



MODUL 2

CPL230 : Pengembangan Perangkat Lunak

Oleh :

5165 –Kundang K Juman
Prodi : Teknik Informatika

www.esaunggul.ac.id



Modul : 2 **METODOLOGI PENGEMBANGAN** **SISTEM INFORMASI**

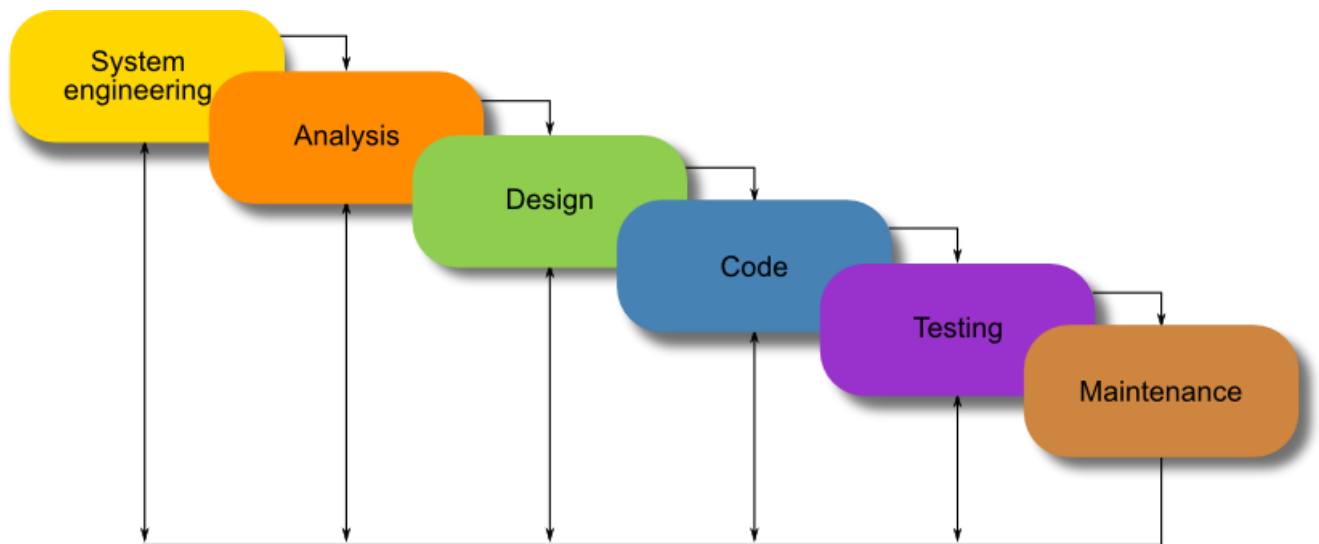
Oleh : Kundang K Juman

METODOLOGI PENGEMBANGAN SISTEM INFORMASI

Untuk pengembangan sistem penelitian ini menggunakan model SDLC (Software Development Life Cycle). System Development Life Cycle (SDLC) adalah proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan

sebuah sistem. SDLC juga merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak, yang terdiri dari tahap-tahap: rencana (planning), analisis (analysis), desain (design), implementasi (implementation), uji coba (testing) dan pengelolaan (maintenance). Definisi Pengembangan Sistem Pengembangan sistem dapat berarti menyusun suatu sistem yang baru untuk menggantikan sistem lama secara keseluruhan atau memperbaiki sistem yang telah ada. Sistem lama perlu diperbaiki atau diganti disebabkan karena beberapa hal yaitu sebagai berikut: 1. Adanya permasalahan-permasalahan yang timbul di sistem yang lama. Permasalahan tersebut dapat berupa : a. Ketidakberesan pada sistem yang lama tidak dapat berjalan atau berfungsi sebagaimana diharapkan b. Pertumbuhan organisasi yang menyebabkan harus disusunnya suatu sistem yang baru. Pertumbuhan organisasi diantaranya adalah kebutuhan informasi yang semakin luas dan volume pengolahan data yang semakin meningkat. 2. Untuk meraih kesempatan-kesempatan. Organisasi mulai merasakan bahwa teknologi informasi perlu digunakan untuk meningkatkan penyediaan informasi sehingga dapat mendukung dalam proses pengambilan keputusan yang akan dilakukan oleh manajemen. 3. Adanya instruksi / desakan dari organisasi. Penyusunan sistem yang baru dapat pula terjadi karena adanya instruksi-instruksi dari pimpinan atau dari luar organisasi karena adanya permasalahan, kesempatan atau instruksi. Sistem yang baru perlu dikembangkan untuk memecahkan permasalahan yang timbul, meraih kesempatan yang ada atau memenuhi instruksi yang diberikan dengan adanya sistem yang baru diharapkan terjadi peningkatan-peningkatan sebagai berikut :

