

Nama : Setiawan

Tugas : Homework Intro to Software Engineering

FULLSTACK

Definisi :

Pengembangan Full Stack (Full Stack Development) merujuk pada pengembangan seluruh aplikasi secara end-to-end, dari sisi depan (front-end) hingga sisi belakang (back-end) dan, dalam beberapa kasus, hingga sisi klien (client-side)

Scope :

- Frontend development
 - Basic : HTML, CSS, Javascript
 - Framework : React, Vue.js, AngularJs
- Backend development
 - Server side language : Node.js (Javascript), Python, Ruby, Java, PHP, C#, dll
 - Framework : Express.js untuk Node.js, Flask untuk Python, Ruby on Rails untuk Ruby, Spring untuk Java dan Laravel untuk PHP
- Database management
 - Tipe : SQL (Structured Query Language) dan NoSQL (Not Only SQL)
 - SQL : MySQL, PostgreSQL, Oracle
 - NoSQL : MongoDB, Redis

PENGEMBANGAN APLIKASI END TO END

Tahapan :

1. Perencanaan dan analisis
2. Desain
3. Pengembangan Frontend
4. Pengembangan Backend
5. Integrasi dan pengujian
6. Pemeliharaan dan peningkatan

Kolaborasi efektif dengan version control

Tools : Git & Mercurial

Manfaat version control :

1. Rekam perubahan
2. Pencatatan riwayat
3. Pemecahan konflik
4. Pemulihan mudah

Toolset sebagai Fullstack developer :

1. Version control - repository : Github, Gitlab, Bitbucket
2. Version control - Git tools : Sourcetree, Gitlens
3. DBMS : MySQL, PostgreSQL, Oracle, MongoDB, Redis
4. API : Postman & Swagger
5. Test & debugging : Jest, Mochachai, Junit5
6. Mobile development : React native & Flutter
7. Layanan cloud : AWS, Google cloud, Azure

8. CI/CD : Jenkins, CircleCI
9. Desain UI/UX : Figma & Sketch

Roadmap Fullstack development :

- HTML : Basic text & shapes
- CSS : Style html
- JS : Interactive elements
- Python : Data processing
- SQL : Data manipulation
- Node.js : Programming server
- Fullstack development

SDLC (Siklus Hidup Pengembangan Perangkat Lunak)

adalah rangkaian proses yang terstruktur dan metodologi yang digunakan untuk mengembangkan perangkat lunak dari awal hingga selesai. SDLC terdiri dari serangkaian tahap yang saling terkait dan dilakukan secara berurutan untuk memastikan bahwa pengembangan perangkat lunak berjalan dengan baik dan sesuai dengan kebutuhan dan tujuan yang ditentukan.

Siklus SDLC :

1. Perencanaan dan analisis
2. Desain
3. Pengembangan Frontend
4. Pengembangan Backend
5. Integrasi dan pengujian
6. Pemeliharaan dan peningkatan

Manfaat Penggunaan SDLC :

1. Prediktabilitas dan Pengendalian Proyek
2. Memenuhi Kebutuhan Pengguna
3. Peningkatan Kualitas Perangkat Lunak
4. Penghematan Biaya dan Waktu
5. Pengelolaan Risiko yang Lebih Baik
6. Meningkatkan Pengawasan dan Evaluasi
7. Efisiensi Tim dan Kolaborasi
8. Peningkatan Dokumentasi

Model Model SDLC

- Waterfall Model
- Iterative Incremental Model
- V-Shaped Model
- Big Bang Model
- Prototype Model
- Agile Model
- Spiral Model

Implementasi design thinking

Langkah-langkah :

- Empathize: Understand User
 - User Research
 - Empathy Mapping
 - User Personas
- Define: Define the Problem Test: Gather User Feedback
 - Problem Statement
 - Stakeholder Alignment
- Ideate: Generate Ideas
 - Brainstorming Sessions
 - Idea Consolidation
- Needs Prototype: Build Quick and Iterative Solutions
 - Low-Fidelity Prototypes
 - High-Fidelity Prototypes
- Test: Gather User Feedback
 - Usability Testing
 - Iterative Testing
- Implement: Develop the Software
 - Agile Development
 - Cross-Functional Collaboration