



**MODUL REKAYASA PERANGKAT LUNAK (RPL)
(CCC-110)**

MODUL 02

Software Development Life Cycle-Classical Waterfall Model

**DISUSUN OLEH
MALABAY,S.KOM,M.KOM**

Universitas
Esa Unggul

UNIVERSITAS ESA UNGGUL

2020

Software Development Life Cycle- Classical Waterfall Model

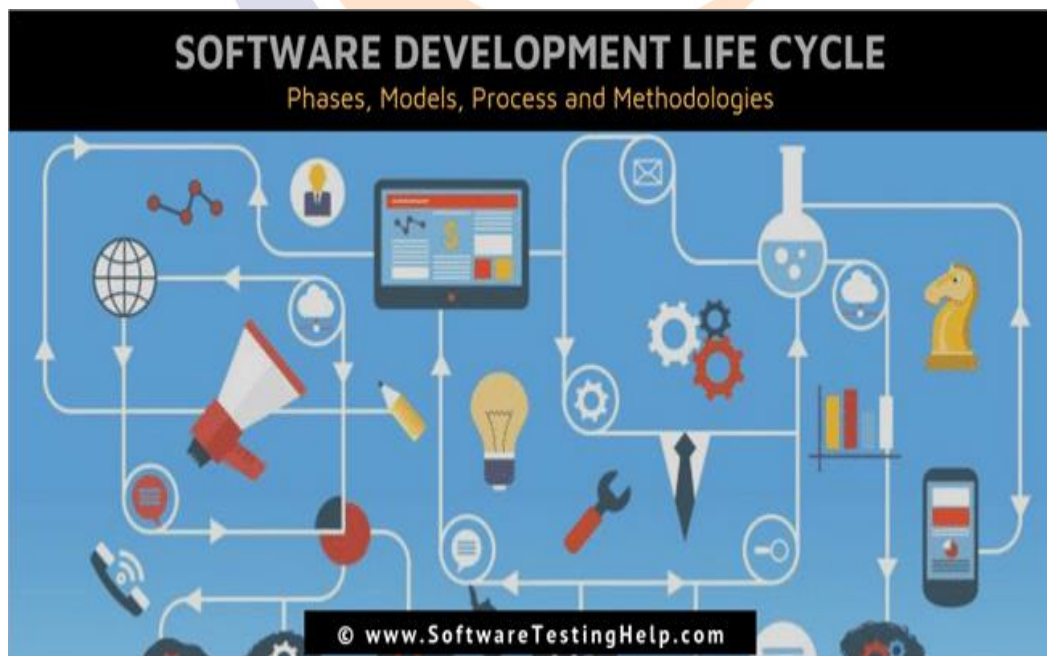
A. Kemampuan Akhir Yang Diharapkan

Setelah mempelajari modul ini, diharapkan mahasiswa mampu : Mahasiswa mampu memahami pengertian *Software Development Life Cycle- Classical Waterfall Model*.

B. Uraian dan Contoh

Siklus Hidup Pengembangan Perangkat Lunak (SDLC) adalah kerangka kerja yang mendefinisikan langkah-langkah yang terlibat dalam pengembangan perangkat lunak di setiap fase. Ini mencakup rencana terperinci untuk membangun, menerapkan, dan memelihara perangkat lunak.

SDLC mendefinisikan siklus lengkap pengembangan yaitu semua tugas yang terlibat dalam perencanaan, pembuatan, pengujian, dan penerapan Produk Perangkat Lunak.



Gambar 1. *Software Development Life Cycle*

SDLC adalah proses yang mendefinisikan berbagai tahapan yang terlibat dalam pengembangan perangkat lunak untuk menghasilkan produk berkualitas tinggi. Tahapan SDLC mencakup siklus hidup lengkap perangkat lunak, yaitu dari awal hingga penghentian produk.

Mengikuti proses SDLC mengarah pada pengembangan perangkat lunak secara sistematis dan disiplin.