a) Informasi Peningkatan terhadap kualitas informasi yang disajikan b) Kinerja Peningkatan terhadap kinerja sistem sehingga menjadi lebih efektif. c) Efisiensi Peningkatan terhadap efisiensi operasi. Efisiensi berbeda dengan ekonomis berhubungan dengan bagaimana sumber daya tersebut digunakan dengan pemborosan yang paling minimum a. Pengendalian Peningkatan terhadap pengendalian untuk mendeteksi dan memperbaiki kesalahan serta kecurangan-kecurangan yang akan terjadi. b. Pelayanan Peningkatan terhadap pelayanan yang diberikan oleh sistem.



Gambar Salah satu Methode Pengembangan Perangkat Lunak

Model – Model Pengembangan Perangkat Lunak Beserta Contoh Penerapannya :

Model Sekuensial Linier atau Waterfall Development Model

Model Pengembangan Air Terjun, merupakan paradigma model pengembangan perangkat lunak paling tua, dan paling banyak dipakai. Model ini mengusulkan sebuah pendekatan perkembangan perangkat lunak yang sistematis dan sekuensial yang dimulai pada tingkat dan kemajuan sistem pada seluruh tahapan analisis, desain, kode, pengujian, dan pemeliharaan.

Berikut Merupakan Tahapan – tahapan Pengembangan Model Sekuensial Linear / Waterfall Development Model :

Rekayasa dan pemodelan sistem/informasi

Langkah pertama dimulai dengan membangun keseluruhan elemen sistem dan memilah bagian-bagian mana yang akan dijadikan bahan pengembangan perangkat lunak, dengan memperhatikan hubungannya dengan Hardware, User, dan Database.

Analisis kebutuhan perangkat lunak

Pada proses ini, dilakukan penganalisaan dan pengumpulan kebutuhan sistem yang meliputi Domain informasi, fungsi yang dibutuhkan unjuk kerja/performansi dan antarmuka. Hasil penganalisaan dan pengumpulan tersebut didokumentasikan dan diperlihatkan kembali kepada pelanggan.

Desain

Pada proses Desain, dilakukan penerjemahan syarat kebutuhan sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuatnya proses pengkodean (coding).

Proses ini berfokus pada struktur data, arsitektur perangkat lunak, representasi interface, dan detail algoritma prosedural.

Pengkodean

Pengkodean merupakan proses menterjemahkan perancangan desain ke bentuk yang dapat dimengerti oleh mesin, dengan menggunakan bahasa pemrograman.

Pengujian

Setelah Proses Pengkodean selesai, dilanjutkan dengan proses pengujian pada program perangkat lunak, baik Pengujian logika internal, maupun Pengujian eksternal fungsional untuk memeriksa segala kemungkinan terjadinya kesalahan dan memeriksa apakah hasil dari pengembangan tersebut sesuai dengan hasil yang diinginkan.

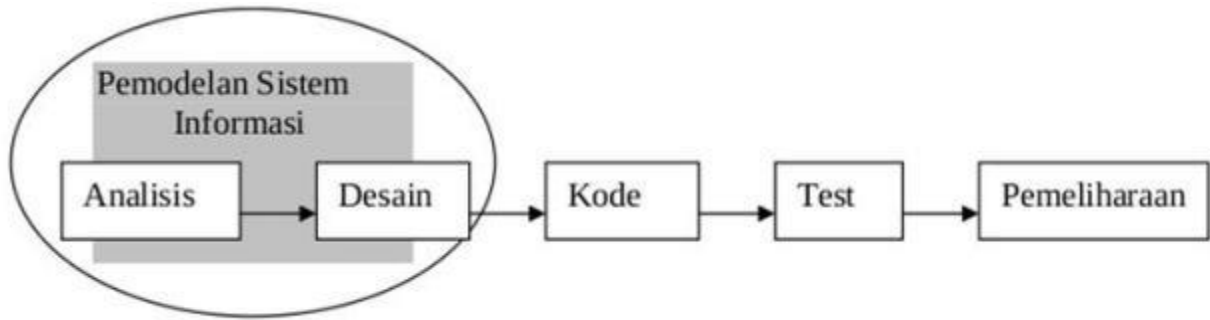
Pengujian

Setelah Proses Pengkodean selesai, dilanjutkan dengan proses pengujian pada program perangkat lunak, baik Pengujian logika internal, maupun Pengujian eksternal fungsional untuk memeriksa segala kemungkinan terjadinya kesalahan dan memeriksa apakah hasil dari pengembangan tersebut sesuai dengan hasil yang diinginkan.

