

Anggota Kelompok:

- Rama Sulaiman Nurcahyo (22/492727/TK/53940)
- Daniel Anantadharya Adyawisesa Linan (22/492989/TK/53975)
- Brandon Rafael Lovelyno (22/500359/TK/54847)

## Rekayasa Perangkat Lunak Model Proses Perangkat Lunak

Tabel 1. Ringkasan Perbandingan SDLC

FEATURE	WATERFALL	V-SHAPED	INCREMENTAL	PROTOTYPING	SPIRAL	CONCUR- RENT	RAD	UNIFIED PROCESS
Requirements Specification	Beginning	Beginning	Beginning	Beginning and during iterations	Beginning	Beginning	Time Boxed Release	Evolving throughout iterations
Cost	Low	Expensive	Low	Medium	Expensive	High	Low	High
Simplicity	Simple	Intermediate	Intermediate	Simple	Intermediate	Complex	Very Simple	Complex
Risk Involvement	High	Low	Easily Manageable	Low to Medium	Low	Medium	Very Low	Easily Manageable
Expertise	High	Medium	High	Medium	High	High	Medium	High
Flexibility to Change	Difficult	Difficult	Easy	Very Easy	Easy	Easy	Easy	Easy
User Involvement	Only at Beginning	At the Beginning	Intermediate	High	High	High	Only at Beginning	High
Flexibility	Rigid	Little Flexible	Less Flexible	Very High	Flexible	High	High	High
Maintenance	Least	Least	Promotes Maintainability	Easily maintained	Typical	Moderate	Easily Maintained	Moderate
Duration	Long	According in Project Size	Very Long	Short to Medium	Long	Medium to Long	Short	According in Project Size

### 1. Waterfall Model

- Definisi

Waterfall Model, atau dikenal sebagai *classic life cycle*, merupakan pendekatan sistematis dan sekuensial. Model ini terdiri dari *requirements specification*, *planning*, *modeling*, *construction*, dan *deployment*.

- Kelemahan

- a. Kelemahan utamanya adalah terjadinya kesulitan apabila terjadi perubahan pada proses yang sudah dikerjakan.
- b. Pembagian tahap yang kurang fleksibel menyebabkan kesulitan apabila terjadi perubahan *requirement* oleh customer.
- c. Kesalahan biasanya hanya terdeteksi pada tahap akhir pengujian sehingga dapat berakibat fatal.

- d. Hasil perangkat lunak yang bekerja tidak tersedia hingga bagian akhir dari proses pengerjaan.
- Kelebihan
  - a. Kelebihan utama adalah model ini memiliki tahapan yang jelas dan terdefinisi dengan baik, memudahkan pemahaman dan manajemen proyek.
  - b. Dengan tahapan yang terpisah, manajer proyek dapat mengelola setiap fase secara terpisah dan memantau kemajuan dengan lebih mudah.
  - c. Dengan spesifikasi kebutuhan yang jelas di awal, ekspektasi hasil akhir dapat diatur lebih baik, mengurangi risiko ketidaksesuaian antara hasil akhir dan kebutuhan pengguna.
- Contoh Kasus Penggunaan

Toyota, Oracle Corporation, dan Aplikasi pemerintahan merupakan contoh perusahaan dan proyek yang menerapkan model *waterfall*. Selain itu, model ini cocok untuk proyek yang relatif berskala kecil hingga sedang yang membutuhkan kecepatan menghasilkan produk dan budget yang dapat diprediksi.

## 2. V Model

- Definisi

V Model, atau Verification and Validation Model, adalah salah satu variasi dari Waterfall Model yang menerapkan verifikasi dan validasi dalam setiap tahap pengembangan, dengan aktivitas pengembangan dan pengujian berjalan secara paralel.
- Kelemahan
  - a. Seperti Waterfall Model, kelemahan utama model ini adalah sifatnya yang kurang fleksibel, yaitu kesulitan dalam menyesuaikan perubahan *requirement*.
  - b. Dengan pengujian di setiap tahap, model ini bisa memakan lebih banyak waktu dan biaya dibandingkan model seperti Agile
  - c. Dengan pengujian akhir yang dilakukan setelah seluruh fase pengembangan, pengguna tidak dapat melihat atau menggunakan produk sampai tahap akhir
- Kelebihan
  - a. Dengan pengujian di setiap tahap, kesalahan dapat lebih mudah dideteksi dan diperbaiki sejak awal, mengurangi risiko masalah besar di akhir pengembangan

- b. Model ini juga memiliki struktur yang terorganisir dengan baik, dengan setiap fase pengembangan memiliki aktivitas pengujian yang sesuai
- c. Dengan fokus yang kuat pada verifikasi dan validasi di setiap langkah, model ini memastikan bahwa produk akhir memenuhi standar kualitas yang lebih tinggi
- Contoh Kasus Penggunaan
 

Perangkat lunak sistem kendali otomotif seperti pada Toyota dikembangkan dengan V Model untuk memastikan bahwa semua spesifikasi dipenuhi dan diuji secara menyeluruh pada setiap fasenya karena sistem kendali tersebut harus berfungsi tanpa cacat dalam berbagai kondisi dan setiap kesalahan dapat menyebabkan kegagalan kritis.

