FULLSTACK WEB DEVELOPER

1. Introduction
2. Pengertian

Pengembangan fullstak (Full Stack development) adalah perembangan seluruh aplikasi secara end to end, dari mulai front end hingga ke backend.

1. Scope penting Full Stack Development
2. Front end Development

Menurut saya, front-end development berfokus pada pengalaman pengguna. Seperti desain antarmuka yang estetis dan responsif, serta interaktivitas. Penguasaan HTML, CSS, dan JavaScript merupakan hal yang standar terutama saat menggunakan framework modern seperti React atau Angular yang memudahkan pembuatan aplikasi web dinamis dan efisien.

1. Back-End Development

Back-end development adalah tulang punggung aplikasi, menangani logika bisnis, otorisasi, dan pemrosesan data. Pengembang back-end bertanggung jawab untuk membangun dan memelihara server, API, serta manajemen data yang aman dan efisien. Bahasa pemrograman seperti Node.js, Python, Ruby, dan Java sering digunakan untuk membangun back-end yang handal.

1. Database Management

Database management mengelola penyimpanan, pengambilan, dan manipulasi data yang efisien. Perannya sangat penting dalam memastikan data tersedia, aman, dan dapat diakses sesuai kebutuhan aplikasi. Pengembang harus menguasai sistem seperti SQL dan NoSQL, serta memastikan integrasi yang baik antara database dan aplikasi.

1. Mobile Development

Mobile development adalah proses pengembangan aplikasi yang dapat berjalan di perangkat mobile seperti smartphone dan tablet. Ini mencakup beberapa aspek penting:

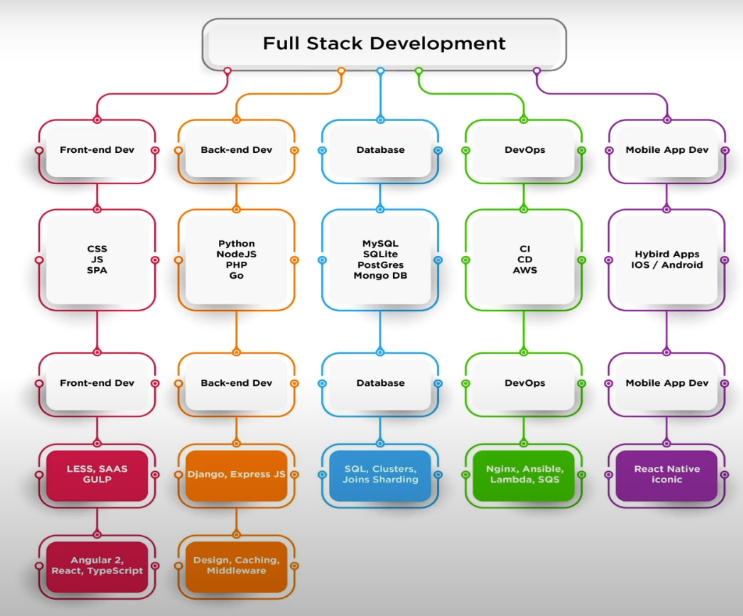
* Platform: Umumnya aplikasi dikembangkan untuk dua platform utama, yaitu Android (menggunakan Java atau Kotlin) dan iOS (menggunakan Swift atau Objective-C).
* Framework: Untuk aplikasi lintas platform, framework seperti React Native dan Flutter memungkinkan pengembangan satu kode dasar yang berjalan di berbagai platform.
* IDE: Alat yang digunakan untuk pengembangan, seperti Android Studio untuk Android dan Xcode untuk iOS.