Tujuan:

Tujuan SDLC adalah untuk memberikan produk berkualitas tinggi yang sesuai dengan kebutuhan pelanggan.

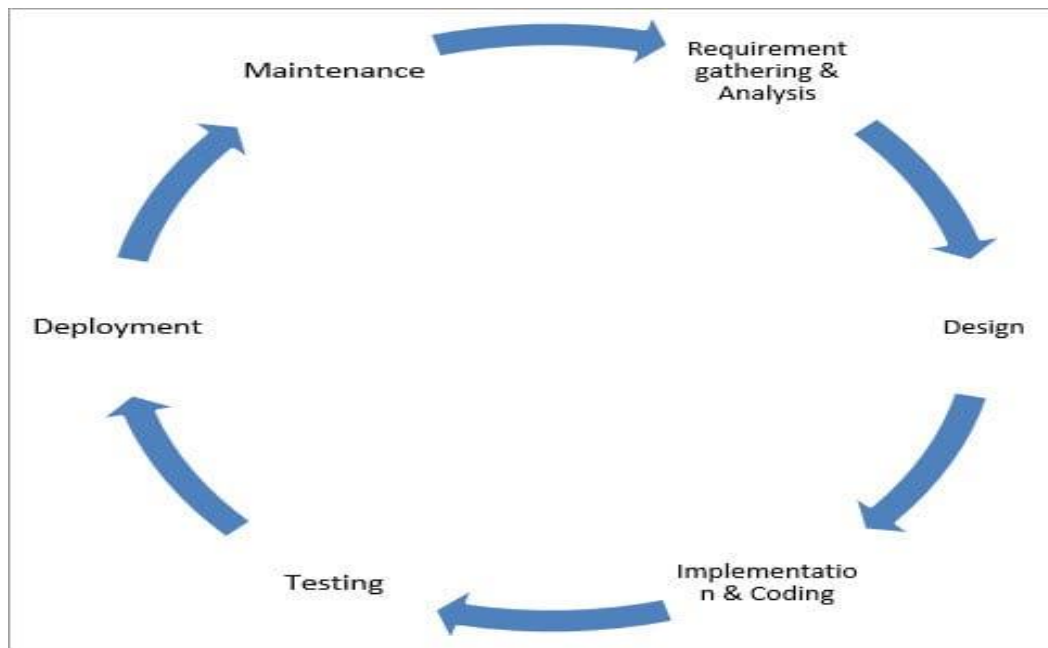
SDLC telah mendefinisikan tahapannya sebagai, Pengumpulan kebutuhan, Desain, Coding, Pengujian, dan Pemeliharaan. Penting untuk mengikuti tahapan untuk menyediakan Produk secara sistematis.

Misalnya, perangkat lunak harus dikembangkan dan tim dibagi untuk mengerjakan fitur produk dan diizinkan bekerja sesuai keinginan. Salah satu pengembang memutuskan untuk mendesain terlebih dahulu sedangkan yang lain memutuskan untuk membuat kode terlebih dahulu dan yang lainnya pada bagian dokumentasi.

Hal ini akan menyebabkan kegagalan proyek karena itu diperlukan pengetahuan dan pemahaman yang baik di antara anggota tim untuk menghasilkan produk yang diharapkan.

Siklus SDLC

Siklus SDLC merupakan proses pengembangan perangkat lunak.



Gambar 2. representation of the SDLC cycle

Tahapan SDLC

Diberikan di bawah ini adalah berbagai fase:

Pengumpulan dan analisis kebutuhan

Rancangan

Implementasi atau pengkodean

Menguji

Penyebaran

Pemeliharaan

Pengumpulan dan Analisis Kebutuhan

Selama fase ini, semua informasi yang relevan dikumpulkan dari pelanggan untuk mengembangkan produk sesuai harapan mereka. Setiap ambiguitas harus diselesaikan hanya dalam fase ini.

