Nama : Risnanda Hasna Mufida

Kelas : 6A

NIM : 18090092

Soal !

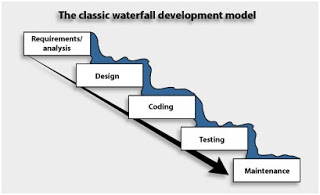
Silahkan teman-teman baca dan cari masing-masing karakteristik dari model proses pengembangan PL yang ada. setelah itu bandingkan perbedaan masing-masing model proses tersebut (model waterfall, model pengembangan evolusioner, model pengembangan formal, model pengembangan pemakaian ulang, model RAD)

Jawab !

Ada beberapa model proses perangkat lunak yang biasa digunakan sesuai dengan kondisi-kondisi tertentu

1. **Model Waterfall**

Model ini adalah model klasik yang mengusung pengembangan perangkat lunak yang sistematis, berurutan/sekuensial dimulai pada tingkat dan kemajuan system pada seluruh persyaratan dalam analisis, perancangan (desain), pengkodean, pengujian (testing), hingga ke tahap pemeliharaan dalam membangun software (perangkat lunak). Berikut ini gambaran dari Linear Sequential Model / waterfall model.



Gambar 1. The linear sequential model

Pada setiap tahapan dianalogikan bak air yang mengalir dari tempat tinggi ke tempat yang lebih rendah, artinya sebuah proses baru bias dilanjutkan setelah satu tahap awal selesai dengan sempurna.

Penjelasan tentang setiap tahapan dapat diringkas sebagai berikut:

*Tahap analisis:* pada tahap ini berlangsung proses pengumpulan kebutuhan secara lengkap untuk dianalisis dan didefinisikan kebutuhan apa saja yang harus dipenuhi oleh program yang akan dibuat, seperti memahami domain permasalahan, tingkah laku, unjuk kerja dan interface (antar muka).

*Tahap desain:* proses ini melibatkan empat atribut sebuah program yaitu struktur data, arsitektur, perangkat lunak, representasi interface, dan detail (algoritma) prosedural.

*Tahap pengkodean:* proses penterjemahan desain ke dalam bentuk bahasa mesin yang dapat dilakukan secara mekanis.

*Tahap pengujian:* proses ini dikerjakan setelah kode dirancang dan difokuskan pada fungsi dan jumlah kesalahan untuk diperbaiki.

*Tahap pemeliharaan:* meliputi penyesuaian atau perubahan yang berkembang seiring dengan adaptasi perangkat lunak dengan kondisi atau situasi sebenarnya setelah disampaikan kepada konsumen atau pelanggan.

Kelebihan metode ini antara lain mudah diaplikasikan karena urutan-urutan pengerjaan sudah sering dipakai; selain itu juga cocok untuk software berskala besar dan yang bersifat umum; yang paling penting, karena langkah-langkahnya sangat sekuensial, pengerjaan proyek akan mudah dikontrol dan terjadwal dengan baik.

Namun, terdapat pula beberapa kelemahan yang menjadi kekurangan dari metode waterfall ini, seperti kurang fleksibel, dikarenakan rincian prosesnya harus benar-benar jelas dan tidak boleh diubah-ubah. Apabila dikerjakan dengan melampaui tahap yang seharusnya maka proses desain yang sebelumnya itu akan berubah total dan memakan waktu yang banyak jika harus mengulang proses.

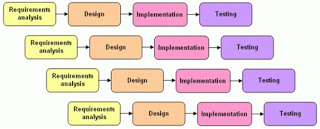
Model waterfal ini sangat sesuai digunakan dalam pengembangan sistem perangkat lunak dan hardware yang luas dan apabila kebutuhan pengguna telah dimengerti dengan baik. Selain itu, juga apabila waktu yang tersedia juga masih cukup banyak.

1. **Model Pengembangan Evolusioner**

Evolutionary software process model, terbagi dua:

**A. Incremental Model**

Model ini merupakan hasil kombinasi elemen-elemen dari model waterfall yang diaplikasikan secara berulang. Elemen-elemen tersebut dikerjakan hingga menghasilkan produk dengan spesifikasi tertentu kemudian proses dimulai dari awal kembali hingga muncul hasil yang spesifikasinya lebih lengkap dari sebelumnya dan tentunya memenuhi kebutuhan pemakai.



Gambar 4. Incremental Model

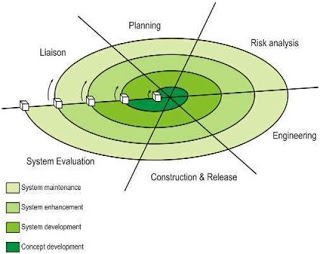
Kelebihan model ini adalah mampu mengakomodasi perubahan secara fleksibel, dengan waktu yang relatif singkat dan tidak dibutuhkan anggota/tim yang banyak untuk menjalankannya.

