

Week 1 – Fullstack Career Path Introduction | Summary

- **Apa itu Programming?**
 - **Programming** adalah proses menginstruksikan computer untuk melakukan tugas atau serangkaian tugas dalam sebuah Bahasa tertentu.
 - **Bahasa Pemrograman** adalah Bahasa yang digunakan untuk berkomunikasi pada computer agar computer dapat memberikan respon berupa output.
 - **Tiap Bahasa Pemrograman memiliki kelebihan dan kekurangannya masing – masing.** Pemilihan Bahasa pemrograman perlu mempertimbangkan:
 - **Project** yang sedang dikerjakan
 - **Waktu** pengerjaan project yang akan dikerjakan
 - **Kode** yang sudah ada saat ini dalam Bahasa apa
 - **Biaya** yang dikeluarkan untuk menggunakan Bahasa tertentu
 - **Familiarity** dari tim terkait penggunaan Bahasa tertentu
 - **Website** adalah Kumpulan halaman yang berisi informasi tertentu dan dapat diakses dengan mudah melalui internet
 - **Internet** awalnya merupakan penemuan untuk komunikasi militer bernama ARPANet, yang kemudian dikembangkan hingga dapat dimanfaatkan oleh masyarakat luas
 - **Version Control System (VCS)** merupakan metode pengelolaan perubahan pada sebuah kode *software*. Hal ini membantu tim untuk melacak dan mengelola perubahan *source code* dari segi kapan diberlakukannya sebuah perubahan, dan siapa yang membuat perubahan tersebut.
 - **Git, Mercurial, dan Tortoise SVN** adalah Platform VCS yang umum digunakan
- **Pengenalan Web Development**
 - **Web Development** merupakan serangkaian kegiatan yang meliputi:
 - Web Design
 - Web Engineering (Frontend & Backend)
 - Web Content Development
 - Infrastructure & Network Security
 - **Web-Development** umumnya memanfaatkan Bahasa pemrograman:
 - HTML/CSS
 - React.js
 - JavaScript
 - Etc.
- **Frontend Development**
 - **Frontend Development** adalah pengembangan aplikasi pada sisi yang berhubungan langsung dengan klien (*client-side*)
 - **Frontend Development meliputi** pembuatan HTML, CSS, dan JavaScript untuk situs web atau aplikasi berbasis web agar pengguna dapat melihat dan berinteraksi langsung terhadapnya.
 - **HTML (HyperText Markup Language)** merupakan bahasa pemrograman yang digunakan untuk mendefinisikan struktur dan isi dari sebuah web.

- **CSS (Cascading Style Sheet)** adalah bahasa yang digunakan untuk membentuk desain visual dari web yang dikembangkan
 - **JavaScript** adalah bahasa pemrograman yang paling populer di dunia, dan biasa digunakan dalam pemrograman web untuk membuat web lebih interaktif.
- **Framework dalam Frontend Development** meliputi:
 - Angular (TypeScript)
 - Vue.js (JavaScript – TypeScript)
 - React.js (JavaScript – TypeScript)
 - Svelte (Standalone)
- **Scope Frontend Development** meliputi:
 - **Membangun halaman web** berdasarkan desain UI/UX
 - **Optimasi** agar web/aplikasi yang dikembangkan bersifat ringan, user-friendly, dan mudah untuk diakses
 - **Bekerja sama** dengan backend dan desainer UI/UX untuk memasang logic aplikasi/web terhadap tampilan UI/UX yang sudah dibangun
 - **Membuat dokumentasi** sebagai rekam jejak pengembangan aplikasi/web
- **Karir Frontend Developer:**
 - UI/UX Designer
 - iOS Developer
 - Android Developer
 - Backend Developer
 - Project Manager
- **Backend Development**
 - **Backend Development** adalah pengembangan aplikasi pada sisi yang berhubungan dengan server dan struktur informasi internal dari aplikasi (*server-sided*)
 - **RESTful API** adalah tipe *interface* yang digunakan dua sistem komputer untuk bertukar informasi secara aman melalui internet. RESTful API biasa digunakan sebagai sarana komunikasi aplikasi dengan aplikasi internal dan/atau 3rd-party lainnya untuk menjalankan beberapa fungsi dalam aplikasi tersebut
 - **Database** merupakan tempat penyimpanan sistematis data dan/atau informasi yang dibutuhkan saat mengakses suatu *software*.
 - **Relational Database** adalah kumpulan item data yang memiliki hubungan yang telah ditentukan sebelumnya. Berbagai item ini disusun menjadi satu set tabel dengan kolom dan baris.
 - **Platform Relational Database** meliputi: MySQL, MariaDB, ORACLE, SQL Server, ElephantSQL, etc.
 - **Non-Relational Database**, kebalikan dari relational database, merupakan sistem penyimpanan data/informasi dalam bentuk non-tabular. Bentuk database ini lebih fleksibel daripada struktur *relational database* berbasis SQL.