Analisis bisnis dan Manajer Proyek mengatur pertemuan dengan pelanggan untuk mengumpulkan semua informasi seperti apa yang ingin dibangun pelanggan, siapa

yang akan menjadi pengguna akhir, apa tujuan produk. Sebelum membangun produk, pemahaman atau pengetahuan inti tentang produk sangat penting.

Rancangan

Dalam fase ini, persyaratan yang dikumpulkan dalam dokumen SRS digunakan sebagai masukan dan arsitektur perangkat lunak yang digunakan untuk mengimplementasikan pengembangan sistem diturunkan.

Implementasi atau Coding

Implementasi / Coding dimulai setelah pengembang mendapatkan dokumen Desain. Desain Perangkat Lunak diterjemahkan ke dalam kode sumber. Semua komponen perangkat lunak diimplementasikan dalam fase ini.

Pengujian

Pengujian dimulai setelah pengkodean selesai dan modul dirilis untuk pengujian. Dalam fase ini, perangkat lunak yang dikembangkan diuji secara menyeluruh dan setiap cacat yang ditemukan diberikan kepada pengembang untuk memperbaikinya.

Pengujian ulang, pengujian regresi dilakukan sampai titik di mana perangkat lunak sesuai dengan harapan pelanggan. Penguji merujuk dokumen SRS untuk memastikan bahwa perangkat lunak sesuai dengan standar pelanggan.

Penerapan

Setelah produk diuji, itu diterapkan di lingkungan produksi atau P3 (pengujian Penerimaan Pengguna) pertama dilakukan tergantung pada harapan pelanggan.

Dalam kasus P3, replika lingkungan produksi dibuat dan pelanggan bersama dengan pengembang melakukan pengujian. Jika pelanggan menemukan aplikasi seperti yang diharapkan, maka tanda tangan disediakan oleh pelanggan untuk ditayangkan.

Pemeliharaan

Setelah penerapan produk di lingkungan produksi, pemeliharaan produk, yaitu jika ada masalah yang muncul dan perlu diperbaiki atau peningkatan apa pun yang harus dilakukan akan ditangani oleh pengembang.

Model Siklus Hidup Pengembangan Perangkat Lunak

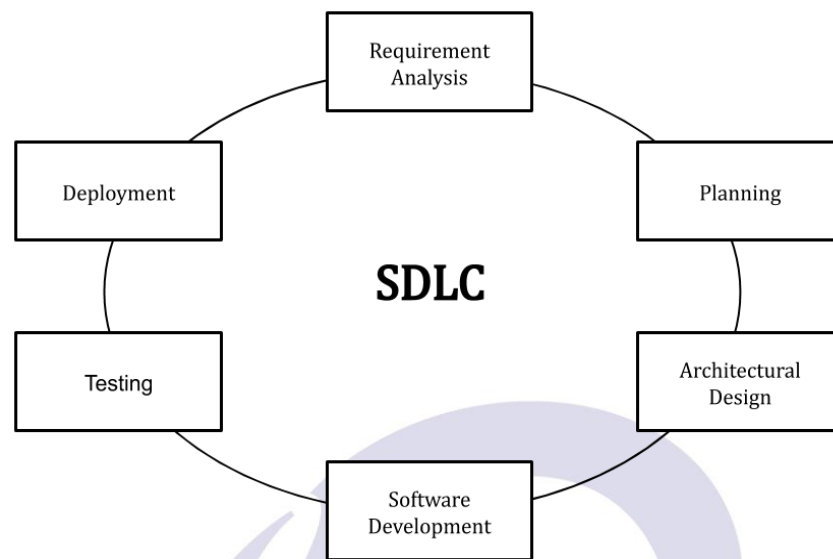
Model siklus hidup perangkat lunak adalah representasi deskriptif dari siklus pengembangan perangkat lunak. Model SDLC mungkin memiliki pendekatan yang berbeda tetapi fase dan aktivitas dasar tetap sama untuk semua model.