### 3. Incremental Model

- Definisi
 

Model ini merupakan metode dari pengembangan perangkat lunak dengan proses desain, implementasi, dan testing dari model dilakukan secara *incremental* (bertambah sedikit demi sedikit seiring waktu) hingga produk selesai dibuat (semua *requirements* telah terpenuhi)
- Kelemahan
  - a. Setiap fase dari iterasinya bersifat *rigid* (kaku) dan tidak bersentuhan dengan iterasi lain. Artinya, setiap langkah harus diselesaikan sebelum langkah berikutnya dimulai.
  - b. Dapat muncul masalah terkait arsitektur sistem karena tidak semua *requirements*-nya terkumpulkan di awal pengembangan.
- Kelebihan
  - a. Model ini memungkinkan tim pengembang untuk menghasilkan *working software* dengan cepat sehingga dapat dievaluasi dan disesuaikan lebih awal pada *life cyclenya*.
  - b. Model ini fleksibel terhadap biaya. Biaya untuk melakukan perubahan pada *requirements* dan *scope* relatif murah.
  - c. Lebih mudah untuk pengetesan dan *debugging* pada iterasi kecil.
  - d. Lebih mudah untuk mengelola risiko karena tiap masalah ditangani pada tiap iterasi.

- e. Lebih mudah diimplementasikan karena terbagi menjadi beberapa iterasi.
- Contoh Kasus Penggunaan : WhatsApp, Windows, dan MS Office.  
Model ini digunakan apabila *requirements* utama dari sistem dapat didefinisikan, *requirements* lain dapat dimunculkan seiring waktu. Model ini cocok apabila produk ingin dipasarkan sesegera mungkin dan terdapat risiko yang tinggi dalam pengembangan perangkat lunak.

#### 4. Prototyping

- Definisi  
Evolutionary Model menghasilkan perangkat lunak dengan versi yang lebih lengkap pada setiap iterasinya. Pada model proses Prototyping, hal ini dilakukan dengan menghasilkan prototipe.
- Kelemahan
  - a. Pembuatan dan pengujian beberapa prototipe dapat menambah biaya dan waktu pengembangan.
  - b. Prototipe sering dibuat dengan kualitas yang tidak konsisten dibandingkan dengan produk akhir, yang dapat mempengaruhi persepsi pengguna tentang produk akhir.
  - c. Pengguna bisa terlalu bergantung pada prototipe dan memiliki ekspektasi yang tidak realistis tentang fitur akhir.
  - d. Iterasi yang berulang dan perubahan yang sering dapat membuat manajemen proyek menjadi kompleks dan sulit untuk diatur.
- Kelebihan
  - a. Pengguna dapat memberikan *feedback* pada prototipe awal sehingga memungkinkan perbaikan dan penyesuaian sebelum produk akhir selesai.
  - b. Memungkinkan perubahan dan penyesuaian yang mudah selama pengembangan ketika terjadi kebutuhan atau *requirement* yang berubah.
  - c. Potensi masalah atau kekurangan dapat diidentifikasi lebih awal dalam proses prototipe, mengurangi risiko kegagalan proyek.
  - d. Prototipe memudahkan komunikasi antara pengembang dan pengguna karena keduanya dapat melihat dan membahas model yang konkret.

- Contoh Kasus Penggunaan

Aplikasi mobile seperti WhatsApp atau Instagram sering menggunakan prototipe untuk menguji dan mengiterasi fitur baru mereka berdasarkan *feedback* pengguna sebelum merilis suatu fitur baru.

## 5. Spiral

- Definisi

Model Spiral adalah model pengembangan yang menggabungkan proses iteratif dari evolutionary model serta proses yang terstruktur oleh waterfall model. Penggabungan tersebut memungkinkan pengembangan berkelanjutan yang terstruktur.

- Kelemahan

- a. Iterasi yang terstruktur, yang terdiri dari analisa, pengembangan, testing, serta feedback, dapat menyebabkan kompleksitas pengerjaan yang tinggi
- b. Proses iterasi menyebabkan kesulitan dalam memelihara dokumentasi software
- c. Karakteristik risk management model memerlukan tingkat keahlian yang tinggi untuk menanggapi setiap resiko pada setiap iterasi

- Kelebihan

- a. Keunikan risk management model ini memungkinkan pengembang untuk melakukan mitigasi masalah dengan tangkas dan cepat
- b. Testing and *refinement* berkelanjutan memungkinkan terjadinya perkembangan yang berkelanjutan
- c. Pengembangan iteratif memungkinkan perubahan *requirements* dapat diintegrasikan tanpa mengganggu jalannya pengembangan software

- Contoh Kasus Penggunaan

Model Spiral digunakan oleh IBM dalam membuat IBM's OS/2. Model ini membantu IBM dalam menangani kompleksitas dan risiko yang terkait dengan pengembangan sistem operasi yang besar dan kompleks.