- **Platform Non-relational Database** meliputi: Redis, Cassandra, mongoDB, Firebase, Couchbase, etc.
- **SQL vs NoSQL**
 - **Struktur Data:**
 - **SQL** menggunakan bentuk tabular dengan baris dan kolom yang terstruktur. Semua data harus diatur sesuai dengan skema yang telah ditentukan sebelumnya.
 - **NoSQL** dapat memiliki struktur yang berbeda tanpa mengikuti format tabular, yang berarti penggunaan NoSQL memungkinkan fleksibilitas dalam pengelolaan data yang bervariasi.
 - **Bahasa Query:**
 - **Penggunaan SQL** mengharuskan user untuk menulis *query* berdasarkan bahasa tertentu (ex. SELECT, INSERT, UPDATE, DELETE) untuk mengambil, memasukkan, memperbarui, atau menghapus data dari database.
 - **Penggunaan NoSQL** tidak menggunakan bahasa *query* yang khusus. Beberapa platform NoSQL memiliki bahasa *query*nya sendiri, beberapa lainnya menggunakan *object-based interface* atau menggunakan bahasa pemrograman lainnya yang lebih umum.
 - **Scalability:**
 - **Database SQL** lebih baik digunakan dalam segi skalabilitas vertikal (meningkatkan daya komputasi server).
 - **Database NoSQL** didesain dengan fokus pada skalabilitas horizontal (menambah jumlah server).
 - **Tipe Data dan Penggunaan:**
 - **SQL** cocok untuk data terstruktur yang memiliki relasi yang jelas, seperti data transaksi, laporan keuangan, etc. (Fixed Schema)
 - **NoSQL** cocok untuk data yang tidak memiliki skema yang tetap, seperti data sensor, data media sosial, serta data – data lain yang cenderung akan berubah seiring berjalannya waktu. (Dynamic Schema)
 - **Penyimpanan Data Hierarki:**
 - **SQL** tidak didesain untuk data hierarkis sehingga pengelolaan data hierarkis dilakukan melalui model "adjacency list" atau "nested set" dalam tabel.
 - **NoSQL**, terutama dengan tipe basis data dokumen atau grafik, lebih efektif dalam pengelolaan data hierarkis dengan menyimpan *nested structure* dalam dokumen

atau menggunakan model grafik untuk menggambarkan hubungan yang kompleks.

- **Penggunaan untuk Kueri Kompleks:**

- **SQL** memiliki keunggulan dalam mengatasi kueri kompleks dan analisis data yang melibatkan relasi antara tabel dengan menggunakan bahasa kueri standar yang kuat dan terstruktur.
- **NoSQL** cenderung lebih cocok untuk skenario di mana kueri kompleks kurang umum dan lebih menitikberatkan pada penyimpanan dan pengambilan data yang cepat, terutama dalam lingkungan dengan volume data besar dan skalabilitas horizontal.

- **Framework dalam Backend Development** meliputi:

- Springboot (Java)
- .Net Framework (C#)
- Rails (Ruby)
- Echo (GoLang)
- Express (JavaScript | Node.js)
- Nest (TypeScript | Node.js)
- Laravel (PHP)

- **Cloud Computing** adalah komputasi menggunakan informasi yang diakses dari suatu ruang virtual atau *cloud*. Penggunaan *cloud* berarti file dapat diakses dari server jarak jauh, sehingga dapat diakses kapan saja dimana saja asalkan ada koneksi internet.

- **Jasa Penyedia Cloud** dapat bersifat *public*, *private* maupun *hybrid*. *Public* berarti layanan tersebut dapat diakses oleh masyarakat umum, dan *Private* hanya disediakan untuk orang – orang tertentu saja.

- **Tipe Cloud Computing** meliputi:

- **Software-as-a-Service (SaaS)**

SaaS adalah model layanan *Cloud Computing* dimana *software* atau aplikasi disediakan oleh penyedia melalui internet kepada pengguna. Layanan SaaS diantaranya adalah: Gmail, Office 365, dan Google Colab

- **Infrastructure-as-a-Service (IaaS)**

IaaS adalah model layanan *Cloud Computing* dimana penyedia menyediakan infrastruktur dasar dalam untuk mengirimkan file dari sistem operasi ke server serta *storage* penyimpanan melalui konektivitas berbasis IP. Layanan IaaS diantaranya adalah: Amazon Web Services (AWS) dan Microsoft Azure.

- **Platform-as-a-Service (PaaS)**

PaaS adalah model layanan dimana penyedia menyediakan platform lengkap untuk pengembangan, pengujian, dan hosting aplikasi. Ini mencakup *runtime environment*, *database*, serta *developer tools*. Dengan layanan ini, developer bisa fokus pada kode aplikasi tanpa

perlu mengelola infrastruktur dasarnya. Layanan PaaS diantaranya adalah Heroku dan Google App Engine.

- **Cloud Deployment**
 - **Cloud Deployment** merupakan salah satu metode deployment aplikasi yang memanfaatkan salah satu atau beberapa tipe *Cloud Computing*. Hal ini berarti melakukan perancangan, perencanaan, pengimplementasian, dan pengoperasian beban kerja via *cloud*
- **Platform Cloud Server Deployment** meliputi:
 - Amazon Web Services (AWS)
 - Microsoft Azure
 - Google Cloud
 - Digital Ocean
 - Linode
 - Etc.
- **Scope Backend Developer** meliputi:
 - **Merancang API** untuk konsumsi Frontend
 - **Merancang struktur model data**
 - **Optimasi** agar sistem berjalan dengan cepat dan dapat digunakan dalam skala besar.
 - **Membuat pengujian otomatis pada sistem.**
- **Karir Backend Developer** meliputi:
 - Software Architect
 - DevOps
 - ProjectManager
 - QA
 - DB Administrator