## **Metode Konvensional**

1. **Rapid application development (RAD)**

a. Pengertian : adalah model proses pembangunan perangkat lunak yang tergolong dalam teknik incremental (bertingkat).

b. Tahapan RAD :

* Rencana kebutuhan (requirement planing), pada tahap ini user dan analis melakukan semacam pertemuan untuk melakukan identifikasi tujuan dari aplikasi atau sistem dan melakukan identifikasi kebutuhan informasi untuk mencapai tujuan.
* Proses design(design workshop), pada tahap ini adalah melakukan proses design dan melakukan perbaikakn-erbaikan apabila masih terdapat tidak sesuain design antara user dan analis.
* Implementasi, setelah design dari sistem yang akan dibuat sudah disetujui baik oleh user dan analis, maka pada tahap ini programmer mengembangkan design menjadi suatu program. Setelah program selesai baik sebagian maupun keseluruhan, maka dilakukan proses pengujian terhadap program tersebut apakah terdapat kesalahan atau tidak.

c. Kondisi Penunjang :

* sistem berjalan sendiri
* kinerja dari sistem bukan faktor terpenting
* distribusi produk yang bersifat sempit
* ruang lingkup yang terbatas
* kehandalan dari sistem bukan faktor terpenting
* membutuhkan teknologi yang tidak terlalu baru
* sistem dapat dipecah-pecah menjadi bagian yang lebih kecil

d. Kondisi Penghambat :

* sistem harus dapat berjalan secara bersamaan dengan sistem yang lama
* komponen-komponen penunjang sangat langka untuk didapatkan
* kinerja yang optimal merupakan faktor terpenting
* distribusi produk yang bersifat luas
* ruang lingkup yang luas
* apabila digunakan untuk membuat sistem operasi, membutuhkan sistem yang handal
* sistem tidak dapat dipecah menjadi bagian yang lebih kecil

f. Keuntungan :

* Membeli sistem yang baru memungkinkan untuk lebih menghemat biaya ketimbang mengembangkan sendiri
* Proses pengiriman menjadi lebih mudah
* Mudah untuk diamati karena menggunnakan model prototype
* Lebih fleksibel karena pengembang dapat melakukan proses design ulang pada saat yang bersamaan
* Bisa mengurangi penulisan kode yang kompleks karena menggunakan wizard
* Keterlibatan user semakin meningkat karena merupakan bagian dari tim dari secara keseluruhan
* Mampu meminimalkan kesalahan-kesalahan dengan menggunkana alat-alat bantuan
* Mempercepat waktu pengembangan sistem secara keseluruhan karena cenderung mengabaikan kualitas
* Tampilan yang lebih standart dann nyaman dengan bantuan software pendukung

f. Kerugian :

* Dengan melakukan pembelian belum tentu bisa menghemat biaya dibandingkan dengan mengembangkan sendiri.
* Membutuhkan biaya tersendiri untuk membeli peralatan-peralatan penunjang seperti misalnya *software*dan *hardware*.
* Kesulitan melakukan pengukuran mengenai kemajuan proses.
* Kurang efisien karena apabila melakukan pengkodean dengan menggunakan tangan bisa lebih efisien.
* Ketelitian menjadi berkurang karena tidak menggunakan metode yang formal dalam melakukan pengkodean.
* Lebih banyak terjadi kesalahan apabila hanya mengutamakan kecepatan dibandingkan dengan biaya dan kualitas.
* Fasilitas-fasilitas banyak yang dikurangi karena terbatasnya waktu yang tersedia.
* Sistem sulit diaplikasikan di tempat yang lain.

1. **Prototype**

a. Pengertian :

adalah salah satu pendekatan dalam rekayasa perangkat lunak yang secara

langsung mendemonstrasikan bagaimana sebuah perangkat lunak atau komponen- komponen perangkat lunak akan bekerja dalam lingkungannya sebelum tahapan konstruksi aktual dilakukan (Howard, 1997).

b. Beberapa model prototype :

* Reusable prototype : Prototype yang akan ditransformasikan menjadi produk final.
* Throwaway prototype : Prototype yang akan dibuang begitu selesai menjalankan maksudnya.
* Input/output prototype : Prototype yang terbatas pada antar muka pengguna (user interface).
* Processing prototype : Prototype yang meliputi perawatan file dasar dan proses-proses transaksi
* System prototype : Prototype yang berupa model lengkap dari perangkat lunak.

c. Tahapan:

***1. pengumpulan kebutuhan***  
developer dan klien bertemu dan menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya. Detil kebutuhan mungkin tidak dibicarakan disini, pada awal pengumpulan kebutuhan

***2. perancangan***  
perancangan dilakukan cepat dan rancangan mewakili semua aspek software yang diketahui, dan rancangan ini menjadi dasar pembuatan prototype.

***3. Evaluasi prototype***  
klien mengevaluasi prototype yang dibuat dan digunakan untuk memperjelas kebutuhan software.