Pemeliharaan

Proses Pemeliharaan merupakan bagian paling akhir dari siklus pengembangan dan dilakukan setelah perangkat lunak dipergunakan. Kegiatan yang dilakukan pada proses pemeliharaan antara lain :

- a. Corrective Maintenance : yaitu mengoreksi apabila terdapat kesalahan pada perangkat lunak, yang baru terdeteksi pada saat perangkat lunak dipergunakan.
- b. Adaptive Maintenance : yaitu dilakukannya penyesuaian/perubahan sesuai dengan lingkungan yang baru, misalnya hardware, periperal, sistem operasi baru, atau sebagai tuntutan atas perkembangan sistem komputer, misalnya penambahan driver, dll.
- c. Perfektive Maintenance : Bila perangkat lunak sukses dipergunakan oleh pemakai. Pemeliharaan ditujukan untuk menambah kemampuannya seperti memberikan fungsi-fungsi tambahan, peningkatan kinerja dan sebagainya.



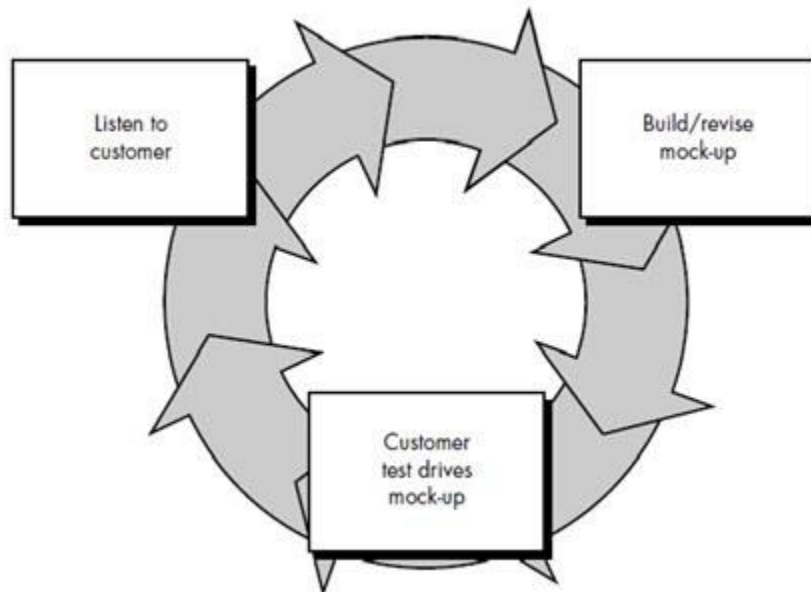
Contoh Penerapan dari Pengembangan Model Sekuensial Linear / Waterfall Development Model

1. **Kelebihan Model Sekuensial Linear / Waterfall Development Model :**
 2. Tahapan proses pengembangannya tetap (pasti), mudah diaplikasikan, dan prosesnya teratur.
 - a. Cocok digunakan untuk produk software/program yang sudah jelas kebutuhannya di awal, sehingga minim kesalahannya.
 - b. Software yang dikembangkan dengan metode ini biasanya menghasilkan kualitas yang baik.
 - c. Dokumen pengembangan sistem sangat terorganisir, karena setiap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya.
2. **Kekurangan Model Sekuensial Linear / Waterfall Development Model :**
 - a. Proyek yang sebenarnya jarang mengikuti alur sekuensial seperti diusulkan, sehingga perubahan yang terjadi dapat menyebabkan hasil yang sudah didapatkan tim pengembang harus diubah kembali/iterasi sering menyebabkan masalah baru.
 - b. Terjadinya pembagian proyek menjadi tahap-tahap yang tidak fleksibel, karena komitmen harus dilakukan pada tahap awal proses.
 - a. Sulit untuk mengalami perubahan kebutuhan yang diinginkan oleh customer/pelanggan.
 - b. Pelanggan harus sabar untuk menanti produk selesai, karena dikerjakan tahap per tahap, dan proses pengerjaannya akan berlanjut ke setiap tahapan bila tahap sebelumnya sudah benar-benar selesai.
 - c. Perubahan ditengah-tengah pengerjaan produk akan membuat bingung tim pengembang yang sedang membuat produk.