Bagaimana SDLC Bekerja

SDLC bekerja dengan menurunkan biaya pengembangan perangkat lunak sekaligus meningkatkan kualitas dan mempersingkat waktu produksi. SDLC mencapai tujuan yang tampaknya berbeda ini dengan mengikuti rencana yang menghilangkan jebakan khas proyek pengembangan perangkat lunak. Rencana itu dimulai dengan mengevaluasi sistem yang ada untuk mengetahui adanya kekurangan.

Selanjutnya, ini mendefinisikan persyaratan sistem baru. Kemudian membuat perangkat lunak melalui tahapan analisis, perencanaan, desain, pengembangan, pengujian, dan penerapan. Dengan mengantisipasi kesalahan yang merugikan seperti gagal meminta umpan balik kepada pengguna akhir atau klien, SDLC dapat menghilangkan pengerjaan ulang yang berlebihan dan perbaikan setelah kejadian.

Penting juga untuk mengetahui bahwa ada fokus yang kuat pada fase pengujian. Karena SDLC adalah metodologi berulang, Anda harus memastikan kualitas kode di setiap siklus. Banyak organisasi cenderung menghabiskan sedikit upaya untuk pengujian sementara fokus yang lebih kuat pada pengujian dapat menghemat banyak pengerjaan ulang, waktu, dan uang. Jadilah cerdas dan tulis jenis tes yang tepat.



Gambar 3. Tahapan Siklus Hidup Pengembangan Perangkat Lunak.

Tahapan dan Teknis bekerja dengan lancar, efisien, dan produktif.

Identifikasi Masalah Saat Ini

“Apa masalah saat ini?” Tahap SDLC ini berarti mendapatkan masukan dari semua pemangku kepentingan, termasuk pelanggan, tenaga penjualan, pakar industri, dan pemrogram. Pelajari kekuatan dan kelemahan sistem saat ini dengan perbaikan sebagai tujuannya.

Rencanakan

Apa yang diinginkan? Dalam tahap SDLC ini, tim menentukan biaya dan sumber daya yang diperlukan untuk melaksanakan persyaratan yang dianalisis. Ini juga merinci risiko yang terlibat dan memberikan sub-rencana untuk mengurangi risiko tersebut.

Dengan kata lain, tim harus menentukan kelayakan proyek dan bagaimana mereka dapat melaksanakan proyek dengan sukses dengan risiko terendah.

Rancangan

“Bagaimana mendapatkan yang diinginkan?” Fase SDLC ini dimulai dengan mengubah spesifikasi perangkat lunak menjadi rencana desain yang disebut Spesifikasi Desain. Semua pemangku kepentingan kemudian meninjau rencana ini dan menawarkan umpan balik dan saran. Sangat penting untuk memiliki rencana untuk mengumpulkan dan memasukkan masukan pemangku kepentingan ke dalam dokumen ini. Kegagalan pada tahap ini hampir pasti akan mengakibatkan pembengkakan biaya paling baik dan kehancuran total proyek paling buruk.

Pembangunan

Pada tahap ini, perkembangan sebenarnya dimulai. Penting bahwa setiap pengembang berpegang pada cetak biru yang disepakati. Selain itu, pastikan Anda memiliki pedoman yang tepat tentang gaya dan praktik kode.

Misalnya, tentukan nomenklatur untuk file atau tentukan gaya penamaan variabel seperti camelCase. Ini akan membantu tim Anda untuk menghasilkan kode yang teratur dan konsisten yang lebih mudah dipahami tetapi juga untuk diuji selama fase berikutnya.

Uji Kode

“Apakah mendapatkan yang diinginkan?” Pada tahap ini, kami menguji cacat dan kekurangan. Kami memperbaiki masalah tersebut sampai produk memenuhi spesifikasi aslinya.

