Nama : Diky Arif Rahman

NIM : A11.2019.11973

Kelas : A11.4605

**1.Metode Waterfall**

**Pengertian Metode Waterfall**

Metode waterfall adalah salah satu jenis model pengembangan aplikasi dan termasuk ke dalam *classic life cycle* (siklus hidup klasik), yang mana menekankan pada fase yang berurutan dan sistematis. Untuk model pengembangannya, dapat dianalogikan seperti air terjun, dimana setiap tahap dikerjakan secara berurutan mulai dari atas hingga ke bawah.

Jadi, untuk setiap tahapan tidak boleh dikerjakan secara bersamaan. Sehingga, perbedaan dari metode waterfall dengan metode agile terletak pada tahapan SDLC -nya. Model ini juga termasuk ke dalam pengembangan perangkat lunak yang terbilang kurang iteratif dan fleksibel. Karena, proses yang mengarah pada satu arah saja seperti air terjun.

**Tahapan Model Waterfall**

Setelah mengetahui apa itu metode waterfall, selanjutnya masuk pada pembahasan mengenai tahapan metode waterfall.

**1. Requirement**

Tahapan metode waterfall yang pertama adalah mempersiapkan dan menganalisa kebutuhan dari software yang akan dikerjakan. Informasi dan *insight* yang diperoleh dapat berupa dari hasil wawancara, survei, studi literatur, observasi, hingga diskusi.

Biasanya di dalam sebuah perusahaan, tim analis akan menggali informasi sebanyak – banyaknya dari klien atau user yang menginginkan produk beserta dengan kebutuhan sistemnya. Selain itu, juga dapat mengetahui setiap batasan dari perangkat lunak yang akan dibuat.

**2. Design**

Tahap yang selanjutnya adalah pembuatan desain aplikasi sebelum masuk pada proses *coding.* Tujuan dari tahap ini, supaya mempunyai gambaran jelas mengenai tampilan dan antarmuka software yang kemudian akan dieksekusi oleh tim programmer.

Untuk proses ini, akan berfokus pada pembangunan struktur data, arsitektur software, perancangan interface, hingga perancangan fungsi internal dan eksternal dari setiap algoritma prosedural. Tim yang mengerjakan tahap ini, biasanya lebih banyak menggunakan UI/UX Designer, atau orang yang memiliki kemampuan dalam bidang desain grafis atau Web Designer.

**3. Implementation**

Tahapan metode waterfall yang berikutnya adalah implementasi kode program dengan menggunakan berbagai tools dan bahasa pemrograman sesuai dengan kebutuhan tim dan perusahaan. Jadi, pada tahap implementasi ini lebih berfokus pada hal teknis, dimana hasil dari desain perangkat lunak akan diterjemahkan ke dalam bahasa pemrograman melalui tim programmer atau developer.

Di dalam tahap pengembangan, biasanya dibagi lagi menjadi 3 tim yang memiliki tugas yang berbeda. Pertama ada *front end* (untuk *client side*), *backend* (untuk *server side*), dan *full stack* (gabungan antara *front end* dan *backend*). Selain itu, pada tahap ini juga dilakukan pemeriksaan lebih dalam terkait dengan modul yang sudah dibuat, apakah berjalan dengan semestinya atau tidak.

**4. Integration & Testing**

Tahap yang keempat, masuk dalam proses integrasi dan pengujian sistem. Pada tahap ini, akan dilakukan penggabungan modul yang sudah dibuat pada tahap sebelumnya. Setelah proses integrasi sistem telah selesai, berikutnya masuk pada pengujian modul.

Yang bertujuan untuk mengetahui apakah perangkat lunak sudah sesuai dengan desain, dan fungsionalitas dari aplikasi apakah berjalan dengan baik atau tidak. Jadi, dengan adanya tahap pengujian, maka dapat mencegah terjadinya kesalahan, bug, atau error pada program sebelum masuk pada tahap produksi. Orang yang bertanggung jawab untuk melakukan *testing* adalah QA (*Quality Assurance*) dan QC (*Quality Control*).

**5. Operation & Maintenance**

Tahapan metode waterfall yang terakhir adalah pengoperasian dan perbaikan dari aplikasi. Setelah dilakukan pengujian sistem, maka akan masuk pada tahap produk dan pemakaian perangkat lunak oleh pengguna (*user*). Untuk proses pemeliharaan, memungkinkan pengembang untuk melakukan perbaikan terhadap kesalahan yang ditemukan pada aplikasi setelah digunakan oleh *user*.

