

Summary

Introduction to Software Engineering

Nurhamidah Mursyidin – Kelompok 7

Full Stack Developer Career Path

1. Introduction Full Stack Web/Mobile Developer

Definisi:

Full Stack Development merupakan pengembangan seluruh aplikasi secara end-to-end, front-end, back-end, dan pada beberapa kasus hingga ke client-side.

Scope:

- **Front-End Development.** Membangun antarmuka pengguna yang menarik dan interaktif dengan menggunakan HTML, CSS, dan JavaScript serta framework seperti React, Angular, Vue.js, atau jQuery.
- **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. Bahasa pemrograman: Node.js, Python, Ruby, Java, PHP, atau C#.
- **Database Management.** Mendesain dan mengelola basis data untuk menyimpan, mengambil, dan memanipulasi data aplikasi. Teknologi database yang digunakan: MySQL, PostgreSQL, MongoDB, atau Firebase.
- **Integration of Front-End and Back-End.** Menghubungkan komponen front-end dengan layanan back-end melalui API (Application Programming Interface) dan menyelaraskan data dan tampilan antara sisi depan dan sisi belakang aplikasi.
- **Version Control and Collaboration.** Menggunakan sistem pengendalian versi, seperti Git untuk mengelola perubahan kode dan kolaborasi dalam tim pengembang dan memastikan kode terus berkembang dengan aman dan sesuai dengan tujuan proyek.
- **Mobile Development.** Memiliki kemampuan untuk mengembangkan aplikasi mobile menggunakan framework seperti React Native, Flutter.

Dasar-Dasar Full Stack Development

<i>Frontend Web Development</i>	<ul style="list-style-type: none">• HTML: mendefinisikan arti dan struktur konten web.• CSS: menata halaman Web.• Javascript: membuat web lebih interaktif. <p>Popular Fronted Framework: React, Vue.js, Angularjs.</p>
<i>Backend Development</i>	<ul style="list-style-type: none">• Bahasa Pemrograman Server-Side Untuk menulis kode di sisi server• Server Framework Framework seperti Express.js untuk Node.js, Flask untuk Python, Ruby on Rails untuk Ruby, Spring untuk Java, dan Laravel untuk PHP,• Database Management Database yang umum digunakan : SQL (MySQL, PostgreSQL, SQL Server) dan NoSQL (MongoDB, Firebase).

<p><i>Database Management</i></p>	<p>Popular Fronted Framework: Express JS, Spring, Laravel, Rails.</p> <ul style="list-style-type: none"> • Database Management System Perangkat lunak yang memungkinkan pengguna untuk mengelola dan mengakses data dalam database. • Tipe Database Dua tipe database utama yang umum digunakan untuk pengembangan aplikasi: SQL (Structured Query Language) atau database relasional dan NoSQL (Not Only SQL) atau database non-relasional. • Bahasa Query SQL adalah bahasa query yang digunakan untuk berinteraksi dengan database SQL. <p>Popular Database Management</p> <p>SQL : PostgreSQL, MySQL</p> <p>Non SQL : mongoDB, redis</p>
<p><i>Mobile Development</i></p>	<ul style="list-style-type: none"> • Platform Mobile Aplikasi mobile dapat dikembangkan untuk berbagai platform, termasuk Android, iOS, dan Windows Phone. Setiap platform memiliki bahasa pemrograman dan lingkungan pengembangan yang khas. Misalnya, aplikasi Android dapat ditulis dalam Java atau Kotlin, sedangkan aplikasi iOS menggunakan Swift atau Objective-C. • IDE (Integrated Development Environment) IDE adalah perangkat lunak yang digunakan untuk mengembangkan aplikasi mobile. IDE menyediakan alat bantu, penyunting kode, pengelola proyek, simulator perangkat, dan fasilitas debugging untuk mempermudah proses pengembangan. <p>Popular Mobile Framework: React Native, Flutter.</p>

2. Skillset Full Stack Web/Mobile Developer

➤ Pengembangan Aplikasi End to End

Definisi:

merupakan pendekatan pengembangan perangkat lunak yang mencakup keseluruhan siklus pembuatan aplikasi, dari tahap perencanaan hingga tahap pengujian dan implementasi.

Tujuan:

Untuk menghasilkan aplikasi yang lengkap, fungsional, dan siap digunakan oleh user.

