

## Summary pengenalan software engineer

1. Scope software engineer
  - Front-End Development  
Membangun antarmuka pengguna yang menarik dan interaktif menggunakan HTML, CSS, dan JavaScript.
  - Back-End Development  
Membangun server dan aplikasi yang berfungsi sebagai "otak" dari aplikasi, menerima permintaan dari sisi depan, memproses data, dan memberikan respons yang sesuai.
  - Database Management  
Mendesain dan mengelola basis data untuk menyimpan, mengambil, dan memanipulasi data aplikasi.
2. Pengembangan Aplikasi End to End
  1. kita harus membuat dulu perencanaan dan analisis terhadap aplikasi yang ingin kita buat biasanya ini dikerjakan oleh product manager
  2. Membuat desain UI untuk aplikasi atau website yang ingin di bangun yang biasa dikerjakan oleh UI/UX Designer
  3. kemudian desain UI ini akan diberikan kepada front end untuk dibuat aplikasi depannya.
  4. setelah itu akan dilakukan pengembangan Back-End
  5. setelah itu akan dilakukan quality assurance untuk memastikan aplikasi itu berjalan dengan semestinya.
  6. setelah aplikasinya launching, akan ada pembaharuan atau perbaikan bug untuk memastikan aplikasi tetap relevan.
3. Tools sets Sebagai Full Stack Developer
  - IDE - Code Editor
  - Version Control - Repository (github, gitlab, dan bitbucket)
  - Version Control - Git Tools (sourcetree dan gitlens)
  - DBMS (My sql, oracle, mongoDB, redis, PostgreSQL)
  - API (Postman dan Swagger)
  - tests dan Debugging (jest, mocha, chai, JUnit 5)
  - Mobile Development (React native dan Flutter)
  - Layanan cloud (aws, google cloud azure, dan alibaba cloud)
  - CI/CD (Jenkins dan Circleci)
  - Desain UI/UX (Figma dan sketch)
4. Roadmap Full stack Development
  1. basic Html
  2. style html menggunakan css
  3. interactive elements menggunakan js
  4. data processing (Python)
  5. data manipulation (MySQL)
  6. programming server dengan node.js
5. Tahapan SDLC
  1. Perencanaan dan Analisis : Tahap pertama ini melibatkan identifikasi masalah atau kebutuhan bisnis serta memecahkan permasalahan yang ingin diselesaikan

2. Desain : Tahap kedua melakukan rancangan buat desain interface, desain database dan arsitektur sistem
  3. Pengembangan : Tahap ketiga akan dikerjakan oleh frond end dan back end untuk pengemabngan aplikasi
  4. Pengujian : tahapan ini melakukan pengujian untuk memastikan fungsi dari fitur atau product.
  5. Penerapan : Para pengembang menulis kode untuk menghasilkan produk perangkat lunak yang berfungsi.
  6. Pemeliharaan : tahapan ini untuk memperbaiki bug dan meningkatkan fitur dan menjaga aplikasi tetap relevan.
6. Manfaat Penggunaan SDLC
    1. Prediktabilitas dan Pengendalian Proyek
    2. Peningkatan Kualitas Perangkat Lunak
    3. Efisiensi Tim dan Kolaborasi
    4. Pengelolaan Risiko yang Lebih Baik
    5. Memenuhi Kebutuhan Pengguna
    6. Penghematan Biaya dan Waktu
    7. Meningkatkan Pengawasan dan Evaluasi
    8. Peningkatan Dokumentas
  7. Model Model SDLC
    1. Waterfall Model
    2. V-Shaped Model
    3. Prototype Model
    4. Spiral Model
    5. Iterative Incremental Model
    6. Big Bang Model
    7. Agile Model
  8. Tahapan Design Thinking
    1. Empathize
    2. Define
    3. Ideate
    4. Prototype
    5. Test
    6. Implement
  9. Sejarah Singkat Terminal Dalam dunia teknologi saat ini, terminal tetap menjadi alat esensial bagi pengembang perangkat lunak, administrator sistem, dan pengguna teknis. Meskipun antarmuka grafis semakin canggih, terminal memberikan fleksibilitas dan kekuatan untuk tugas khusus dan otomatisasi di komputer modern.
  10. Command line dasar

Command	Windows	Linux / macOS	Description
List Files	<code>`dir`</code>	<code>`ls`</code>	List files and directories in the current folder
Change Directory	<code>`cd`</code>	<code>`cd`</code>	Change current working directory
Make Directory	<code>`mkdir`</code>	<code>`mkdir`</code>	Create a new directory
Remove Directory	<code>`rmdir`</code>	<code>`rm -r`</code>	Remove a directory
Delete Files	<code>`del`</code>	<code>`rm`</code>	Delete files
Copy Files	<code>`copy`</code>	<code>`cp`</code>	Copy files
Move / Rename	<code>`move`</code>	<code>`mv`</code>	Move or rename files/directories

Command	Windows	Linux / macOS	Description
Display File	<code>`type`</code>	<code>`cat`</code>	Display file content
Display Text	<code>`echo`</code>	<code>`echo`</code>	Display text or enable/disable echoing of commands
List Processes	<code>`tasklist`</code>	<code>`ps`</code>	List running processes
Terminate Process	<code>`taskkill`</code>	<code>`kill`</code>	Terminate processes

Command	Windows	Linux / macOS	Description
Print Directory	N/A	<code>`pwd`</code>	Print the current working directory
Create File	N/A	<code>`touch`</code>	Create an empty file or update timestamp
Change Permissions	N/A	<code>`chmod`</code>	Change file permissions
Change Ownership	N/A	<code>`chown`</code>	Change file ownership
Search in Files	N/A	<code>`grep`</code>	Search for patterns in files
Display Manual	N/A	<code>`man`</code>	Display the manual page for a command
Execute as Admin	N/A	<code>`sudo`</code>	Execute a command with administrative privileges
Ping	N/A	<code>`ping`</code>	Send ICMP echo requests to check network connectivity
Network Config	<code>`ipconfig`</code>	<code>`ifconfig`</code>	Display network interface configurations
Secure Shell	N/A	<code>`ssh`</code>	Connect to a remote host using SSH

11. Control Git : metode yang digunakan untuk melacak dan mengelola perubahan dalam kode sumber atau berkas proyek
12. Git : sistem kontrol versi terdistribusi yang memungkinkan pengembang perangkat lunak untuk melacak perubahan dalam kode mereka, berkolaborasi dengan anggota tim, dan mengelola revisi kode secara efektif.