Jadi, pada intinya model waterfall ini dalam proses pemakaiannya mengikuti prinsip dari air terjun. Dimana setiap pekerjaan akan dilakukan secara berurutan mulai dari atas hingga ke bawah. Hal tersebut yang merupakan karakteristik dari SDLC ini.

Kami juga akan memberikan sedikit contoh fase SDLC model waterfall, menurut para ahli:

**A. Metode Waterfall Menurut Pressman 2015**

* Communication
* Planning
* Modelling
* Construction
* Deployment

**B. Metode Waterfall Menurut Sommerville 2011**

* Requirement Definition
* System and Software Design
* Implementation and Unit Testing
* Integration and System Testing
* Operation and Maintenance

**Kelebihan Metode Waterfall**

Berikut ini merupakan beberapa kelebihan yang dimiliki oleh metode waterfall, antara lain:

**1. *Workflow* yang jelas**

Dengan menggunakan model SDLC jenis ini, mempunyai rangkaian alur kerja sistem yang jelas dan terukur. Masing – masing tim, memiliki tugas dan tanggung jawab sesuai dengan bidang keahliannya. Serta dapat menyelesaikan pekerjaan sesuai dengan alokasi waktu yang telah ditentukan sebelumnya.

**2. Hasil dokumentasi yang baik**

Waterfall merupakan pendekatan yang sangat metodis, dimana setiap informasi akan tercatat dengan baik dan terdistribusi kepada setiap anggota tim secara cepat dan akurat.

Dengan adanya dokumen, maka pekerjaan dari setiap tim akan menjadi lebih mudah, serta mengikuti setiap arahan dari dokumen tersebut.

**3. Dapat menghemat biaya**

Kelebihan yang selanjutnya tentu saja dari segi *resource* dan biaya yang dikeluarkan oleh suatu perusahaan dengan menggunakan model ini. Jadi, dalam hal ini klien tidak dapat mencampuri urusan dari tim pengembang aplikasi. Sehingga pengeluaran biaya menjadi lebih sedikit.

Berbeda dengan metode Agile, yang mana klien dapat memberikan masukan dan *feedback* kepada tim developer terkait dengan perubahan atau penambahan beberapa fitur. Sehingga perusahaan akan mengeluarkan biaya yang lebih besar daripada Waterfall.

**4. Digunakan untuk pengembangan software berskala besar**

Metode ini dinilai sangat cocok untuk menjalankan pembuatan aplikasi berskala besar yang melibatkan banyak sumber daya manusia dan prosedur kerja yang kompleks. Akan tetapi, Model ini juga dapat digunakan untuk proyek berskala kecil dan menengah. Tentu saja disesuaikan dengan kondisi dan kebutuhan proyek yang diambil.

**Baca juga**: [Penjelasan Lengkap Metode Agile dalam Pengembangan Perangkat Lunak](https://www.sekawanmedia.co.id/metode-agile-development/)

**Kelemahan dari Metode Waterfall**

Berikut ini terdapat beberapa kelemahan dari metode waterfall, diantaranya adalah sebagai berikut:

**1. Membutuhkan tim yang solid**

Untuk menggunakan model SDLC ini, tentu saja membutuhkan dukungan dari setiap *stakeholders* yang ada. Setiap tim harus mempunyai kerja sama dan koordinasi yang baik. Dikarenakan, apabila salah satu tim tidak dapat menjalankan tugas dengan semestinya, maka akan sangat berpengaruh terhadap alur kerja tim yang lain.

**2. Masih kurangnya fleksibilitas**

Semua tim dituntut untuk bekerja sesuai dengan arahan dan petunjuk yang telah ditetapkan di awal. Sehingga, klien tidak dapat mengeluarkan pendapat dan *feedback* kepada tim pengembang. Klien hanya dapat memberikan masukan pada tahap awal perancangan sistem perangkat lunak saja.

**3. Tidak dapat melihat gambaran sistem dengan jelas**

Dengan model waterfall, customer tidak dapat melihat gambaran sistem secara jelas. Berbeda dengan model agile yang dapat terlihat dengan baik meskipun masih dalam proses pengembangan.

**4. Membutuhkan waktu yang lebih lama**

Proses pengerjaan dengan menggunakan waterfall terbilang cukup lama jika dibandingkan dengan model SDLC yang lain. Karena, tahapan pengerjaan aplikasi yang dilakukan satu per satu membuat waktu yang dibutuhkan menjadi lebih lama. Sebagai contoh, tim developer tidak akan bisa melakukan proses *coding* jika tim designer belum menampilkan tampilan desain dari aplikasi.

