## HASIL JAWABAN NO 1

## RANGKUMAN/SUMMARY MATERI SOFTWARE ENGINEER

Pengembangan Full Stack (Full Stack Development) merujuk pada pengembangan seluruh aplikasi secara end-to-end, dari sisi depan *(fronĞ-end)* hingga sisi belakang *(bacfi-end)* dan, dalam beberapa kasus, hingga sisi klien *(clienĞ-side)*

## SCOPE FULLSTACK DEVELOPER

* **Front-End Development**

Membangun antarmuka pengguna yang menarik dan interaktif menggunakan HTML, CSS, dan JavaScript. Menggunakan framework dan pustaka front-end, seperti React,

Angular, Vue.js, atau jQuery, untuk mempercepat pengembangan dan meningkatkan efisiensi.

## 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. Menggunakan bahasa pemrograman server-side seperti Node.js,

Python, Ruby, Java, PHP, atau C#.

## Database Management

Mendesain dan mengelola basis data untuk menyimpan, mengambil, dan

memanipulasi data aplikasi. Menggunakan teknologi database seperti MySQL, PostgreSQL, MongoDB, atau Firebase

## Dasar-Dasar Frontend Web Development

1. **HTML**

HTML (HyperText Markup Language) adalah blokbangunan paling dasar dari Web. Ini mendefinisikan arti dan struktur konten web.

## CSS

CSS adalah bahasa yang kami gunakan untuk menata halaman Web. Dan biasanya CSS digunakan untuk kosmetik dari sebuah web yang kita punya.

## Javascript

JavaScript adalah bahasa pemrograman paling populer di dunia. JavaScript adalah bahasa pemrograman Web.

JS biasanya digunakan untuk membuat web kita punya lebih interaktif.

# Popular Frontend Framework Full Stack Web Development

* Angular js, react js dan vue js

# Dasar-Dasar Backend Development

* **Bahasa Pemrograman Server-Side**

Bahasa pemrograman seperti Node.js (JavaScript), Python, Ruby, Java, PHP, C , dan lain-lain, digunakan 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

Jenis database yang umum digunakan adalah SQL (MySQL, PostgreSQL, SQL Server) dan NoSQL (MongoDB, Firebase).

# Popular Backend Framework Full Stack Web Development

Express js, spring, Laravel, Rails

# Dasar-Dasar Database Management

* **Database Management System**

Perangkat lunak yang memungkinkan pengguna untuk mengelola dan mengakses data dalam database. DBMS menyediakan antarmuka untuk berinteraksi dengan database

# Tipe Database

Ada dua tipe database utama yang umum digunakan dalam

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. Bahasa query memungkinkan pengguna untuk melakukan operasi seperti SELECT , INSERT , UPDATE DELETE

# Popular Database Management Full Stack Web Development

SQL : PosgreSQL, ORACLE, MYSQL

No SQL : mongodb, redis

# Dasar-Dasar Mobile Development

## 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.

## Popular Mobile Framework Full Stack Mobile Development

React Native dan Flutter

# Pengambangan Aplikasi End to End

Merupakan pendekatan 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.

# Tahap pengembangan Aplikasi End to End

Pada tahap awal perencanaan dan analisis, terdapat kebutuhan untuk mengumpulkan persyaratan dan mendapatkan pemahaman mendalam tentang tujuan aplikasi, target

pengguna, dan lingkungan operasional. Hal ini melibatkan pelaksanaan analisis kebutuhan dan riset pasar untuk mengidentifikasi fitur-fitur utama yang harus disertakan dalam selama fase pengembangan front-end. Tim pengembangan berfokus menjelaskan proses

pengembangan back-end dan front-end suatu aplikasi. Pada fase pengembangan back-end, pengembang menggunakan bahasa pemrograman sisi server seperti Node.js, Python, Ruby, atau Java, bersama dengan kerangka back-end seperti Express.js, Flask, atau Ruby on Rails, untuk membangun server Logika sisi-sisi dan fungsionalitas bisnis aplikasi. Pada fase

pengembangan front-end, tim pengembangan berfokus pada pembuatan antarmuka

pengguna dan interaksi menggunakan HTML, CSS, dan JavaScript, dan dapat menggunakan kerangka kerja front-end seperti React, Angular, atau Vue.js untuk mempercepat proses

pengembangan.mbuatan antarmuka pengguna dan interaksi pengguna aplikasi.