Singkatnya, kami ingin memverifikasi apakah kode tersebut memenuhi persyaratan yang ditentukan.

Penerapan Perangkat Lunak

Pada tahap ini, tujuannya adalah untuk menyebarkan perangkat lunak ke lingkungan produksi sehingga pengguna dapat mulai menggunakan produk. Namun, banyak organisasi memilih untuk memindahkan produk melalui lingkungan penerapan yang berbeda seperti lingkungan pengujian atau pementasan.

Ini memungkinkan setiap pemangku kepentingan untuk bermain dengan aman dengan produk sebelum dirilisnya ke pasar. Selain itu, hal ini memungkinkan terjadinya kesalahan akhir sebelum dirilis produk.

Manfaat SDLC

SDLC yang dilakukan dengan benar dapat memungkinkan kontrol manajemen dan dokumentasi tingkat tertinggi. Pengembang memahami apa yang harus mereka bangun dan mengapa. Semua pihak menyetujui tujuan di muka dan melihat rencana yang jelas untuk mencapai tujuan itu. Semua orang memahami biaya dan sumber daya yang dibutuhkan.

Beberapa kekurangan dapat mengubah implementasi SDLC sebagai penghalang bagi pengembangan. Kegagalan untuk memperhitungkan kebutuhan pelanggan dan semua pengguna dan pemangku kepentingan dapat mengakibatkan pemahaman yang buruk tentang persyaratan sistem di awal. Manfaat SDLC hanya ada jika rencana tersebut diikuti dengan baik.

Metode air terjun cenderung mengambil setiap langkah ini secara bergantian. Keluaran dari satu menjadi masukan untuk langkah berikutnya.

Langkah-langkah berikut kurang lebih sama dari satu metodologi ke metodologi lainnya.

Planning

Tahap perencanaan melibatkan aspek manajemen proyek dan produk. Ini mungkin termasuk:

Alokasi sumber daya (baik manusia dan material)

Perencanaan kapasitas

Penjadwalan proyek

Perkiraan biaya

Penyediaan

Keluaran dari tahap perencanaan meliputi: rencana proyek, jadwal, perkiraan biaya, dan kebutuhan pengadaan. Idealnya, Manajer Proyek dan staf

Pengembangan berkolaborasi dengan tim Operasi dan Keamanan untuk memastikan semua perspektif terwakili.

Requirements

Bisnis harus berkomunikasi dengan tim TI untuk menyampaikan kebutuhan mereka untuk pengembangan dan peningkatan baru. Fase persyaratan mengumpulkan persyaratan ini dari pemangku kepentingan bisnis dan ahlinya.

Arsitek, tim Pengembangan, dan Manajer Produk bekerja dengan ahlinya untuk mendokumentasikan proses bisnis yang perlu diotomatiskan melalui perangkat lunak. Output dari fase ini dalam proyek Waterfall biasanya berupa dokumen yang mencantumkan persyaratan ini. Metode tangkas, sebaliknya, dapat menghasilkan banyak tugas yang harus dilakukan.

Design and Prototyping

Setelah persyaratan dipahami, arsitek dan pengembang perangkat lunak dapat mulai merancang perangkat lunak. Proses desain menggunakan pola yang telah ditetapkan untuk arsitektur aplikasi dan pengembangan perangkat lunak. Arsitek dapat menggunakan kerangka arsitektur seperti TOGAF untuk membuat aplikasi dari komponen yang ada, mempromosikan penggunaan kembali dan standardisasi.

Pengembang menggunakan Pola Desain yang telah terbukti untuk memecahkan masalah algoritmik dengan cara yang konsisten. Fase ini juga dapat mencakup beberapa pembuatan prototipe cepat, juga dikenal sebagai lonjakan, untuk membandingkan solusi guna menemukan yang paling cocok. Output dari fase ini meliputi: rancangan dokumen yang mencantumkan pola dan komponen yang dipilih untuk proyek tersebut.