- d. Adanya waktu kosong (menganggur) bagi pengembang, karena harus menunggu anggota tim proyek lainnya menuntaskan pekerjaannya.

Model Prototype :

Metode Prototype merupakan suatu paradigma baru dalam metode pengembangan perangkat lunak dimana metode ini tidak hanya sekedar evolusi dalam dunia pengembangan perangkat lunak, tetapi juga merevolusi metode pengembangan perangkat lunak yang lama yaitu sistem sekuensial yang biasa dikenal dengan nama SDLC atau waterfall development model.



Dalam Model Prototype, prototype dari perangkat lunak yang dihasilkan kemudian dipresentasikan kepada pelanggan, dan pelanggan tersebut diberikan kesempatan untuk memberikan masukan sehingga perangkat lunak yang dihasilkan nantinya betul-betul sesuai dengan keinginan dan kebutuhan pelanggan.

Perubahan dan presentasi prototype dapat dilakukan berkali-kali sampai dicapai kesepakatan bentuk dari perangkat lunak yang akan dikembangkan.

Teknik – teknik Prototyping Meliputi :

- Perancangan Model
- Perancangan Dialog

- Simulasi

Berikut adalah 4 langkah yang menjadi karakteristik dalam proses pengembangan pada metode prototype, yaitu :

- Pemilihan fungsi
- Penyusunan Sistem Informasi
- Evaluasi
- Penggunaan Selanjutnya

Metode ini menyajikan gambaran yang lengkap dari suatu sistem perangkat lunak, terdiri atas model kertas, model kerja dan program. Pihak pengembang akan melakukan identifikasi kebutuhan pemakai, menganalisa sistem dan melakukan studi kelayakan serta studi terhadap kebutuhan pemakai, meliputi model interface, teknik prosedural dan teknologi yang akan dimanfaatkan.

Berikut adalah Tahapan – tahapan Proses Pengembangan dalam Model Prototype, yaitu :

Pengumpulan kebutuhan

Pelanggan dan pengembang bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat.

Membangun prototyping

Membangun prototyping dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan (misalnya dengan membuat input dan format output).

Evaluasi protootyping

Evaluasi ini dilakukan oleh pelanggan, apakah prototyping yang sudah dibangun sudah sesuai dengan keinginan pelanggan atau belum. Jika sudah sesuai, maka langkah selanjutnya akan diambil. Namun jika tidak, prototyping direvisi dengan mengulang langkah-langkah sebelumnya.

Mengkodekan sistem

Dalam tahap ini prototyping yang sudah di sepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.

Menguji sistem

Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, kemudian dilakukan proses Pengujian. Pengujian ini dilakukan dengan White Box, Black Box, Basis Path, pengujian arsitektur, dll.

Evaluasi Sistem

Pelanggan mengevaluasi apakah perangkat lunak yang sudah jadi sudah sesuai dengan yang diharapkan . Jika ya, maka proses akan dilanjutkan ke tahap selanjutnya, namun jika perangkat lunak yang sudah jadi tidak/belum sesuai dengan apa yang diharapkan, maka tahapan sebelumnya akan diulang.

Menggunakan sistem

Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan.

Model Prototyping ini sangat sesuai diterapkan untuk kondisi yang beresiko tinggi di mana masalah-masalah tidak terstruktur dengan baik, terdapat fluktuasi kebutuhan pemakai yang berubah dari waktu ke waktu atau yang tidak terduga, bila interaksi dengan pemakai menjadi syarat mutlak dan waktu yang tersedia sangat terbatas sehingga butuh penyelesaian yang segera. Model ini juga dapat berjalan dengan maksimal pada situasi di mana sistem yang diharapkan adalah yang inovatif dan mutakhir sementara tahap penggunaan sistemnya relatif singkat.

Berikut merupakan Jenis – jenis dari Prototyping :

Feasibility prototyping

digunakan untuk menguji kelayakan dari teknologi yang akan digunakan untuk system informasi yang akan disusun.

Requirement prototyping

digunakan untuk mengetahui kebutuhan aktivitas bisnis user.

Desain Prototyping

digunakan untuk mendorong perancangan sistem informasi yang akan digunakan.

Implementation prototyping

merupakan lanjutan dari rancangan prototype, prototype ini langsung disusun sebagai suatu sistem informasi yang akan digunakan.

Contoh Penerapan Metode Prototype.

Sebuah rumah sakit ingin membuat aplikasi sistem database untuk

Contoh Penerapan Metode Prototype.

