan *workflow* sebagai fasilitasi komputerisasi atau otomatisasi proses bisnis secara keseluruhan atau sebagian, sedangkan WfMS digambarkan sebagai sebuah sistem yang mendefinisikan, menciptakan, dan mengelola pelaksanaan *workflow* melalui penggunaan perangkat lunak, yang berjalan pada satu atau lebih *workflow engine*, yang mampu menafsirkan definisi proses, berinteraksi dengan peserta *workflow*, dan jika diperlukan, meminta penggunaan alat dan aplikasi teknologi informasi (Chaffey, 1963).

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 mereka melakukan tugas-tugasnya. Keuntungan bisnis utama dari penerapan sebuah *workflow system* adalah waktu penyelesaian dan biaya dari proses bisnis yang ada sekarang dapat dikurangi.

2.2.1 *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 2.3 akan dijelaskan bagaimana sebuah kegiatan dipecah menjadi *workitem* individu yang harus diselesaikan. Setiap *workitem* dilakukan oleh sebuah *resource*, baik perangkat lunak, perangkat keras, atau seorang personil yang memiliki tanggung jawab untuk melakukannya. *Workitem* 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.



Gambar 2.3 Elemen Kerja Kunci dalam Sistem *Workflow* (Chaffey, 1963)

a. *Process Elements (Work Activities or 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.

b. *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 partisipan *workflow* diberikan 1 (satu) atau beberapa peran (*role*) 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.

c. *Dependencies and Business Rules*

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

d. *Workflow Queue*

Workflow system 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.

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

f. *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.

2.2.2 *Administrative Workflow System*

Administrative workflow system adalah sebuah sistem *workflow* umum, yang memanfaatkan penggunaan formulir elektronik yang terhubung dengan email. Sistem ini biasa diaplikasikan ke dalam tugas-tugas administrasi rutin seperti persetujuan pengajuan cuti, pemrosesan pemesanan pembelian, dll. *The Gartner Group* memperkirakan bahwa 83% dari semua dokumen bisnis di Amerika Serikat adalah dokumen formulir dengan biaya pembelian tahunan sebesar 6-8 milyar dolar Amerika dan biaya pemrosesan mencapai 360 milyar dolar Amerika. Formulir-formulir kertas ini menjadi target dari 1995 *Paper Reduction Act* (Chaffey, 1963).

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 koordinasi pengolahan formulir yang sekarang ditangani oleh logika bisnis yang dibangun ke dalam aplikasi (Chaffey, 1963).

2.3 Managing Employee Information from Hire to Retire.

Organisasi sumber daya manusia (SDM) membutuhkan akses ke semua dokumen karyawan yang berkaitan dari satu titik akses yang aman, daripada mencarinya di beberapa sistem dan area penyimpanan fisik di seluruh bagian perusahaan. Apakah dokumen yang dimaksud adalah resume yang baru diserahkan atau yang telah diarsipkan, perjanjian karyawan dan buku panduan, dokumentasi penggajian, panduan kebijakan atau dokumen lain yang berhubungan dengan karyawan, informasi tersebut harus mudah diakses, sambil dikelola dengan aman.

Kebanyakan organisasi SDM “terpendam” di dalam kertas, dan perusahaan yang bergantung pada dokumen kertas sebagai sumber informasi karyawan dapat membayar harga yang sangat mahal untuk melangsungkan lingkungan SDM yang menggunakan dokumen cetak. Mengumpulkan, mengolah, menyimpan, dan menunjukkan informasi karyawan secara manual dapat menciptakan risiko yang cukup besar, inefisiensi dan biaya, administrasi yang berlebihan dan tidak perlu, risiko keamanan, biaya dan kebutuhan penyimpanan, dokumen yang hilang, biaya-biaya penyalinan dan pengiriman, dan masalah-masalah lainnya yang bisa menjadi beban yang tidak perlu bagi departemen SDM ketika solusi dan proses yang ada dapat mempersingkat operasi SDM.