Software development

Fase ini menghasilkan perangkat lunak yang sedang dikembangkan. Bergantung pada metodologi, fase ini dapat dilakukan dalam "sprint" yang dibatasi waktu,

(Agile) atau dapat dilanjutkan sebagai satu blok upaya (Waterfall.) Terlepas dari metodologi, tim pengembangan harus menghasilkan perangkat lunak yang berfungsi secepat mungkin. Pemangku kepentingan bisnis harus dilibatkan secara teratur untuk memastikan bahwa harapan mereka terpenuhi. Output dari fase ini adalah perangkat lunak fungsional yang dapat diuji.

Testing

Fase pengujian SDLC bisa dibilang salah satu yang paling penting. Tidak mungkin memberikan perangkat lunak berkualitas tanpa pengujian. Ada berbagai macam pengujian yang diperlukan untuk mengukur kualitas:

Kualitas kode

Pengujian unit (uji fungsional)

Tes integrasi

Pengujian kinerja

Pengujian keamanan

Cara terbaik untuk memastikan bahwa pengujian dijalankan secara teratur, dan tidak pernah dilewati untuk kepentingan, adalah dengan mengotomatiskannya. Tes dapat diotomatiskan menggunakan alat Integrasi Berkelanjutan, seperti Codeship, misalnya. Output dari tahap pengujian adalah perangkat lunak fungsional, siap untuk diterapkan ke lingkungan produksi.

Deployment

Fase penerapan, idealnya, merupakan fase yang sangat otomatis. Dalam perusahaan dengan kematangan tinggi, fase ini hampir tidak terlihat; perangkat lunak diterapkan segera setelah siap. Perusahaan dengan masa jatuh tempo yang lebih rendah, atau di beberapa industri yang sangat diatur, prosesnya melibatkan beberapa persetujuan manual. Namun, bahkan dalam kasus tersebut, yang terbaik adalah penerapannya sendiri sepenuhnya otomatis dalam model penerapan berkelanjutan. Alat Application Release Automation (ARA) digunakan di perusahaan menengah dan besar untuk mengotomatiskan penerapan aplikasi ke lingkungan Produksi. Sistem ARA biasanya terintegrasi dengan alat Integrasi

Berkelanjutan. Output dari fase ini adalah rilis ke Produksi perangkat lunak yang berfungsi.

Operations and Maintenance

Fase operasi dan pemeliharaan adalah "akhir dari permulaan," bisa dikatakan. Siklus Hidup Pengembangan Perangkat Lunak tidak berakhir di sini. Perangkat lunak harus dipantau secara konstan untuk memastikan pengoperasian yang benar. Bug dan cacat yang ditemukan dalam Produksi harus dilaporkan dan ditanggapi, yang sering kali memberi umpan balik ke dalam proses. Perbaikan bug mungkin tidak mengalir melalui seluruh siklus, namun, setidaknya proses yang disingkat diperlukan untuk memastikan bahwa perbaikan tidak menimbulkan masalah lain

Model Air Terjun

Model air terjun adalah model pertama yang digunakan di SDLC. Ini juga dikenal sebagai model sekuensial linier.

Dalam model ini, hasil dari satu tahap merupakan masukan untuk tahap berikutnya. Pengembangan fase berikutnya dimulai hanya ketika fase sebelumnya selesai.

Pertama, pengumpulan dan analisis kebutuhan dilakukan. Setelah persyaratan dibekukan maka hanya Desain Sistem yang dapat dimulai. Di sini, dokumen SRS yang dibuat adalah keluaran untuk tahap Persyaratan dan berfungsi sebagai masukan untuk Desain Sistem.

Dalam Perancangan Sistem Arsitektur dan Perancangan Perangkat Lunak, dokumen yang berfungsi sebagai masukan untuk tahap selanjutnya dibuat yaitu Implementasi dan pengkodean

Pada tahap Implementasi dilakukan pengkodean dan perangkat lunak yang dikembangkan menjadi masukan untuk tahap selanjutnya yaitu pengujian.