## 6. Concurrent

- Definisi

Model Concurrent adalah pendekatan dalam pengembangan perangkat lunak dengan beberapa fase dilakukan secara bersamaan atau beriringan (secara *concurrent*).

- Kelemahan

- a. Dengan fase pengembangan yang dilakukan secara bersamaan, diperlukan manajemen proyek yang lebih kompleks dan koordinasi yang baik antar tim.
- b. Dengan banyaknya aktivitas yang berjalan bersamaan, perubahan di satu bagian dapat berdampak pada bagian lain.
- c. Dengan berbagai fase yang berjalan bersamaan, melacak kemajuan keseluruhan proyek bisa menjadi lebih sulit.

- Kelebihan

- a. Model ini sangat fleksibel dalam menanggapi perubahan kebutuhan selama pengembangan. Perubahan pada satu bagian dapat segera diimplementasikan tanpa harus menunggu fase lain selesai.
- b. Dengan berbagai fase pengembangan yang dilakukan bersamaan, waktu keseluruhan untuk menyelesaikan proyek dapat lebih singkat.
- c. Dengan aktivitas yang berjalan secara paralel, masalah atau risiko dapat diidentifikasi dan diatasi lebih cepat.

- Contoh Kasus Penggunaan

Perangkat lunak yang besar dan kompleks seperti Amazon memiliki banyak fitur dan layanan yang dapat dikembangkan secara paralel. Model Concurrent memungkinkan tim yang berbeda untuk bekerja pada berbagai bagian platform seperti pencarian produk, checkout, dan rekomendasi secara bersamaan.

## 7. RAD

- Definisi

RAD Model merupakan pendekatan yang berfokus pada pengembangan software berkualitas tinggi secara cepat. Kecepatan pengembangan dalam model ini dicapai dengan pendekatan *component-based* yang menggabungkan *reusable modules*.

- Kelemahan

- a. RAD Model membutuhkan resource serta komitmen tinggi dari pengembang dan pengguna untuk bekerja secara *rapid*
- b. Developer dituntut untuk dapat melakukan modularisasi sistem dengan tepat
- c. Adanya kemungkinan software yang dihasilkan tidak optimum
- Kelebihan
  - a. Model ini dapat menghasilkan software dalam waktu yang relatif cepat
  - b. Proses iteratif model ini memastikan kesesuaian software dengan kebutuhan pengguna
  - c. Proses iteratif model membuat masalah lebih mungkin terdeteksi oleh feedback pengguna serta *frequent testing*
- Contoh Kasus Penggunaan
 

Software yang menggunakan model ini adalah Microsoft Access, FileMaker Pro, dan OutSystems. Ciri utama software tersebut adalah pemanfaatan antarmuka grafis dan komponen yang dapat digunakan kembali untuk mempercepat pembuatan aplikasi, memungkinkan prototyping cepat dan iterasi berdasarkan umpan balik pengguna.

## 8. Unified Process

- Definisi
 

Unified Process Model adalah pendekatan terstruktur yang berfokus pada pengembangan yang iteratif dengan setiap iterasi mengandung tahap *revisiting* dan *refining* berbasis pada feedback pengguna. Selain itu, model ini juga dikenal oleh pengembangan yang incremental di mana fungsionalitas baru ditambahkan pada setiap increment.
- Kelemahan
  - a. Kebutuhan Unified Process model akan dokumentasi yang jelas dapat memberikan beban tambahan pada pihak pengembang
  - b. Proses iteratif, yang menggunakan dokumentasi, menyebabkan model membutuhkan waktu serta usaha yang besar
  - c. Proses iteratif serta incremental model tidak cocok untuk pengembangan software kecil
- Kelebihan

- a. Pembagian fase yang jelas (Inception, Elaboration, Construction, dan Transition) memastikan project dikerjakan secara sistematis
  - b. Model ini mendukung perkembangan berkelanjutan pada setiap increment
  - c. Tahap iterasi memungkinkan pengembang untuk lebih tangkas dalam mendeteksi serta mengatasi isu
  - Contoh Kasus Penggunaan
- Proyek Eclipse IDE dan beberapa plugin-nya dikembangkan dengan mengikuti prinsip-prinsip Unified Process, termasuk perencanaan iteratif, desain, dan pengujian.



## Referensi

- [1] D. Singh, A. Thakur, A. Chaudhary, “A Comparative Study between Waterfall and Incremental Software Development Life Cycle Model”, *International Journal of Emerging Trends in Science and Technology*, vol. 2, no. 4, pp. 2202-2208, April 2015, Available: <https://journals.indexcopernicus.com/api/file/viewByFileId/174825>.
- [2] R. S. Pressman, “Software Engineering: A Practitioner’s Approach Seventh Edition”, *McGraw-Hill*, 2005. New York.
- [3] I. Somerville. “Software Engineering Ninth Edition”, *Pearson Education*, 2011. Massachusetts.
- [4] A. Mishra, D. Dubey. “A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios”, *International Journal of Advance Research in Computer Science and Management Studies*, vol. 1, no. 5, October 2013.