**II**. Skillset Full Stack Web / Mobile Developer

1. Pengembangan aplikasi End to End

merupakan pndekatan pengembangan perangkat lunak yang mencakup keseluruhan siklus pembuatan aplikasi, dari tahap perencanaan hingga tahap pengujian dan implementasi. Tujuannya adalah untuk menghasilkan aplikasi yang lengkap, fungsional, dan siap digunakan oleh pengguna akhir. Berikut tahapan-tahapan dalam pengembangan aplikasi end-to-end

1. Perencanaan : Identifikasi kebutuhan pengguna dan buat spesifikasi teknis untuk aplikasi.
2. Desain : Kembangkan wireframe dan prototipe UI/UX untuk visualisasi aplikasi.
3. Pengembangan Front-End : Bangun antarmuka pengguna menggunakan HTML, CSS, dan JavaScript atau framework mobile.
4. Pengembangan Back-End : Kembangkan logika bisnis, API, dan pengaturan server untuk aplikasi.
5. Integrasi dan Pengujian : Gabungkan front-end dan back-end, lalu lakukan pengujian fungsional dan non-fungsional untuk memastikan aplikasi berjalan dengan baik.
6. Implementasi : Luncurkan aplikasi ke pengguna, baik melalui App Store atau server produksi.
7. Pemeliharaan dan Peningkatan : Pantau performa aplikasi, perbaiki bug, dan lakukan pembaruan serta peningkatan berdasarkan umpan balik pengguna dan kebutuhan teknologi terbaru.



B. Kolaborasi Efektif

Version control (pengendalian versi) adalah sistem yang memungkinkan pengembang perangkat lunak untuk melacak perubahan pada kode sumber aplikasi selama pengembangan. Contohnya GIT dan mercurial.

Manfaat Version Control untuk Berkolaborasi

1. Rekam Perubahan

Setiap kali pengembang membuat perubahan pada kocie sistem version control merekam detail perubahan tersebut

1. Pencatatan Riwayat

Version control memungkinkan tim untuk melihat riwayat lengkap dari semua perubahan yang terjadi pada proyek dari awal hingga saat ini

1. Pemecahan Konflik

Ketika dua atau lebih pengembang melakukan perubahan pada area kode yang sorna, version control membantu mengidentifikasi dan manyelesaikan konflik

1. Pemulihan Mudah

Version control memungkinkan pengembang untuk memulihkan kode ke versi sebelumnya jika ada masalah atau bug yang terjad, sehingga mengurangi risiko kehilangan pekerjaan

III. Tools Full Stack Web/Mobile Developer

Berikut beberapa tools yang digunakan sebagai full stack developer

1. IDE - Code Editor



1. Version Control - repository



1. Version Control - Git Tools



1. DBMS

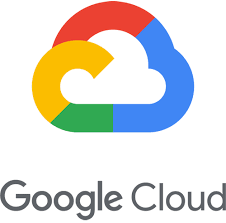
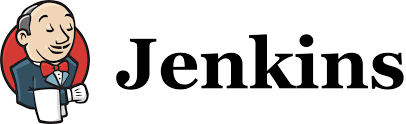
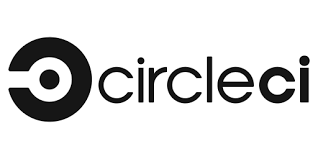