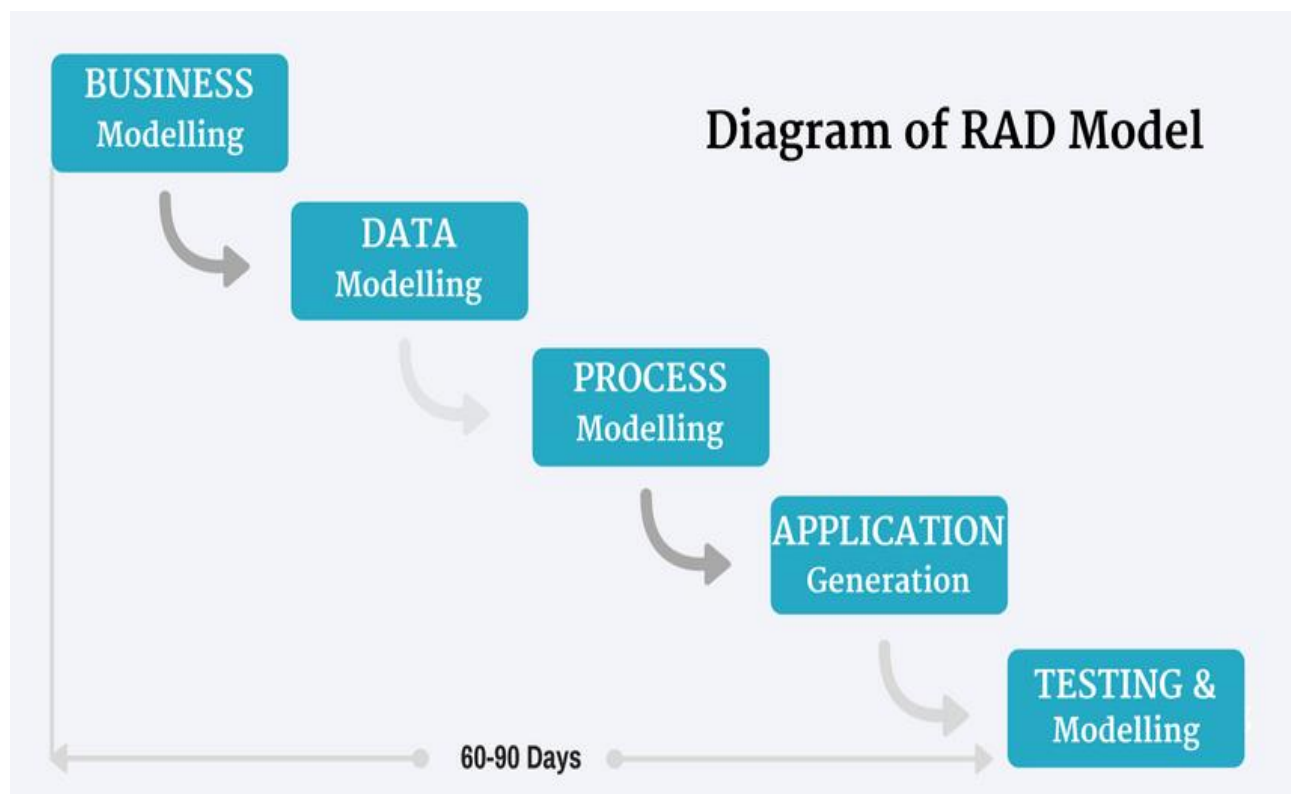
Sebuah rumah sakit ingin membuat aplikasi sistem database untuk pendataan pasiennya. Seorang atau sekelompok programmer akan melakukan identifikasi mengenai apa saja yang dibutuhkan oleh pelanggan, dan bagaimana model kerja program tersebut. Kemudian dilakukan rancangan program yang diujikan kepada pelanggan. Hasil/penilaian dari pelanggan dievaluasi, dan analisis kebutuhan pemakai kembali dilakukan.

- **Kelebihan Model Prototype :**
- Pelanggan berpartisipasi aktif dalam pengembangan sistem, sehingga hasil produk pengembangan akan semakin mudah disesuaikan dengan keinginan dan kebutuhan pelanggan.
- Penentuan kebutuhan lebih mudah diwujudkan.
- Mempersingkat waktu pengembangan produk perangkat lunak.
- Adanya komunikasi yang baik antara pengembang dan pelanggan.
- Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan.
- Lebih menghemat waktu dalam pengembangan sistem.
- Penerapan menjadi lebih mudah karena pelanggan mengetahui apa yang diharapkannya.
- **Kekurangan Model Prototype :**
- Proses analisis dan perancangan terlalu singkat.
- Biasanya kurang fleksibel dalam menghadapi perubahan.
- Walaupun pemakai melihat berbagai perbaikan dari setiap versi prototype, tetapi pemakai mungkin tidak menyadari bahwa versi tersebut dibuat tanpa memperhatikan kualitas dan pemeliharaan jangka panjang.