Perulangan ketiga proses ini terus berlangsung hingga semua kebutuhan terpenuhi. Prototype-prototype dibuat untuk memuaskan kebutuhan klien dan untuk memahami kebutuhan klien lebih baik. Prototype yang dibuat dapat dimanfaatkan kembali untuk membangun software lebih cepat, namun tidak semua prototype bisa dimanfaatkan.

d. Keuntungan:

* Pemodelan membutuhkan partisipasi aktif dari end-user. Hal ini akan meningkatkan sikap dan dukungan pengguna untuk pengerjaan proyek. Sikap moral pengguna akan meningkat karena system berhubungan nyata dengan mereka.
* Perubahan dan iterasi merupakan konsekuensi alami dari pengembangan system-sehingga end user memiliki keinginan untuk merubah pola pikirnya. Prototyping lebih baik menempatkan situasi alamiah ini karena mengasumsikan perubahan model melalui iterasi kedalam system yang dibutuhkan.
* Prototyping adalah model aktif, tidak pasif, sehingga end user dapat melihat, merasakan, dan mengalaminya.
* Kesalahan yang terjadi dalam prototyping dapat dideteksi lebih dini
* Prototyping dapat meningkatkan kreatifitas karena membolehkan adanya feedback dari end user. Hal ini akan memberikan solusi yang lebih baik.
* Prototyping mempercepat beberapa fase hidup dari programmer.

e. Kekurangan :

* Prototyping memungkinkan terjadinya pengembalian terhadap kode, implementasi, dan perbaikan siklus hidup yang dugunakan untuk mendominasi sistem informasi.
* Prototyping tidak menolak kebutuhan dari fase analisis sistem. Prototype hanya dapat memecahkan masalah yang salah dan memberi kesempatan sebagai sistem pengembangan konvensional.
* Prototyping dapat mengurangi kreatifitas perancangan.

**3. Waterfall**

a. Pengertian :

Metode air terjun atau yang sering disebut metode waterfall sering dinamakan siklus hidup klasik (classic life cycle), dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (planning), permodelan (modeling), konstruksi (construction), serta penyerahan sistem ke para pelanggan/pengguna (deployment), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan (Pressman, 2012).

b. Tahapan :

1.     *Requirement Analisis*

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi ini biasanya dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

2.      *System Design*

Spesifikasi kebutuhan dari tahap sebelumnya akan dipelajari dalam fase ini dan desain sistem disiapkan. Desain Sistem membantu dalam menentukan perangkat keras*(hardware)* dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

3.      *Implementation*

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut *unit,* yang terintegrasi dalam tahap selanjutnya. Setiap *unit* dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai *unit testing*.

4.        *Integration & Testing*

Seluruh *unit* yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing *unit*. Setelah integrasi seluruh sistem diuji untuk mengecek  setiap kegagalan maupun kesalahan.

5.        *Operation & Maintenance*

Tahap akhir dalam model *waterfall*. Perangkat lunak yang  sudah  jadi,  dijalankan  serta dilakukan  pemeliharaan. Pemeliharaan termasuk dalam memperbaiki  kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi *unit* sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

c. Kelebihan & Kekurangan

**Kelebihan model *Linear Sequential/Waterfall* :**

* Mudah diaplikasikan
* Memberikan template tentang metode analisis, desain, pengkodean, pengujian, dan pemeliharaan
* Cocok digunakan untuk produk software yang sudah jelas kebutuhannya di awal, sehingga minim kesalahannya

**Kekurangan model *Linear Sequential/Waterfall* :**

* Terjadinya pembagian proyek menjadi tahap-tahap yang tidak fleksibel, karena komitmen harus dilakukan pada tahap awal proses
* Sulit untuk mengalami perubahan kebutuhan yang diinginkan customer
* Customer harus sabar untuk menanti produk selesai, karena dikerjakan tahap per tahap,menyelesaikan tahap awal baru bisa ke tahap selanjutnya
* Perubahan ditengah-tengah pengerjaan produk akan membuat bingung team work yang sedang membuat produk
* Adanya waktu menganggur bagi pengembang, karena harus menunggu anggota tim proyek lainnya menuntaskan pekerjaannya.

**4. Spiral**

a. Pengertian :

adalah model proses perangkat lunak yang evolusioner, merangkai sifat iterative dari prototype dengan cara control dan aspek sistematis dari model sekuensial linier. Model yang berpotensi untuk pengembangan versi pertambahan perangkat lunak secara cepat.

b. Kerangka Kerja

Model spiral dibagi menjadi sejumlah aktifitas kerangka kerja, disebut juga wilayah tugas, di antara tiga sampai enam wilayah tugas. Tahap-tahap model tersebut dapat dijelaskan secara ringkas sebagai berikut.

1. Tahap Liason: pada tahap ini membangun komunikasi yang efektif di antara pengembangan dan pelanggan.

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

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

4. Tahap Rekayasa (engineering): pembuatan prototipe atau pembangunan satu atau lebih representasi dari aplikasi tersebut

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

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

c. Kelebihan & Kekerungan

**Kelebihan**  
1. Sangat mempertimbangkan resiko kemungkinan munculnya kesalahan sehingga sangat dapat diandalkan untuk pengembangan perangkat lunak skala besar.  
2. 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.  
3. Baik pengembang maupun pemakai dapat cepat mengetahui letak kekurangan dan kesalahan dari sistem karena proses-prosesnya dapat diamati dengan baik.

**Kekurangan**

1.Waktu yang dibutuhkan untuk mengembangkan perangkat lunak cukup panjang demikian juga biaya yang besar.

2. Sangat tergantung kepada tenaga ahli yang dapat memperkirakan resiko.

3. Terdapat pula kesulitan untuk mengontrol proses. Sampai saat ini, karena masih relatif baru, belum ada bukti apakah metode ini cukup handal untuk diterapkan.

4. Meyakinkan konsumen (khusunya dalam situasi kontrak) bahwa pendekatan evolusioner bisa dikontrol