Kekurangannya adalah tidak cocok untuk proyek berukuran besar (lebih dari 200.000 baris coding) dan sulit untuk memetakan kebutuhan pemakai ke dalam rencana spesifikasi tiap-tiap hasil dari increament.

Model ini cocok dipakai untuk proyek kecil dengan anggota tim yang sedikit dan ketersediaan waktu yang terbatas.

**B. Spiral Model/Spiral Boehm**

Model ini mengadaptasi dua model perangkat lunak yang ada yaitu model prototyping dengan pengulangannya dan model waterfall dengan pengendalian dan sistematikanya. Model ini dikenal dengan sebutan Spiral Boehm. Pengembang dalam model ini memadupadankan beberapa model umum tersebut untuk menghasilkan produk khusus atau untuk menjawab persoalan-persoalan tertentu selama proses pengerjaan proyek.



Gambar 5. Model Spiral Boehm

*Tahap-tahap model ini dapat dijelaskan secara ringkas sebagai berikut.*

*Tahap Liason:* pada tahap ini dibangun komunikasi yang baik dengan calon pengguna/pemakai

*Tahap Planning (perencanaan):* pada tahap ini ditentukan sumber-sumber informasi, batas waktu dan informasi-informasi yang dapat menjelaskan proyek.

*Tahap Analisis Resiko:* mendefinisikan resiko, menentukan apa saja yang menjadi resiko baik teknis maupun manajemen.

*Tahap Rekayasa (engineering):* pembuatan prototipe

*Tahap Konstruksi dan Pelepasan (release):* pada tahap ini dilakukan pembangunan perangkat lunak yang dimaksud, diuji, diinstal dan diberikan sokongan-sokongan tambahan untuk keberhasilan proyek.

*Tahap Evaluasi:* Pelanggan/pemakai/pengguna biasanya memberikan masukan berdasarkan hasil yang didapat dari tahap engineering dan instalasi.

Kelebihan model ini adalah sangat mempertimbangkan resiko kemungkinan munculnya kesalahan sehingga sangat dapat diandalkan untuk pengembangan perangkat lunak skala besar. Pendekatan model ini dilakukan melalui tahapan-tahapan yang sangat baik dengan menggabungkan model waterfall ditambah dengan pengulangan-pengulangan sehingga lebih realistis untuk mencerminkan keadaan sebenarnya. Baik pengembang maupun pemakai dapat cepat mengetahui letak kekurangan dan kesalahan dari sistem karena proses-prosesnya dapat diamati dengan baik.

Kekurangan model ini adalah waktu yang dibutuhkan untuk mengembangkan perangkat lunak cukup panjang demikian juga biaya yang besar. Selain itu, sangat tergantung kepada tenaga ahli yang dapat memperkirakan resiko. Terdapat pula kesulitan untuk mengontrol proses. Sampai saat ini, karena masih relatif baru, belum ada bukti apakah metode ini cukup handal untuk diterapkan.

Model Boehm sangat cocok diterapkan untuk pengembangan sistem dan perangkat lunak skala besar di mana pengembang dan pemakai dapat lebih mudah memahami kondisi pada setiap tahapan dan bereaksi terhadap kemungkinan terjadinya kesalahan. Selain itu, diharapkan juga waktu dan dana yang tersedia cukup memadai.

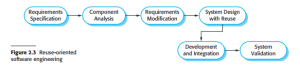
1. **Model Pengembangan Formal**

Pada model ini, digunakan notasi matematika yang terperinci dan penuh ketelitian dalam mengidentifikasi desain dan menguji sistem yang berbasis komputer. Metode ini sering dipakai untuk spesifikasi yang detail, rancangan dan verifikasi pada bagian-bagian sistem yang penting (bersifat kritikal) seperti pada sistem avionic dan aerospace, serta pada sistem keamanan yang kritikal pada monitor jantung, ATM (Anjungan Tunai Mandiri) dan pada perbankan.

Secara khusus, metode formal sangat cocok dijalankan pada sistem yang kompleks.

1. **Model Pengembangan Berorientasi Pemakaian Ulang (Reuse-oriented software engineering)**

Metode pengembangan yang berorientasi pemakaian ulang ini bergantung pada sejumlah besar komponen perangkat lunak yang dapat didaur ulang, yang bisa didapat, dan beberapa kerangka kerja integrasi untuk komponen-komponen ini.