- Pengembang kadang-kadang membuat kompromi implementasi dengan menggunakan sistem operasi yang tidak relevan dan algoritma yang tidak efisien.

Model Rapid Application Development (RAD)

Rapid Application Development (RAD) adalah sebuah model proses pengembangan perangkat lunak sekuensial linier yang menekankan siklus pengembangan yang sangat pendek (kira-kira 60 sampai 90 hari). Model RAD ini merupakan sebuah adaptasi “kecepatan tinggi” dari model sekuensial linier dimana pengembangan cepat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen.



Berikut adalah Tahapan – tahapan Proses Pengembangan dalam Model Rapid Application Development (RAD), yaitu :

Bussiness Modeling

Fase ini untuk mencari aliran informasi yang dapat menjawab pertanyaan berikut:

- a. Informasi apa yang menegndalikan proses bisnis?
- b. Informasi apa yang dimunculkan?
- c. Di mana informasi digunakan ?
- d. Siapa yang memprosesnya ?

Data Modeling

Aliran informasi yang didefinisikan sebagai bagian dari fase bussiness modeling disaring ke dalam serangkaian objek data yang dibutuhkan untuk menopang bisnis tersebut. Karakteristik (atribut) masing-masing objek diidentifikasi dan hubungan antar objek-objek tersebut didefinisikan.

Proses Modeling

Aliran informasi yang didefinisikan di dalam fase data modeling ditransformasikan untuk mencapai aliran informasi yang perlu bagi implementasi sebuah fungsi bisnis. Gambaran pemrosesan diciptakan untuk menambah, memodifikasi, menghapus, atau mendapatkan kembali sebuah objek data.

Aplication Generation

Selain menggunakan bahasa pemrograman generasi ketiga, RAD juga memakai komponen program yang telah ada atau menciptakan komponen yang bisa dipakai lagi. Ala-alat bantu bisa dipakai untuk memfasilitasi konstruksi perangkat lunak.

Testing dan Turnover

Karena proses RAD menekankan pada pemakaian kembali, banyak komponen program telah diuji. Hal ini mengurangi keseluruhan waktu pengujian. Tetapi komponen baru harus diuji dan semua interface harus dilatih secara penuh.

Kelebihan Model RAD :

Lebih efektif dari Pengembangan Model waterfall/sequential linear dalam menghasilkan sistem yang memenuhi kebutuhan langsung dari pelanggan.