Pada tahap pengujian, kode yang dikembangkan diuji secara menyeluruh untuk mendeteksi cacat pada perangkat lunak. Cacat dicatat ke alat pelacakan cacat dan diuji ulang setelah diperbaiki. Pencatatan bug, Tes Ulang, Pengujian regresi berlangsung hingga perangkat lunak dalam status go-live.

Pada fase Deployment, kode yang dikembangkan dipindahkan ke produksi setelah persetujuan diberikan oleh pelanggan.

Setiap masalah dalam lingkungan produksi diselesaikan oleh pengembang yang berada dalam pemeliharaan.



Gambar 4. *Waterfall model*

Keuntungan Model Air Terjun:

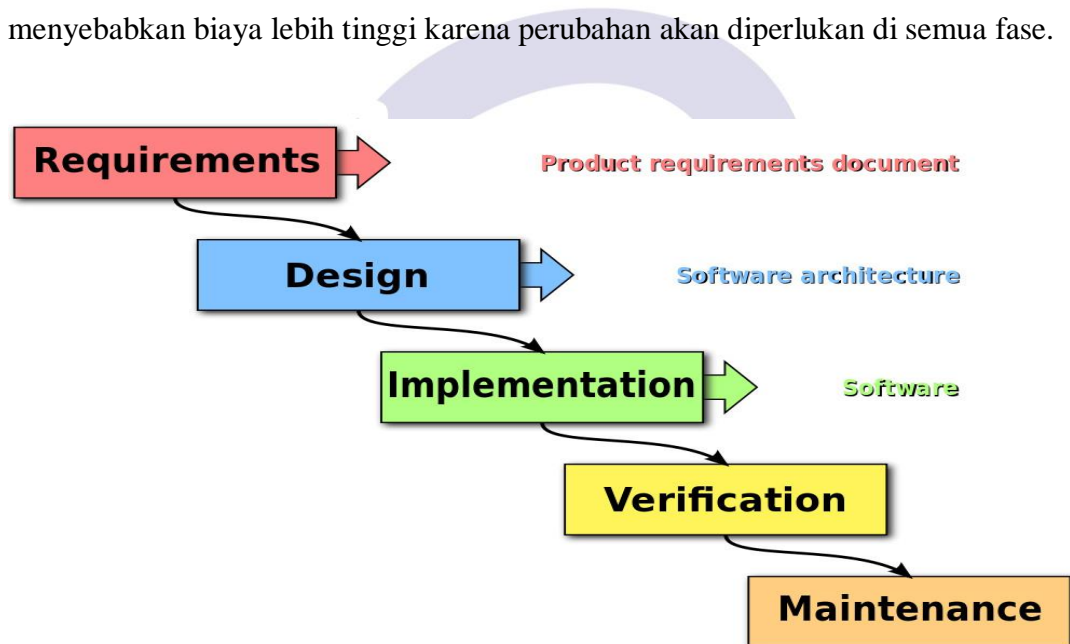
Model air terjun adalah model sederhana yang dapat dengan mudah dipahami dan merupakan model yang semua tahapannya dilakukan secara bertahap.

Hasil dari setiap fase didefinisikan dengan baik, dan ini tidak mengarah pada kerumitan dan membuat proyek mudah dikelola.

Kekurangan model air terjun:

Model air terjun memakan waktu & tidak dapat digunakan dalam proyek berdurasi pendek karena dalam model ini fase baru tidak dapat dimulai hingga fase yang sedang berlangsung selesai.

Model air terjun tidak dapat digunakan untuk proyek yang memiliki persyaratan yang tidak pasti atau di mana persyaratan terus berubah karena model ini mengharapkan persyaratan menjadi jelas dalam tahap pengumpulan dan analisis persyaratan itu sendiri dan setiap perubahan pada tahap selanjutnya akan menyebabkan biaya lebih tinggi karena perubahan akan diperlukan di semua fase.