Tahapan:

- 1) **Perencanaan dan analisis:** pengumpulan data untuk analisis kebutuhan dan riset pasar.
- 2) **Desain:** perancangan UI/UX agar tampilannya lebih menarik, merencanakan arsitektur aplikasi, termasuk pemilihan teknologi, database, dan framework yang sesuai.
- 3) **Pengembangan Front-End:** pengembangan bagian depan aplikasi untuk memberikan tampilan yang interaktif menggunakan HTML, CSS, dan JavaScript.
- 4) **Pengembangan Back-End:** pengembangan sisi server dan logika bisnis aplikasi. Tahap ini melibatkan bahasa pemrograman server-side seperti Node.js, Python, Ruby, atau Java dan menggunakan framework back-end seperti Express.js, Flask, atau Ruby on Rails.

- 5) **Integrasi dan Pengujian:** mengintegrasikan bagian depan dan belakang aplikasi melalui API (Application Programming Interface). Selanjutnya, melakukan pengujian aplikasi untuk memastikan semua fitur berfungsi dengan benar dan mengidentifikasi serta memperbaiki bug yang mungkin ada.
- 6) **Pemeliharaan dan Peningkatan:** memperbaiki bug dan menangani perubahan lingkungan atau kebutuhan bisnis.

➤ **Kolaborasi Efektif – Version Control**

Definisi:

Pengendalian versi (version control) merupakan suatu sistem yang memungkinkan pengembang melacak perubahan pada kode sumber aplikasi selama pengembangan. Memungkinkan kolaborasi yang efisien di antara anggota tim, terutama ketika banyak orang bekerja pada proyek yang sama.

Popular Version Control : git dan mercurial

Manfaat:

- ✓ Mampu merekam perubahan yang dilakukan pengembang
- ✓ Memungkinkan mencatat riwayat dari perubahan yang dilakukan
- ✓ Mampu mengidentifikasi dan menyelesaikan konflik
- ✓ Mampu melakukan pemulihan kode ke versi sebelumnya

Penggunaan Version Control untuk Berkolaborasi:

- Inisialisasi Proyek
- Pengembangan Paralel
- Branching
- Merge
- Pull Request

3. Tools Full Stack Web/Mobile Developer

Tools Sets penting yang bisa digunakan:

- IDE - Code Editor : **visual studio code**. Mampu menjalankan beberapa project.
- Version Control - Repository : **Github, Gitlab, dan Bitbucket**.
- Version Control - Git Tools : **Sourcetree, Gitlens**.
- DBMS : **Postgre SQL, My SQL, Oracle, mongoDB, Redis**.
- API : **Postman, Swagger**.
- Test dan Debugging : **Jest, Mocha Chai, Junit 5**.
- Mobile Development : **React Native, Flutter**.
- Layanan Cloud : **Aws, Google Cloud, Azure**.
- CI/CD : **Jenkins, Circleci**.
- Desain UI/UX : **Figma, Sketch**.

Roadmap:

- 1) HTML - Basic Texts dan Shapes
- 2) CSS - Style HTML
- 3) JS - Interactive Elements
- 4) Python - Data Processing
- 5) SQL - Data Manipulation
- 6) Node.JS - Programming Serving
- 7) Full Stack Development

SDLC & Design Thinking Implementation

1. What is SDLC

Definisi:

SDLC atau Siklus Hidup Pengembangan Perangkat Lunak merupakan rangkaian proses yang terstruktur dan metodologi yang digunakan dalam pengembangan perangkat lunak dari awal hingga selesai.

SDLC terdiri dari serangkaian tahapan yang dijalankan secara runtut untuk memastikan bahwa pengembangan perangkat lunak berjalan dengan baik dan sesuai dengan kebutuhan dan tujuan yang ditentukan.