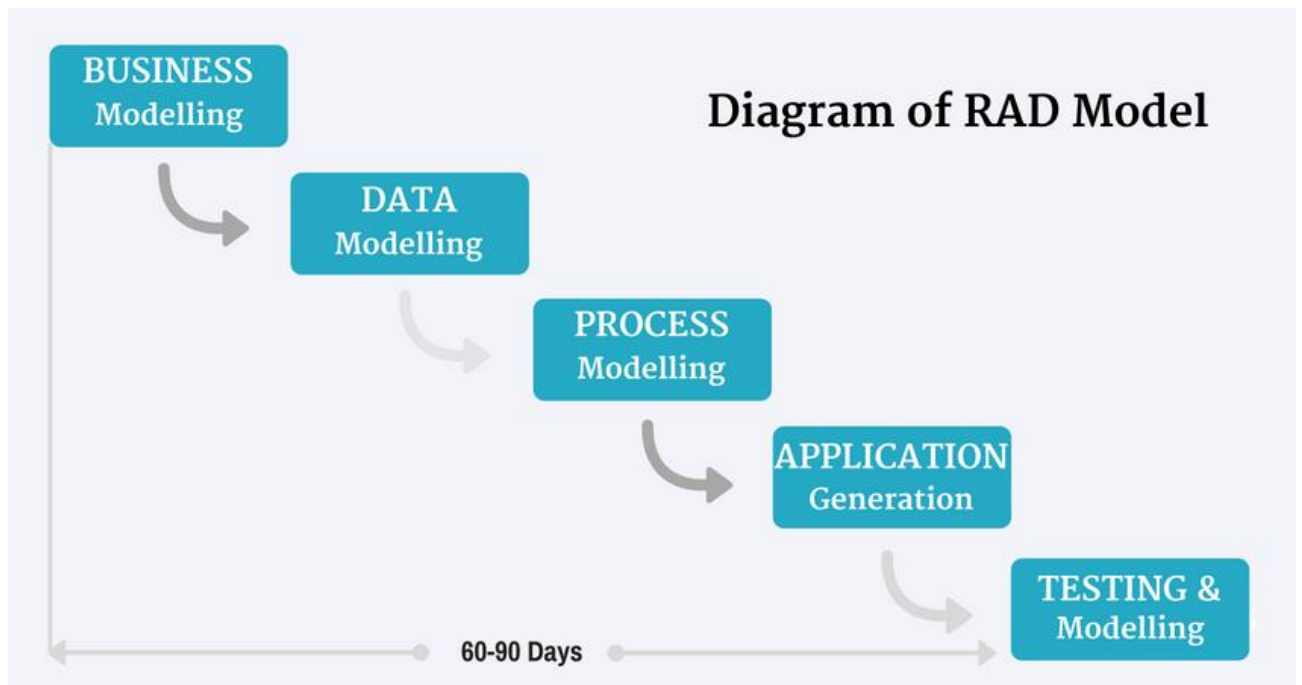
- Cocok untuk proyek yang memerlukan waktu yang singkat.
- Model RAD mengikuti tahap pengembangan sistem seperti pada umumnya, tetapi mempunyai kemampuan untuk menggunakan kembali komponen yang ada sehingga pengembang tidak perlu membuatnya dari awal lagi sehingga waktu pengembangan menjadi lebih singkat dan efisien.
- **Kekurangan Model RAD :**
- Model RAD menuntut pengembangan dan pelanggan memiliki komitmen di dalam aktivitas rapid-fire yang diperlukan untuk melengkapi sebuah sistem, di dalam kerangka waktu yang sangat diperpendek. Jika komitmen tersebut tidak ada, proyek RAD akan gagal.
- Tidak semua aplikasi sesuai untuk RAD, bila system tidak dapat dimodulkan dengan teratur, pembangunan komponen penting pada RAD akan menjadi sangat bermasalah.
- RAD tidak cocok digunakan untuk sistem yang mempunyai resiko teknik yang tinggi.
- Membutuhkan Tenaga kerja yang banyak untuk menyelesaikan sebuah proyek dalam skala besar.
- Jika ada perubahan di tengah-tengah pengerjaan maka harus membuat kontrak baru antara pengembang dan pelanggan.

Model Evolutionary Development / Evolutionary Software Process Models

Model Evolutionary Development bersifat iteratif (mengandung perulangan). Hasil prosesnya berupa produk yang makin lama makin lengkap sampai versi terlengkap dihasilkan sebagai produk akhir dari proses. Model Evolutionary Development / Evolutionary Software Process terbagi menjadi 2, yaitu :

Model Incremental

Model Incremental merupakan hasil kombinasi elemen-elemen dari model waterfall yang diaplikasikan secara berulang, atau bisa disebut gabungan dari Model linear sekuensial (waterfall) dengan Model Prototype. 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.



Model ini berfokus pada penyampaian produk operasional dalam Setiap pertambahanya. Pertambahan awal ada di versi stripped down dari produk akhir, tetapi memberikan kemampuan untuk melayani pemakai dan juga menyediakan platform untuk evaluasi oleh pemakai. Model ini cocok dipakai untuk proyek kecil dengan anggota tim yang sedikit dan ketersediaan waktu yang terbatas.

Pada proses Pengembangan dengan Model Incremental, perangkat lunak dibagi menjadi serangkaian increment yang dikembangkan secara bergantian.

Contoh Penerapan Model Incremental

Perangkat lunak pengolah kata yang dikembangkan dengan menggunakan paradigma pertambahan akan menyampaikan manajemen file, editing, serta fungsi penghasilan dokumen