Gambar 5. *Waterfall model*

Metode pengembangan perangkat lunak air terjun mengikuti jalur yang kaku dan telah ditentukan sebelumnya melalui serangkaian fase. Metode ini diadaptasi dari teknik tradisional. Ironisnya, makalah yang dikreditkan sebagai asal mula metode Waterfall, menggambarannya sebagai cacat fundamental. Metode yang sekarang dikenal sebagai "Air Terjun" secara keliru berasal dari kesalahpahaman tentang karya asli ini. Terlepas dari kenyataan itu, Waterfall menjadi metodologi standar yang sangat umum untuk proyek besar di seluruh dunia.

Metodologi air terjun dimulai dengan fase perencanaan dan desain yang panjang. Setelah dikembangkan, perangkat lunak tersebut kemudian melewati tahapan

pengujian, dan akhirnya digunakan untuk digunakan. Air terjun dianggap oleh banyak orang terlalu kaku untuk beradaptasi dengan perubahan kebutuhan. Itu tidak mendukung umpan balik selama proses, yang mengarah pada penerapan persyaratan yang mungkin telah berubah selama upaya pengembangan. Kelemahan di Waterfall ini menyebabkan pengembangan metodologi yang lebih fleksibel, seperti Agile.

Memudahkan memahami *Waterfall Model* sebagai berikut :

1. Pengembangan perangkat lunak mengalir secara berurutan dari titik awal ke titik akhir.
 2. Proses desain tidak dipecah menjadi model individu
 3. Pelanggan hanya dapat melihat produk di akhir proyek
 4. Model air terjun lebih aman karena sangat berorientasi pada rencana
 5. Semua jenis proyek dapat diperkirakan dan diselesaikan.
 6. Hanya pada akhirnya, seluruh produk diuji. Jika kesalahan persyaratan ditemukan atau ada perubahan yang harus dilakukan, proyek harus dimulai dari awal
 7. Proses pengembangan bertahap, dan fase ini jauh lebih besar daripada iterasi. Setiap fase diakhiri dengan deskripsi mendetail dari fase berikutnya.
 8. Dokumentasi adalah prioritas utama dan bahkan dapat digunakan untuk melatih staf dan meningkatkan perangkat lunak dengan tim lain
 9. Hanya setelah tahap pengembangan, tahap pengujian dijalankan karena bagian-bagian yang terpisah tidak berfungsi penuh.
 10. Semua fitur yang dikembangkan dikirimkan sekaligus setelah fase implementasi yang panjang.
 11. menguji bekerja secara terpisah dari pengembang
 12. Penerimaan pengguna dilakukan di akhir proyek.
 13. Pengembang tidak terlibat dalam persyaratan dan proses perencanaan.
- Biasanya, terjadi penundaan waktu antara tes dan pengkodean

C. Latihan

1. Model pertama yang digunakan di SDLC juga dikenal sebagai model?
2. Kerangka kerja yang mendefinisikan langkah-langkah yang terlibat dalam pengembangan perangkat lunak di setiap fase. Ini mencakup rencana terperinci untuk membangun, menerapkan, dan memelihara perangkat lunak.

D. Kunci Jawaban

1. Sekuensial Linier
2. Siklus Hidup Pengembangan Perangkat Lunak (SDLC)

E. Daftar Pustaka

1. Roger S. Pressman, Software Engineering A Practioner's Apporach, 2014
2. Ian Sommerville, Software Engineering (10th Edition), 2015
3. <https://www.softwaretestinghelp.com/software-development-life-cycle-sdlc/>
4. <https://stackify.com/what-is-sdlc/>
5. <https://raygun.com/blog/software-development-life-cycle/#infographic>
6. <https://www.guru99.com/agile-scrum-extreme-testing.html#2>



Universitas
Esa Unggul