# Penggunaan Version Control untuk Berkolaborasi

## Inisialisasi Proyek

Tim memulai proyek dengan membuat repositori version control. Repositori ini akan menyimpan semua kode sumber, file, dan perubahan yang dilakukan selama

pengembangan.

## Pengembangan Paralel

Setiap anggota tim akan memiliki salinan repositori pada komputernya sendiri. Mereka dapat bekerja secara paralel, membuat perubahan

## Branching

Version control memungkinkan pembuatan cabang (branch) yang terpisah dari kode utama. Ini memungkinkan tim untuk mengisolasi perubahan dan fitur yang sedang dikembangkan

## Merge

Setelah fitur atau perubahan selesai, cabang dapat digabungkan kembali ke cabang utama (biasanya disebut sebagai "merge").

## Pull Request

Di beberapa platform version control seperti GitHub, GitLab, dan Bitbucket, pull request adalah mekanisme yang memungkinkan pengembang untuk mengajukan perubahan

mereka untuk ditinjau oleh anggota tim lain sebelum digabungkan ke cabang utama.

# Materi GIT

Version Control adalah metode yang digunakan untuk melacak dan mengelola perubahan dalam kode sumber atau file proyek. Dalam sistem kontrol versi terpusat, terdapat repositori pusat yang berfungsi sebagai "master" untuk menyimpan seluruh riwayat proyek.

Pengembang membuat perubahan pada salinan lokalnya dan kemudian mengirimkan perubahan tersebut ke repositori pusat. Contoh sistem kontrol versi terpusat adalah Subversion (SVN). Dalam sistem kontrol versi terdistribusi, setiap anggota tim memiliki

salinan lengkap seluruh repositori, artinya mereka memiliki riwayat perubahan yang lengkap, bukan hanya versi terbaru. Contoh sistem kontrol versi terdistribusi termasuk Git, Mercurial, dan Bazaar.

## Apa aitu GIT?

Git adalah 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. Topic ini akan mempelajari menginstal Git pada

berbagai sistem operasi, menginisialisasi repositori Git baru, dan melakukan commit pertama.

**Installing GIT**

Sumber: <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

**Install Git pada Windows**

* Panduan langkah demi langkah untuk menginstal Git di Windows.
* Download dan install GIT pada link berikut
* Memverifikasi instalasi dan mengatur PATH Environment Variables

*NoĞe ĞhaĞ Ğhis is a projecĞ called GiĞ for Windows, which is separaĞe from GiĞ iĞself;*

# Dasar Dasar Command GIT

* git init Menginisialisasi direktori sebagai repositori Git kosong
* git clone Menduplikasi repositori Git yang sudah ada ke direktori lokal.
* git status Menampilkan status perubahan yang belum dikomit di repositori lokal.
* git add Menambahkan perubahan ke area persiapan (staging area) untuk disiapkan menjadi commit
* git commit Membuat commit dari perubahan yang sudah di-staging dan menambahkan pesan commit.
* git push Mengirimkan commit ke repositori jarak jauh (remote repository).
* git pull Mengambil commit terbaru dari repositori jarak jauh dan menggabungkannya ke repositori lokal.
* git branch Menampilkan daftar cabang (branch) yang ada di repositori dan menunjukkan cabang
* git checkout Beralih ke cabang lain atau ke commit tertentu.
* git merge Menggabungkan perubahan dari satu cabang ke cabang aktif.
* git log Menampilkan daftar commit beserta riwayatnya dalam repositori.
* git remote Menampilkan daftar repositori jarak jauh yang terhubung dengan repositori lokal.
* git fetch Mengambil informasi terbaru dari repositori jarak jauh tanpa menggabungkan perubahan
* git diff Menampilkan perbedaan antara versi yang sudah di-staging dengan versi sebelumnya.
* git reset Mengembalikan file yang sudah di-staging ke direktori kerja sebelumnya.

**Hasil Jawaban No 2**