**Kesimpulan**

Metode waterfall adalah model pengembangan perangkat lunak yang dilakukan secara berurutan dan sangat sistematis. Untuk proses pengerjaannya sendiri, dapat dilakukan dengan berbagai tim proyek yang dibagi sesuai dengan keahlian masing – masing. Setidaknya, terdapat 5 fase dalam tahapan metode waterfall, yaitu analisa kebutuhan, desain, implementasi, integrasi & pengujian, serta operasi dan pemeliharaan.

**2. Metode Prototype**

***Metode Prototype*** – [Prototype](https://en.wikipedia.org/wiki/Prototype) merupakan sebuah metode pengembangan software yang cukup banyak digunakan. Dengan metode ini, pengembang dan pelanggan bisa saling berinteraksi selama proses pengembangan software. Hal ini tentu sangat menguntungkan dan semakin memudahkan dalam pembuatan perangkat lunak.

Metode Prototype adalah teknik pengembangan sistem yang menggunakan prototype untuk menggambarkan sistem sehingga klien atau pemilik sistem mempunyai gambaran jelas pada sistem yang akan dibangun oleh tim pengembang.

Prototype dalam bahasa Indonesia disebut purwarupa (rupa awal). Prototype adalah rupa awal dari sistem yang menggambarkan rupa akhir dari sebuah sistem.

## ****Keuntungan dari Metode Prototype****

Pertama-tama penting untuk memahami metode prototipe yang paling baik digunakan ketika sistem yang diinginkan perlu memiliki banyak interaksi dengan pengguna akhir.

Saat menggunakan model jenis ini, kesalahan biasanya dapat dideteksi lebih cepat dan umpan balik pengguna yang lebih cepat tersedia untuk menghasilkan solusi yang lebih baik. Dalam metodologi ini model kerja dari sistem disediakan, pengguna mendapatkan pemahaman yang lebih baik tentang sistem yang sedang dikembangkan.

Developer bisa bekerja menentukan kebutuhan klien dengan baik, Efisiensi waktu tinggi dalam pengembangan sistem serta Lebih mudah dalam penerapannya karena klien mengetahui apa yang dibutuhkan.

## ****Kekurangan Metode Pembuatan Prototype****

Metode ini dapat meningkatkan kompleksitas. Rencana Anda mungkin mulai melampaui rencana awal Anda. Selain itu, Fokus pada prototipe terbatas dapat mengalihkan pengembang dari analisis lengkap proyek dengan benar. Namun itulah mengapa ada tahap penyempurnaan.

Klien terus menerus menambah requirement dari sistem, pengen dibuatkan yang seperti inilah seperti itulah, sehingga menambah kompleksitas pembuatan sistem. Sistem akan terhambat jika komunikasi kedua belah pihak tidak berjalan secara efektif.

## ****Tahapan Dalam Metode Prototype****

Ada beberapa tahapan dalam metode prototype. Beberapa sumber menyebutkan prototype mempunyai 3,4,5,6 atau 7 tahapan. Dikutip dari guru99 model prototype setidaknya mempunyai 6 tahapan sebagai berikut:

**Tahap 1: Requirements Gathering and Analysis (Analisis Kebutuhan)**

Tahapan model prototype dimulai dari analisis kebutuhan. Dalam tahap ini kebutuhan sistem didefinisikan dengan rinci. Dalam prosesnya, klien dan tim developer akan bertemu untuk mendiskusikan detail sistem seperti apa yang diinginkan oleh user.

**Tahap 2: Quick Design (Desain cepat)**

Tahap kedua adalah pembuatan desain sederhana yang akan memberi gambaran singkat tentang sistem yang ingin dibuat. Tentunya berdasarkan diskusi dari langkah 1 diawal.

**Tahap 3: Build Prototype (Bangun Prototipe)**

Setelah desain cepat disetujui selanjutnya adalah pembangunan prototipe sebenarnya yang akan dijadikan rujukan tim programmer untuk pembuatan program atau aplikasi.

**Tahap 4: User Evaluation (Evaluasi Pengguna Awal)**

Di tahap ini, sistem yang telah dibuat dalam bentuk prototipe di presentasikan pada klien untuk di evaluasi. Selanjutnya klien akan memberikan komentar dan saran terhadap apa yang telah dibuat.

**Tahap 5: Refining Prototype (Memperbaiki Prototipe)**

Jika klien tidak mempunyai catatan revisi dari prototipe yang dibuat, maka tim bisa lanjut pada tahapan 6, namun jika klien mempunyai catatan untuk perbaikan sistem, maka fase 4-5 akan terus berulang sampai klien setuju dengan sistem yang akan dikembangkan.