Pengolahan dokumen kertas karyawan membutuhkan sumber daya yang besar untuk menyimpan dan mencari dokumen, dan meningkatkan kemungkinan untuk salah menaruh dokumen penting. Kertas juga memperkenalkan risiko memamerkan informasi rahasia kepada individu yang tidak berwenang. Ketika informasi karyawan berada dalam bentuk dokumen cetak, dua atau lebih orang

tidak dapat mengakses informasi yang sama secara bersamaan, tanpa membuat salinan. Proses SDM yang berkuat pada dokumen berbentuk kertas berpotensi mengganggu penyelesaian transaksi karyawan, dan melibatkan manajemen pencatatan yang lebih tinggi dan biaya pengiriman.

Selain dokumen bagi karyawan saat ini, kebanyakan organisasi SDM mempertahankan dokumen yang berkaitan dengan bekas karyawan dalam jumlah yang lebih besar. Perusahaan sering menyimpan dokumen karyawan lebih lama dari yang diperlukan secara hukum, lebih lanjut memaparkan dokumen-dokumen tersebut kepada risiko potensial, penyimpanan berkelanjutan, dan biaya pemeliharaan.

Proses manual dapat memakan waktu yang berlebihan untuk menyelesaikannya, yang dapat menyebabkan penundaan lebih lanjut dalam proses lain yang bergantung pada penyelesaian proses manual SDM.

Transformasi dokumen kertas menjadi dokumen digital merupakan sebuah kemajuan besar dalam mengurangi biaya dan membawa efisiensi ke organisasi SDM. Efisiensi lebih lanjut dapat dicapai dengan mengganti formulir kertas dengan formulir elektronik yang secara otomatis menangkap data terstruktur dan tidak terstruktur, memulai atau memungkinkan partisipasi dalam proses bisnis, dan melalui integrasi yang cermat, bahkan dapat digunakan untuk memperbarui informasi yang disimpan dalam basis data dan sistem *Enterprise Resource Planning* (ERP). Informasi karyawan yang ditangkap dan disimpan secara elektronik menghilangkan kebutuhan untuk kembali memasukkan data, mengurangi kesalahan data, memungkinkan akses bersamaan oleh beberapa orang, menghilangkan pengarsipan yang salah, dan sangat mengurangi kebutuhan untuk

menyalin, mencetak dan menyuratkan dokumen SDM (IBM *Software Group*, 2008).

2.4 Perangkat Lunak

Perangkat lunak (*software*) adalah (1) Perintah (program komputer) yang bila dieksekusi, memberikan fungsi dan unjuk kerja seperti yang diinginkan. (2) Struktur data yang memungkinkan program memanipulasi informasi secara proporsional, dan (3) Dokumen yang menggambarkan operasi dan kegunaan program (Pressman, 2002). Perangkat lunak tidak hanya program komputer saja, tetapi juga semua dokumentasi terkait dan data konfigurasi yang diperlukan untuk membuat program tersebut beroperasi dengan benar (Sommerville, 2001).

Menurut Pressman (2002), perangkat lunak lebih merupakan elemen logika dan bukan merupakan elemen sistem fisik. Dengan demikian, perangkat lunak memiliki ciri yang berbeda dari perangkat keras:

1. Perangkat lunak dibangun dan dikembangkan, tidak dibuat dalam bentuk yang klasik.
2. Perangkat lunak tidak pernah usang.
3. Sebagian besar perangkat lunak dibuat secara *custom built*, serta tidak dapat dirakit dari komponen yang sudah ada.

Terdapat dua tipe dari produk perangkat lunak (Sommerville, 2001), yaitu:

1. *Generic Software*

Perangkat lunak generik adalah perangkat lunak mandiri (*stand alone*) yang diproduksi oleh sebuah perusahaan pengembangan perangkat lunak dan dijual di pasaran secara bebas.

2. *Custom Software*

Custom software (atau yang dipesan terlebih dahulu) adalah perangkat lunak yang dipesan oleh seorang pembeli tertentu. Perangkat lunak ini dikembangkan khusus oleh kontraktor perangkat lunak untuk pembeli tersebut.