Tahap-tahap pengembangan :

1. Analisis komponen. Jika diketahui spesifikasi persyaratan, komponen-komponen untuk implementasi spesifikasi tersebut akan dicari. Biasanya, tidak ada kesesuaian yang tepat dan komponen yang dapat dipakai hanya memberikan sebagian dari fungsional yang dibutuhkan.
2. Modifikasi kebutuhan. Pada tahap ini, kebutuhan dianalisis dengan menggunakan informasi mengenai komponen yang telah didapat. Kebutuhan kemudian dimodifikasi untuk merefleksikan komponen yang tersedia. Jika modifikasi tidak mungkin dilakukan, maka kegiatan analisis komponen bisa dulang untuk mencari solusi alternatif.
3. Perancangan sistem dengan pemakaian ulang. Pada tahap ini, kerangka kerja sistem dirancang, atau kerangka kerja yang telah ada dipakai ulang. Perancang memperhitungkan komponen yang dipakai ulang dan mengatur kerangka kerja untuk menyesuaikan. Beberapa perangkat lunak yang baru mungkin perlu dirancang jika komponen yang dapat dipakai ulang tidak tersedia.
4. Pengembangan dan integrasi. Perangkat lunak yang tidak dapat dibeli akan dikembangkan dan komponen dari sistem COTS (*Commercial Off-The-Shelf system*) diintegrasikan untuk membentuk sistem. Integrasi sistem pada model ini bisa merupakan kegiatan yang terpisah.

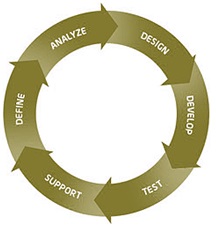
Keuntungan: mengurangi besarnya perangkat lunak yang akan dikembangkan, serta memperkecil biaya dan resiko.

Ada tiga jenis komponen perangkat lunak yang dapat digunakan dalam proses *reuse-oriented*:

1. Layanan web yang dikembangkan sesuai dengan standar layanan dan yang tersedia untuk permintaan jauh.
2. Koleksi objek yang dikembangkan sebagai paket yang diintegrasikan dengan kerangka komponen seperti NET atau J2EE.
3. Sistem perangkat lunak Stand-alone yang dikonfigurasi untuk digunakan dalam lingkungan tertentu.

**5. RAD Model**

RAD adalah proses pembangunan Perangkat Lunak yang menekankan pada siklus pengembangan yang pendek dan singkat. Model ini mengawinkan model waterfall dan model component based construction.



Gambar 3. Model RAD

*Secara ringkas, tahapan-tahapan RAD adalah sebagai berikut.*

*Tahap Pemodelan Bisnis:* dibuat agar dapat menjawab pertanyaan-pertanyaan berikut: informasi apa yang mengontrol proses bisnis? Informasi apa yang didapat? Siapa yang mendapatkannya? Untuk siapa informasi itu ditujukan? Siapa yang akan memprosesnya?

*Tahap Pemodelan Data:* informasi-informasi yang dipadu dari pemodelan bisnis dipilah-pilah ke menjadi sekumpulan objek data yang masing-masing objek diidentifikasikan dan ditentukan hubungan antara objek-objek tersebut.

*Tahap Pemodelan Proses:* aliran informasi yang didapat dalam proses pemodelan data diolah sedemikian untuk dapat menopang fungsi-fungsi bisnis. Prosesnya dikreasikan untuk menambah, memodifikasi, menghapus dan atau mendapatkan kembali sebuah objek data.

*Tahap Pembuatan Aplikasi:* RAD dapat saja memakai kembali komponen program yang sudah ada bila dimungkinkan, atau membuat komponen yang dapat digunakan lagi bila diperlukan di masa mendatang. RAD juga diasumsikan menggunakan teknik generasi keempat (4GT).

*Tahap Pengujian dan Pergantian:* Proses RAD menekankan pada pemakaian kembali yang memungkinkan berkurangnya keseluruhan waktu pengujian, namun komponen harus diuji dan harus dilatih secara penuh dan terintegrasi.

Kelebihan model RAD: tahap-tahap RAD membuatnya mampu untuk menggunakan kembali komponen yang ada (reusable object), karena setiap komponen software dikerjakan secara terpisah dengan tim-tim tersendiri sehingga dapat digunakan juga untuk aplikasi lain yang pada akhirnya akan menghemat waktu. Penggunaan tim yang terpisah untuk mengerjakan pekerjaan yang berbeda membuat pekerjaan lebih cepat dalam proses integrasi dan efisien terhadap waktu tanpa mengacaukan aplikasi.

Kelemahan model RAD: Tidak begitu cocok untuk proyek dengan skala besar karena dibutuhkan sumber daya manusia yang semakin banyak seiring dengan semakin banyaknya komponen yang dikerjakan, selain itu, semakin besar proyek, semakin kompleks pula koordinasi yang dibutuhkan. Dalam waktu yang singkat, rasanya sulit untuk pengembang dan pemakai berkomitmen untuk melaksanakan berbagai kegiatan untuk melengkapi sistem. Apalagi bila sistem ternyata tidak dapat dimodularisasi sementara sistem mempunyai resiko teknik yang tinggi.

Model RAD sangat tepat diterapkan untuk sistem yang telah jelas dan lengkap kebutuhannya, di mana terdapat komponen-komponen yang dapat dipakai kembali dalam proyek yang berskala kecil dengan waktu pengembangan perangkat lunak yang singkat.