**HOME WORK WEEK 3**

**Nama : Oktavia Rahmadani**

Introduction to software engineering

Full Stack Development yang bertuju pada pengemabangan seluruh aplikasi secara end-to-end, dari sisi depan (front-end) sehingga sisi belakang (back-end) dan, dalam beberapa kasus, hingga sisi klien (client-side)

* Front-End Development

menggunakan HTML, CSS, dan JavaScript sebagai dasar untuk pembangunan antarmuka, dan pemanfaatan framework dan pustaka front-end seperti React, Angular, Vue.js, atau jQuery.

* Back-End Development

Membangun server dan aplikasi yang berperan sebagai “otak” dari aplikasi, server ini akan menerima permintaan dari sisi depan, memproses data, dan memberikan respons yang sesuai. Menggunakan bahasa pemrograman server-side, Node.js, Python, Ruby, Java, PHP, atau C# untuk mengimplementasinya.

* Database Management

Melibatkan mendesain dan mengelola basis data menggunakan teknologi database seperti MySQL, PostgreSQL, MongoDB, atau firebase. bertujuan untuk menyimpan, mengambil, dan manipulasi data aplikasi.

* Dasar Front-End Web Development

HyperText Markup Language yaitu blokbangunan paling dasar dari webyang mendefinisikan arti dan struktur konten web.

CCS yaitu untuk menata halaman web dan untuk kosmetik dari sebuah web yang kita punya.

Javascript yaitu untuk membuat web kita punya lebih interaktif.

* Dasar Back-End development

Bahasa pemrograman server-side seperti Node.js, Python, Ruby, Java, PHP, C# yang digunakan untuk menulis kode di sisi server.

Server Framework seperti Express.js untuk Node.js Flash untuk Python, Ruby on rails untuk Ruby, Spring untuk java, dan Laravel untuk PHP.

Database Management seperti SQL (MySQL, PostgreSQL, SQL server) dan NoSQL (MongoDB, Firebase)

* Dasar Database Management

DBMS adalah perangkat lunak untuk mengelola data dalam database.

Ada dua tipe utama: SQL (relasional) dan NoSQL (non-relasional). SQL digunakan untuk berinteraksi dengan database melalui bahasa query seperti SELECT, INSERT, UPDATE, DELETE.

* Dasar Mobile Development

Aplikasi mobile dapat dikembangkan untuk berbagai platform seperti Android, iOS, dan Windows Phone, dengan bahasa pemrograman dan lingkungan pengembangan yang khas. Contohnya, Android menggunakan Java atau Kotlin, sedangkan iOS menggunakan Swift atau Objective-C.

IDE (Integrated Development Environment) adalah perangkat lunak yang membantu pengembangan aplikasi mobile dengan menyediakan alat bantu, penyunting kode, pengelola proyek, simulator perangkat, dan fasilitas debugging.

**SDLC**

SDLC (Siklus Hidup Pengembangan Perangkat Lunak) adalah rangkaian proses yang terstruktur dan metodologi yang digunakan untuk mengembangkan perangkat lunak dari awal hingga selesai. Dengan menggunakan SDLC secara efektif, organisasi dapat meningkatkan keberhasilan dan efisiensi dalam mengembangkan aplikasi, memastikan pengiriman produk berkualitas tepat waktu, dan memberikan nilai yang lebih besar bagi pelanggan dan stakeholder.

**Model - Model SDLC**

1. **Waterfall**, model SDLC yang linier dan berurutan. Setiap tahap dalam model ini harus selesai sebelum memulai tahap berikutnya. Tahapannya meliputi analisis, perencanaan, desain, pengembangan, pengujian, implementasi, dan pemeliharaan.
2. **V-Shaped**, model yang terkait erat dengan model waterfall, tetapi menekankan pada pengujian. Tahapan pengujian diwakili oleh garis miring "V", yang berarti bahwa setiap tahap pengembangan memiliki tahapan pengujian yang sesuai.
3. **Prototype,** model pengembangan perangkat lunak yang bertujuan untuk menciptakan prototipe atau contoh awal sebelum mengembangkan versi finalnya. Model ini fokus pada pemahaman kebutuhan pengguna dan mengumpulkan umpan balik untuk memastikan bahwa perangkat lunak akhir sesuai dengan ekspektasi dan persyaratan pengguna.
4. **Spiral,** menggabungkan elemen model spiral dengan pendekatan inkremental. Setiap siklus spiral membangun pada inkrementasi sebelumnya, menghasilkan perangkat lunak yang semakin berkembang dengan fitur yang lebih banyak setiap siklusnya.
5. **Iterative Incremental Model**, melibatkan pengulangan siklus pembangunan dan peningkatan perangkat lunak dalam tahapan-tahapan kecil. Setiap iterasi menambahkan lebih banyak fitur hingga produk akhir mencapai tingkat kesempurnaan yang diinginkan.
6. **Big Bang Model**, model yang kurang terstruktur, di mana semua tahapan pengembangan dilakukan tanpa perencanaan yang detail. Pengembangan dimulai tanpa melakukan analisis dan perencanaan yang mendalam.
7. **Agile Model**, pendekatan kolaboratif dan iteratif yang berfokus pada pengiriman perangkat lunak secara berkala dan inkremental. Tim bekerja dalam sprint (iterasi singkat) dan selalu terbuka untuk perubahan persyaratan pengguna

**Steps Design Thinking**

1. Empathize: Understand User Needs

2. Define: Define the Problem

3. Ideate: Generate Ideas

4. Prototype: Build Quick and Iterative Solutions

5. Test: Gather User Feedback

6. Implement: Develop the Software

C. Basic Git & Collaborating Using Git

**Terminal and IDE**

Terminal adalah antarmuka teks yang digunakan untuk berinteraksi dengan

sistem operasi melalui baris perintah. Merupakan alat yang kuat untuk

mengelola proyek perangkat lunak, mengedit berkas, menjalankan perintah,

dan lainnya. Beberapa IDE (Integrated Development Environment) memiliki

integrasi dengan terminal, memungkinkan pengembang untuk menjalankan

perintah terminal dari dalam IDE.

**Installing, Initializing, and Committing GIT**

Git adalah sistem kontrol versi terdistribusi yang memungkinkan

pengembang melacak perubahan dalam kode mereka. Instalasi Git dapat

dilakukan di berbagai platform seperti Windows, Linux, dan macOS. Perintah

dasar Git termasuk ‘git init’ (menginisialisasi repositori), ‘git clone’

(menduplikasi repository), ‘git add’ (menambahkan perubahan ke staging

area), ‘git commit’ (membuat commit), dan lainnya.

**Collaborating Using Git**

Git memungkinkan kolaborasi antara pengembang dalam pengelolaan proyek

perangkat lunak. Langkah-langkah kolaborasi melibatkan mengundang

kolaborator, membuat branch, membuat perubahan, mengelola konflik, dan

mengirimkan permintaan tarik (pull request) untuk menggabungkan

perubahan. Kolaborator dapat bekerja di cabang (branch) yang berbeda

untuk menghindari konflik langsung dengan perubahan yang dilakukan oleh

orang lain.