1. API
2. Test dan Debugging

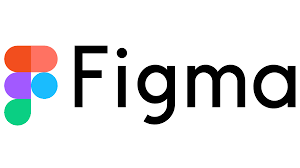


1. Mobile Development

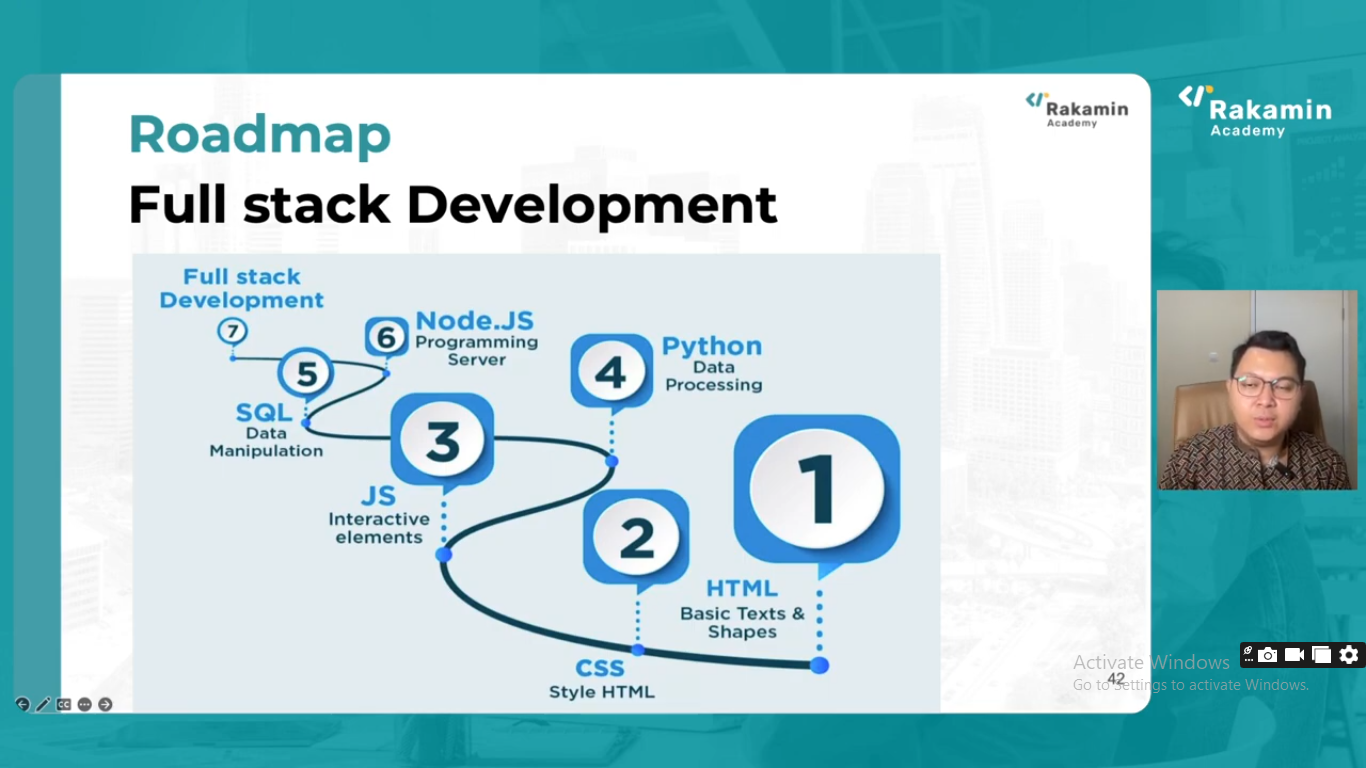


1. Layanan Cloud
2. CI / CD

\

1. Desain UI/UX

Roadmap Full Stack Development



**SDLC & DESAIN THINKING IMPLEMENTATION**

1. Pengertian

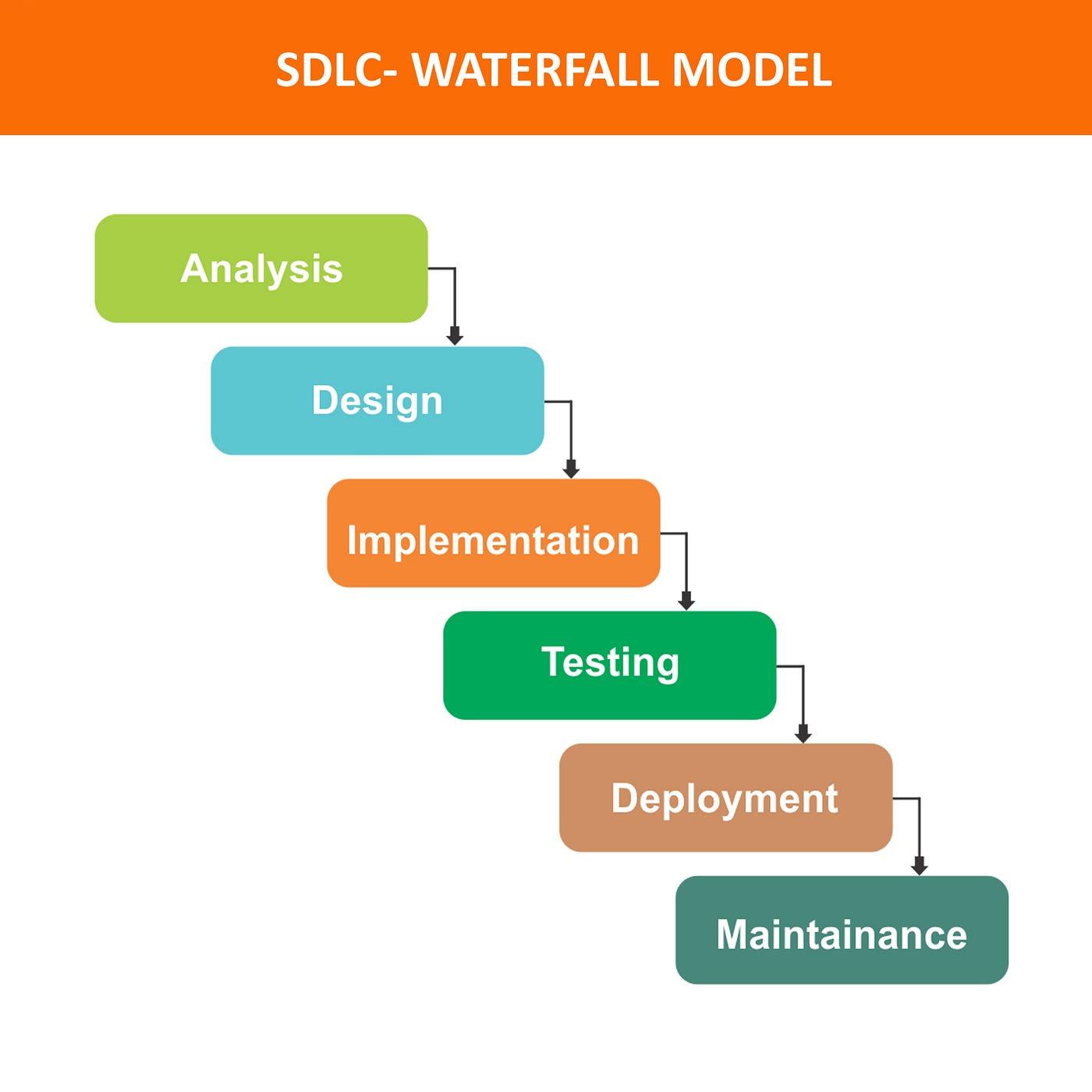
SDLC (Siklus Hidup Pengembangan Perangkat Lunak) adalah rangkaian proses yang terstruktur dan metodologi yang digunakan untuk mengembangkan perangkat lunak dari awal hingga selesai. Berikut penjelasan singkat tahapan dalam SDLC (Software Development Life Cycle):

1. Perencanaan dan Analisis : Menentukan tujuan dan ruang lingkup proyek. Mengumpulkan dan mendokumentasikan kebutuhan fungsional dan non-fungsional perangkat lunak.
2. Desain Sistem : Membuat arsitektur perangkat lunak dan antarmuka pengguna.
3. Pengembangan : Menulis kode dan membangun perangkat lunak.
4. Pengujian : Memverifikasi bahwa perangkat lunak berjalan dengan benar dan bebas dari bug.
5. Implementasi: Mendistribusikan perangkat lunak kepada pengguna.
6. Pemeliharaan: Melakukan perbaikan dan pembaruan setelah peluncuran.