Menurut Jogiyanto (2003), terdapat dua macam perangkat lunak:

1. *System Software*

Perangkat lunak yang mengoperasikan sistem komputer. Dapat dikelompokkan lagi dalam 3 bagian, yaitu: *Operating System*, *Utility*, dan *Language*

2. *Application Software*

Program yang ditulis dan diterjemahkan oleh *language software* untuk menyelesaikan aplikasi tertentu.

2.4.1 Rekayasa Perangkat Lunak

Fritz Bauer (Pressman, 2002) memberikan definisi pada konferensi seminar masalah pengembangan perangkat lunak: Rekayasa perangkat lunak adalah pengembangan dan penggunaan prinsip pengembangan suara untuk memperoleh perangkat lunak secara ekonomis yang handal dan bekerja secara efisien pada mesin nyata. Berdasarkan definisi ini, IEEE (Pressman, 2002) telah mengembangkan definisi yang lebih komprehensif, yaitu rekayasa perangkat lunak adalah: (1) Aplikasi dari sebuah pendekatan kuantitatif, disiplin, dan sistematis kepada pengembangan, operasi, dan pemeliharaan perangkat lunak; yaitu aplikasi dari rekayasa perangkat lunak. (2) Studi tentang pendekatan-pendekatan seperti pada (1).

Rekayasa perangkat lunak adalah suatu disiplin rekayasa yang berkaitan dengan semua aspek produksi perangkat lunak dari tahap awal spesifikasi sistem, sampai dengan pemeliharaan sistem setelah sistem tersebut dipakai (Sommerville, 2001). Rekayasa merupakan analisis, desain, konstruksi, verifikasi, dan manajemen kesatuan teknik (atau sosial). Tanpa memperdulikan kesatuan yang dikembangkan, terdapat beberapa pertanyaan yang harus dimunculkan dan dijawab (Pressman, 2002), yaitu:

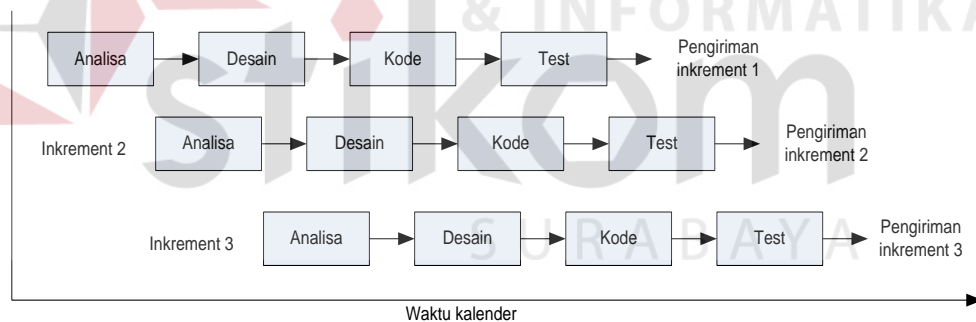
- a. Masalah apa yang akan dipecahkan?
- b. Karakteristik kesatuan apakah yang dipakai untuk menyelesaikan masalah tersebut?
- c. Bagaimanakah kesatuan (dan pemecahan tersebut) diadakan?
- d. Bagaimanakah kesatuan tersebut dibangun?
- e. Pendekatan apakah yang akan dipakai untuk menemukan kesalahan-kesalahan yang dibuat di dalam desain dan konstruksi dari kesatuan tersebut?
- f. Bagaimanakah kesatuan tersebut ditopang selama proses adaptasi yang lama pada saat koreksi, serta ketika perbaikan dibutuhkan oleh para pemakai kesatuan tersebut?