**Tahap 6: Implement Product and Maintain (Implentasi dan Pemeliharaan)**

Pada fase akhir ini, produk akan segera dibuat oleh para programmer berdasarkan prototipe akhir, selanjutnya sistem akan diuji dan diserahkan pada klien. Selanjutnya adalah fase pemeliharaan agar sistem berjalan lancar tanpa kendala.

**3. Metode spiral**

Model spiral diperkenalkan pertama kali oleh Barry Boehm pada makalahnya yang berjudul Spiral Model of Software Development and Enhancement. Barry Boehm menjelaskan bahawa model spiral merupakan model yang sangat berguna untuk melakukan pembangunan proyek-proyek besar dan prosesnya dilakukan dengan memperhatikan resiko proyek sehingga pada akhirnya akan menghasilkan model proses yang tepat sesuai kebutuhan pengguna.

Model Spiral adalah salah satu metode yang dapat digunakan dalam pengembangan perangkat lunak. Model spiral merupakan penggabungan dari model prototyping dan model waterfall. Model prototyping yang fokus pada penyajian atau presentasi kepada user dengan format input dan output kemudian perangkat lunak akan dievaluasi. Model waterfall yang fokus kepada proses pengembangan perangkat lunak yang sistematis atau berurutan. Model spiral menekankan pada Analisa resiko setiap tahapannya.

Fungsi model spiral adalah untuk melakukan perubahan, penambahan dan pengembangan perangkat lunak dengan memaksimalkan aspek kecepatan dan ketepatan berdasarkan keinginan dan kebutuhan penggunanya.

**Tahapan dalam Spiral Model**

Dalam penerapan Model Spiral, terdapat lima tahapan untuk merealisasikan penggunaannya, yaitu sebagai berikut:

1. Tahap Liason

Tahap ini berhubungan dengan komunikasi antara pihak-pihak yang terlibat dalam pengembangan softaware (seperti: system analyst) dengan pelanggan (user). Tujuannya adalah memperbaiki dan mengembangan software sesuai kebutuhan dan keinginan hingga memuaskan pelanggan.

1. Tahap planning

Tahap perencanaan meliputi estimasi biaya yang digunakan, batas waktu, pengaturan jadwal, identifikasi lingkungan kerja, sumber-sumber informasi untuk melakukan iterasi (Teknik perulangan). Hasil dari tahapan ini adalah dokumen spesifikasi kebutuhan sistem dan bisnis.

1. Tahap analisis risiko

Tahap analisis reisiki berfungsi untuk mengidentifikasi resiko yang berpotensi akan terjadi dan menghasilkan solusi alternatif secara teknis dan manajemen saat strategi mitigasi (upaya untuk mengurangi resiko bencana) direncanakan dan diselesaikan.

1. Tahapan rekayasa (engineering)

Pada tahap rekayasa, beberapa kegiatan ini yang akan dilakukan, yaitu:

* Menguji, coding dan mengembangkan software
* Menginstal software
* Membuat prototype
* Mendesain dokumen
* Meringkas suatu pengujian software
* Membuat laporan atas kekurangan dari software agar segera diperbaiki

1. Tahap evaluasi

Pada tahap evaluasi, system analyst membutuhkan masukan dan tanggapan dari para user dalam mengevaluasi perangkat/produk yang diuji dan memastikan bahwa produk dibutuhkan sesuai ketentuan yang telah dibicarakan diawal dengan user. System analyst memastikan pelanggan puas dengan produk yang akan dihasilkan untuk menjawab persoalan bisnis mereka. Selain itu, system analyst harus tetap memantau resiko yang akan terjadi seperti faktor-faktor yang dapat menyebabkan cost overrun (pembengkakan biaya).

**Kelebihan dalam menggunakan model spiral :**

1. Pembangunan dan perubahan perangkat lunak yang terjadi dapat diselesaikan secara sistematis
2. Mudah dalam mengestimasi biaya karena proses pembuatan prototype yang jelas dan terencana dalam tahapan yang sistematis
3. Manajemen dan analisa risiko yang lebih cepat dan mudah
4. Mudah dalam melakukan perubahan kebutuhan dan dokumentasi
5. Produksi software bisa terjadi lebih cepat

**Kekurangan dalam menggunakan model spiral :**

1. Tidak cocok dan sulit diimplementasikan dalam projek kecil
2. Memakan waktu yang cukup lama
3. Membutuhkan best practice atau pengalaman sebelumnya karena proses yang sangat kompleks
4. Resiko dalam tahap planning cukup besar. Misalnya terjadi perbedaan dalam jadwal pengembangan dan anggaran belanja.