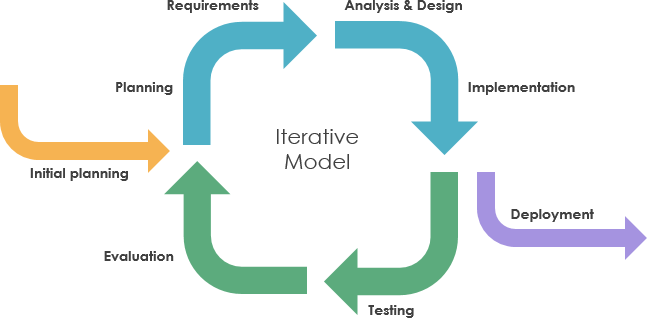
Berikut manfaat dari penggunaan SDLC:

1. Pengelolaan risiko yang llebih baik
2. efisiensi tim dan kolaborasi
3. memenuhi kebutuhan pengguna
4. penghematan biaya
5. peningkatan dokumentas

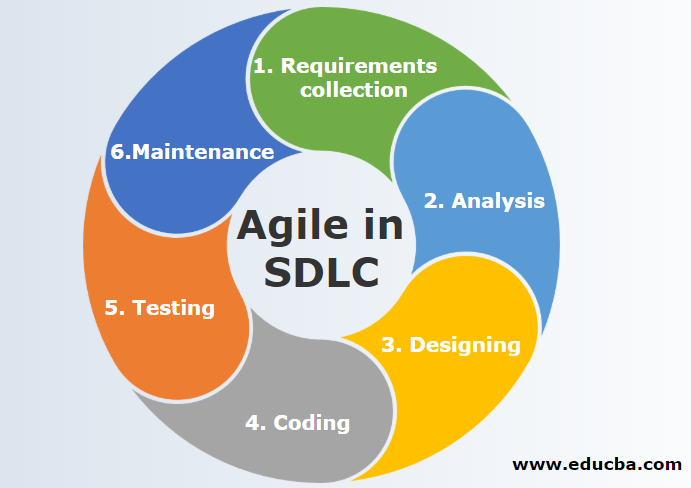
II. Model SDLC

1. Waterfall Model 

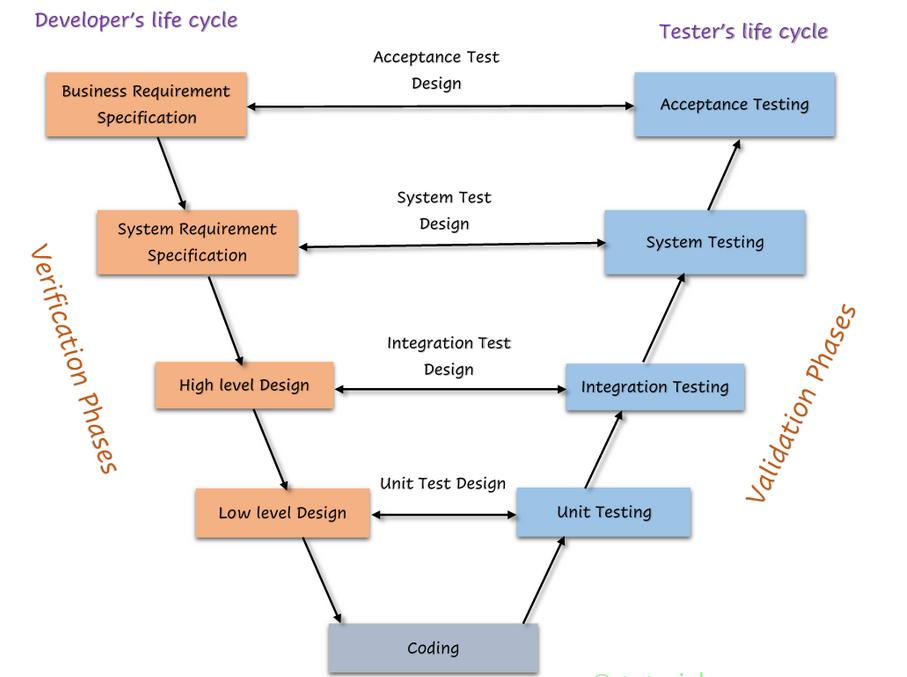
Proses linier di mana setiap tahap harus diselesaikan sebelum tahap berikutnya dimulai. Cocok untuk proyek yang terdefinisi dengan baik.