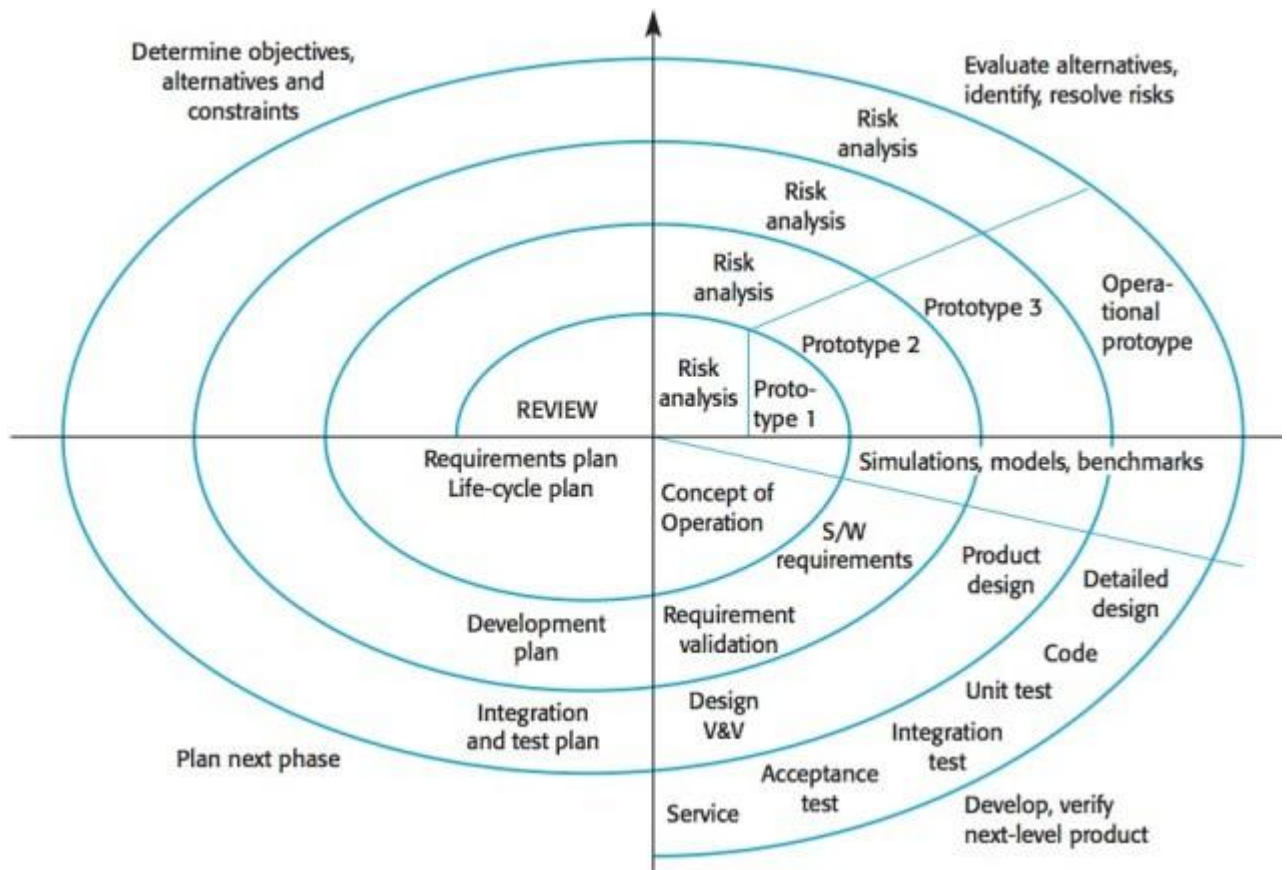
pada penambahan pertama, dan selanjutnya. Penambahan pertama dapat disebut sebagai produk inti (core product). Dan pada penambahan selanjutnya, produk inti akan dikembangkan terus hingga menghasilkan produk jadi yang siap untuk digunakan/dipasarkan.

Kelebihan Model Incremental :

- Personil bekerja optimal.
- mampu mengakomodasi perubahan secara fleksibel, dengan waktu yang relatif singkat dan tidak dibutuhkan anggota/tim kerja yang banyak untuk menjalankannya.
- Pihak konsumen dapat langsung menggunakan dahulu bagian-bagian yang telah selesai dibangun. Contohnya pemasangan data karyawan.
- Mengurangi trauma karena perubahan sistem. Klien dibiasakan perlahan-lahan menggunakan produknya setiap bagian demi bagian.
- Memaksimalkan pengembalian modal investasi konsumen.
- **Kekurangan Model Incremental :**
- Tidak cocok untuk proyek berukuran besar (lebih dari 200.000 baris coding).
- Sulit untuk memetakan kebutuhan pemakai ke dalam rencana spesifikasi tiap-tiap hasil dari increment.

Model Spiral / Model 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.



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 Spiral/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.

Referensi :

1. <http://www.markijar.com/2015/04/kelebihan-dan-kekurangan-model-proses>.
2. [htmlhttp://id.wikipedia.org/wiki/Proses_pengemb](http://id.wikipedia.org/wiki/Proses_pengemb)

•