2.4.2 Incremental Model

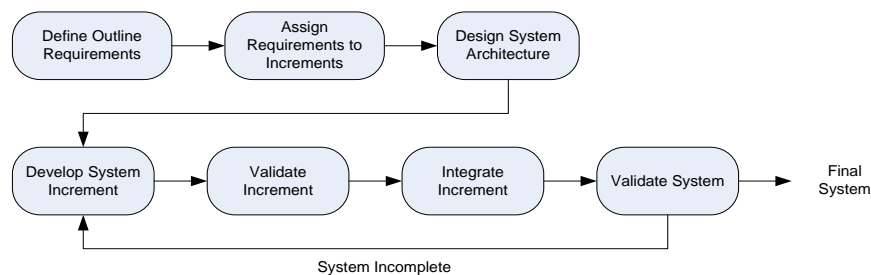
Incremental Model (model inkremental) adalah model dalam rekayasa perangkat lunak yang menggabungkan elemen-elemen model sekuensial linier (diaplikasikan secara berulang) dengan filosofi purwarupa iteratif (Pressman, 2002). Model inkremental dalam pengembangan adalah sebuah sarana untuk mengurangi pengerjaan ulang dalam proses pengembangan perangkat lunak dan memberikan beberapa peluang bagi pelanggan untuk menunda keputusan tentang

detil kebutuhan pelanggan tersebut sampai pelanggan tersebut mendapatkan pengalaman dengan sistem yang sedang dikembangkan (Sommerville, 2001).

Pada saat model inkremental dipergunakan, peningkatan pertama sering merupakan produk inti (*core product*), yaitu sebuah model peningkatan yang dipergunakan, tetapi beberapa antarmuka tambahan (beberapa diketahui dan beberapa tidak) tetap tidak disampaikan. Produk inti tersebut dipergunakan oleh pelanggan (atau mengalami pengkajian detil). Sebagai hasil dari pemakaian dan/atau evaluasi, maka dikembangkan rencana untuk peningkatan selanjutnya. Rencana tersebut menekankan modifikasi produk inti untuk secara lebih baik memenuhi kebutuhan para pelanggan dan penyampaian fitur serta fungsionalitas tambahan. Proses ini diulangi mengikuti penyampaian setiap peningkatan sampai bisa menghasilkan produk yang lengkap (Pressman, 2002).



Gambar 2.4 Alur Model Inkremental (Pressman, 2002)



Gambar 2.5 Alur Model Inkremental (Sommerville, 2001)

Model inkremental memiliki beberapa keuntungan (Sommerville, 2001):

- a. Pelanggan tidak harus menunggu sampai seluruh sistem diselesaikan dan pelanggan bisa mendapatkan nilai dari sistem tersebut. Peningkatan pertama memenuhi kebutuhan mereka yang paling kritis sehingga perangkat lunak dapat segera digunakan.
- b. Pelanggan dapat menggunakan peningkatan awal sebagai purwarupa dan mendapatkan pengalaman yang menyebutkan persyaratan untuk peningkatan sistem kemudian.
- c. Ada risiko yang lebih rendah dari kegagalan proyek secara keseluruhan. Meskipun masalah mungkin ditemui di beberapa tahapan peningkatan, ada kemungkinan bahwa beberapa akan berhasil dikirim ke pelanggan.
- d. Layanan berprioritas tertinggi diserahkan pertama kali dan kemudian tahapan peningkatan selanjutnya diintegrasikan dengan peningkatan sebelumnya, tidak dapat dihindari bahwa layanan sistem yang paling penting menerima paling banyak pengujian. Ini berarti bahwa pelanggan cenderung tidak mengalami kegagalan perangkat lunak di bagian yang paling penting dari sistem.

Model inkremental juga memiliki kekurangan (Sommerville, 2001):

- a. Tahapan peningkatan harus relatif kecil (tidak lebih dari 20.000 baris kode) dan masing-masing tahapan peningkatan harus memberikan beberapa fungsi sistem. Oleh karenanya, bisa sulit untuk memetakan kebutuhan pelanggan menjadi tahapan peningkatan dengan ukuran yang tepat.
- b. Kebanyakan sistem memerlukan seperangkat fasilitas dasar yang digunakan oleh berbagai bagian dari sistem tersebut. sebagaimana persyaratan tidak

didefinisikan secara rinci sampai tahapan peningkatan akan diimplementasikan, sulit untuk mengetahui fasilitas umum yang dibutuhkan semua tahapan peningkatan.