1. Iterative 

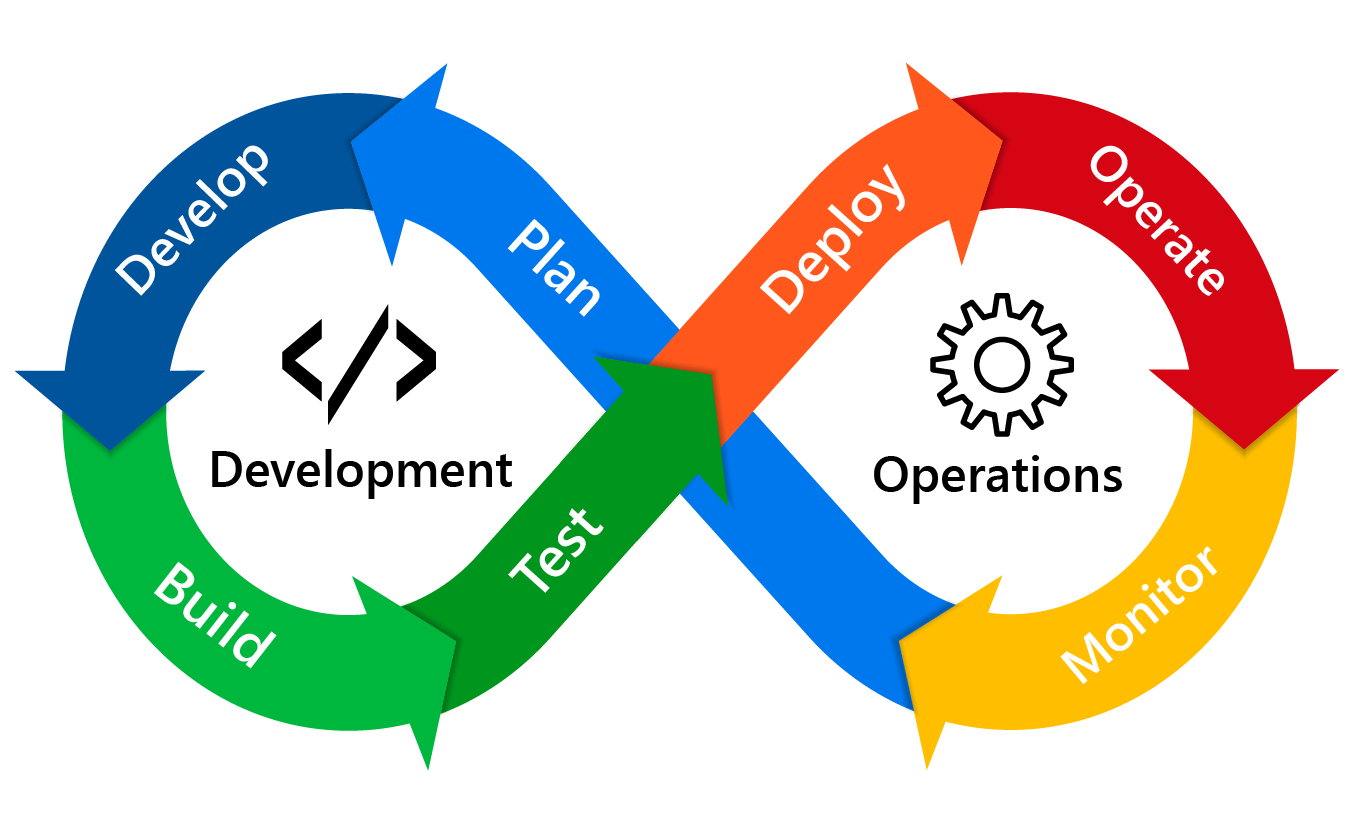
Model Perangkat lunak dikembangkan melalui pengulangan siklus, memungkinkan peningkatan bertahap.