Siklus SDLC:

- 1) Planning (Perencanaan)
- 2) Analysis (Analisis)
- 3) Design (Desain)
- 4) Development (Pengembangan)
- 5) Testing & Integration (Pengujian dan Penerapan)
- 6) Maintenance (Pemeliharaan)

2. Model-Model Software Development Life Cycle (SDLC)

SDLC Models:

- *Waterfall*. Model SDLC yang linear dan berurutan, cocok untuk proyek dengan persyaratan yang jelas dan stabil.
- *V-Shaped*. Model yang berkaitan erat dengan Waterfall, tetapi menekankan pada pengujian. Cocok untuk proyek dengan fokus pada kualitas tinggi.
- *Prototype*. Model pengembangan perangkat lunak yang bertujuan untuk menciptakan prototipe atau contoh awal sebelum mengembangkan versi finalnya.
- *Spiral*. Gabungan elemen model spiral dengan pendekatan inkremental. Cocok untuk proyek besar dan kompleks dengan banyak risiko.
- *Iterative Incremental*. Model dengan pengulangan siklus pembangunan dan peningkatan perangkat lunak dalam tahapan-tahapan kecil. Cocok untuk proyek dengan waktu dan anggaran yang terbatas.
- *Big Bang*. Model yang kurang terstruktur karena semua tahapan pengembangan dilakukan tanpa perencanaan yang detail, tanpa analisis dan perencanaan yang mendalam. Cocok untuk proyek kecil atau prototyping.
- *Agile*. Model dengan pendekatan kolaboratif dan iteratif yang berfokus pada pengiriman perangkat lunak secara berkala dan inkremental. Cocok untuk proyek dengan lingkungan yang dinamis dan persyaratan yang berubah-ubah.

3. Design Thinking Implementation

Step Design Thinking:

- **Empathize: Understand User Needs**

yaitu tahapan awal design thinking dimana fokus memahami secara mendalam pengguna akhir, diantaranya kebutuhan, keinginan, serta masalah yang mereka hadapi. Hal-hal yang dilakukan antara lain: user research, empathy mapping dan user personas.

➤ **Define: Define the Problem**

yaitu tahap menganalisis informasi yang telah diperoleh pada fase empati untuk menentukan masalah dan menetapkan tujuan yang jelas untuk proyek. Kegiatan kunci meliputi: problem statement dan stakeholder alignment.

➤ **Ideate: Generate Ideas**

yaitu tahap ideasi mendorong pemikiran kreatif dan menghasilkan berbagai solusi potensial. Kegiatan yang dilakukan meliputi: brainstorming sessions dan idea consolidation.

➤ **Prototype: Build Quick and Iterative Solutions**

yaitu tahapan untuk fokus pada menciptakan representasi nyata dari ide-ide yang dipilih. Kegiatan kunci yang dilakukan antara lain: low-fidelity prototypes dan high fidelity prototypes.

➤ **Test: Gather User Feedback**

yaitu tahap pengujian melibatkan pengumpulan umpan balik dari pengguna nyata untuk memvalidasi solusi-solusi tersebut. Kegiatan kunci yang dilakukan meliputi: usability testing dan iterative testing.

➤ **Implement: Develop the Software**

yaitu tahapan untuk menerjemahkan desain ke dalam kode yang sebenarnya kemudian di implementasikan. Kegiatan kunci yang dilakukan meliputi: agile development dan cross-functional collaboration.

Basic Git & Collaborating Using Git

Memahami Version Control Git

Definisi Version Control:

Kontrol versi adalah metode yang digunakan untuk melacak dan mengelola perubahan dalam kode sumber atau berkas proyek.

Definisi Git:

Git merupakan suatu sistem kontrol versi terdistribusi yang memungkinkan pengembang perangkat lunak untuk melacak perubahan dalam kode yang dibuat, berkolaborasi dengan anggota tim lainnya dan mengelola revisi kode secara efektif.

Langkah-langkah untuk memahami kontrol versi dan Git yaitu:

- Sistem Kontrol Versi Terpusat (Centralized Version Control System), yaitu dimana ada satu repositori sentral yang berfungsi sebagai "master" untuk menyimpan seluruh sejarah proyek. Setiap pengembang melakukan perubahan pada salinan lokal, kemudian mengirimkan perubahan tersebut ke repositori sentral. Contoh sistem kontrol versi terpusat adalah Subversion (SVN).
- Sistem Kontrol Versi Terdistribusi (Distributed Version Control System), dimana setiap anggota tim memiliki salinan lengkap dari seluruh repositori. Ini berarti setiap pengembang memiliki salinan lengkap sejarah perubahan, tidak hanya salinan terbaru. Contoh sistem kontrol versi terdistribusi adalah Git, Mercurial, dan Bazaar.