1. Agile

Fleksibel dan adaptif, fokus pada pengembangan bertahap melalui sprint. Cocok untuk proyek yang dinamis dan sering berubah.

1. V-Model

Varian dari waterfall, dengan pengujian paralel di setiap tahap.

1. DevOps

Integrasi pengembangan (Dev) dan operasi (Ops) untuk kolaborasi terus-menerus.

1. Sejarah

Generasi Terminal telah berkembang pesat seiring waktu:

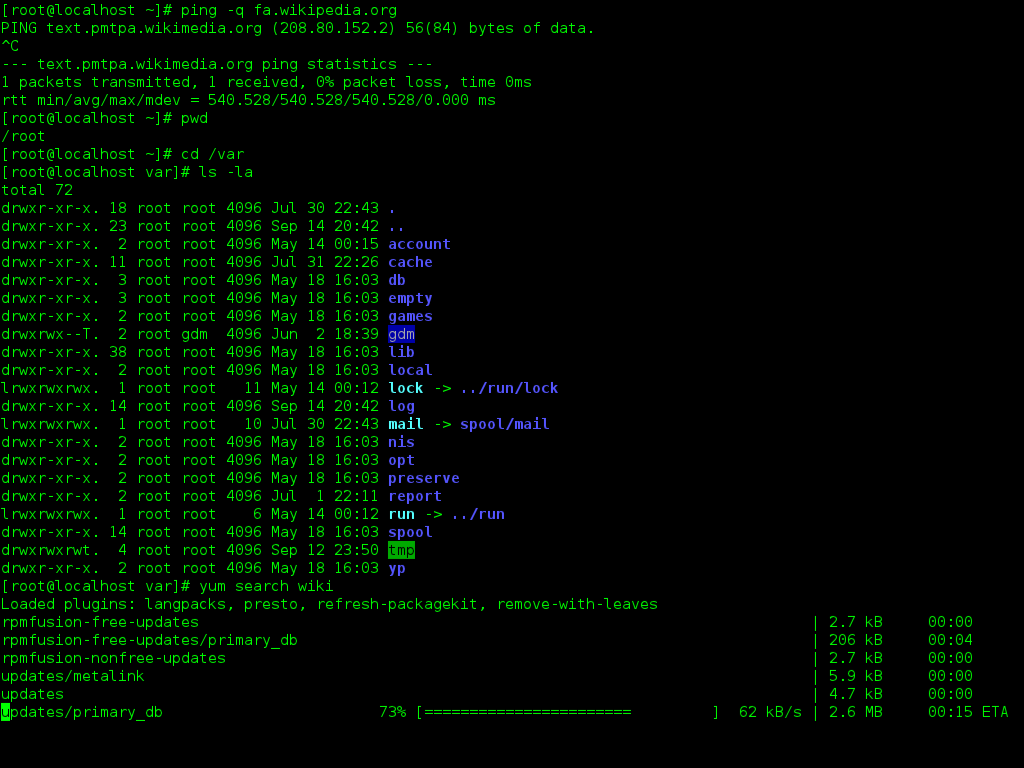
1. Terminal Generasi Pertama (Teletype) 

Pada 1920-an hingga 1960-an, terminal awal berupa teletype (TTY), yang mencetak output di atas kertas. Ini bekerja seperti mesin ketik dengan k

Terminal dan IDEabel.

1. Video Display Terminal (VDT)

Pada 1970-an, terminal berkembang menjadi tampilan visual, dengan layar yang dapat menampilkan teks, menggantikan teletype.

1. Command-Line Interfaces (CLI)

Pada 1980-an dan seterusnya, terminal menjadi perangkat lunak dengan shell seperti Bash atau DOS, memungkinkan pengguna mengirimkan perintah langsung ke sistem operasi.

1. Terminal Modern

Saat ini, terminal adalah aplikasi perangkat lunak berbasis teks yang berjalan di sistem operasi modern seperti macOS, Linux, dan Windows, memungkinkan tugas-tugas administratif, scripting, dan pengelolaan sistem secara efisien.