

Damar Galih

Homework

Introduction to Software Engineering

Rakamin IT Full Stack Developer

MSIB Batch 7

Summary Introduction to Software Engineer

a. Definisi dan Scope Fullstack

Menurut saya Fullstack mengacu pada keseluruhan teknologi yang digunakan dalam pengembangan sebuah software, mencakup **front end (sisi klien)** dan **back end (sisi server)**. Front end melibatkan elemen visual dan interaktif yang dilihat dan digunakan oleh pengguna, seperti desain, layout dan interaksi dengan HTML, CSS serta JavaScript. Sementara itu, back end menangani logika bisnis, pengelolaan server, database dan API, yang memungkinkan aplikasi berfungsi secara dinamis. Scope full stack mencakup penggunaan teknologi **front end**, **server-side**, **manajemen database** serta **DevOps** untuk deployment dan pengelolaan server.

b. Pengembangan Software End-to-End

Pengembangan software end-to-end adalah proses menyeluruh dalam menciptakan sebuah software, mulai dari **perencanaan** hingga **implementasi**, **pengujian** dan **deployment**. Setiap tahapan memiliki tujuan dan peran spesifik untuk memastikan software yang dikembangkan berjalan dengan baik dan sesuai kebutuhan pengguna. Berikut saya jabarkan tentang tahapan-tahapannya:

1. Perencanaan (Planning)

Dalam tahapan pertama, hal yang harus dilakukan adalah pengumpulan kebutuhan (requirements gathering) untuk memahami apa yang diinginkan pengguna dan mendefinisikan tujuan proyek. Tujuannya tentu untuk merumuskan solusi yang tepat yang akan diimplementasikan di dalam software.

2. Analisis Kebutuhan (Requirements Analysis)

pada tahap kedua ini, Kebutuhan pengguna dipecah menjadi spesifikasi teknis yang lebih detail, seperti fitur, fungsionalitas dan alur kerja sistem. Ini dilakukan untuk mempermudah tim pengembang dalam mengembangkan software nantinya.

3. Perancangan (Design)

Setelah kebutuhan terdefinisi, kita masuk pada tahap perancangan yang mencakup desain arsitektur software, database, antarmuka pengguna (UI/UX). Dalam tahap ini dapat berupa blueprint yang akan diimplementasikan.

4. Pengembangan (Development)

Tim pengembang mulai menulis kode dan mengimplementasikan desain yang telah direncanakan. Pengembangan dilakukan pada bagian front end (tampilan dan interaksi pengguna) dan back end (logika bisnis, database, server).

5. Pengujian (Testing)

Setelah pengembangan selesai, software diuji untuk menemukan bug atau kesalahan serta memastikan bahwa sistem berjalan sesuai dengan kebutuhan yang diharapkan. Tahap ini mencakup unit testing, integration testing dan user acceptance testing (UAT).

6. Deployment

Pengujian selesai dan software stabil, aplikasi di-deploy ke lingkungan produksi agar dapat digunakan oleh pengguna akhir. Ini melibatkan pengaturan server, konfigurasi sistem, dan penanganan manajemen versi.

7. Pemeliharaan (Maintenance)

Proses pemeliharaan dimulai, mencakup pembaruan (updates), perbaikan bug, serta peningkatan fitur berdasarkan feedback pengguna.

c. Beberapa Teknologi Penyusun Fullstack Development

1. Front End

HTML, CSS dan JavaScript, serta framework seperti React, Vue.js, dan Angular untuk menciptakan tampilan dinamis dan interaktif.

2. Back End

Node.js, Python (Django, Flask), Ruby on Rails, dan PHP (Laravel) untuk menangani operasi server-side.

3. Database

MySQL, PostgreSQL, MongoDB, dan SQLite untuk menyimpan dan mengambil data dari back-end.

4. DevOps

Docker, Kubernetes, AWS, Jenkins, dan CI/CD pipelines untuk otomatisasi proses build, testing, dan deployment.

5. Mobile Apps

React Native, Flutter, Swift (iOS), dan Kotlin (Android) untuk membuat software yang dapat berjalan di perangkat seluler, baik native maupun cross-platform.

d. Version Control System (Collaboration)

Version control system (VCS) adalah perangkat lunak yang digunakan untuk **melacak dan mengelola perubahan pada source code dari waktu ke waktu**, juga dikenal sebagai **revision control** atau **source control system**. Version control system memungkinkan developer melacak setiap modifikasi kode, menemukan kesalahan, serta membandingkan perubahan dengan versi sebelumnya. Berikut beberapa fungsi dari version control system:

1. Pelacakan.

2. Riwayat Dokumen

3. Branching and Merging

Developer dapat mengerjakan suatu dokumen yang sama secara independen dalam waktu bersamaan. Masing-masing kontributor yang bekerja pada alur perubahan independen disebut sebagai cabang (branch).

4. Pengurangan Duplikasi dan Kesalahan

5. Management Overview

Memantau penulisan kode, perubahan kode, serta dampak perubahan tersebut terhadap source code yang bersangkutan.

Penggunaan version control system umumnya dilakukan melalui terminal. Salah satu contoh dari version control system adalah git. Berikut beberapa command basic yang sering digunakan di Git untuk collaboration:

1. Inisialisasi Git di Project

```
git init
```

Menginisialisasi repository Git baru di direktori proyek yang ada.

2. Cloning a repository

```
git clone https://github.com/GusSurya031/finpro-msib-7-kelompok-1.git
```

Melakukan cloning sebuah repository pada github ke local.

3. Checking status

```
git status
```

Menampilkan status file, termasuk file yang sudah diubah, yang belum di-commit dan yang siap di-commit.

4. Adding file to stages area

```
git add damar-summary.txt
```

Atau untuk menambahkan semua file:

```
git add .
```

Menambahkan file atau perubahan yang diinginkan ke staging area sebelum di-commit.

5. Commit

```
git commit -m "Restructure: Move all files to damar-homework folder"
```

Commit perubahan yang sudah ada di staging area dengan pesan yang menjelaskan perubahan tersebut.

6. Push to repository github

```
git push origin damar-homework
```

Mengirim commit dari branch lokal ke repository remote sesuai dengan branchnya.

7. Pull

```
git pull origin main
```

Mengambil perubahan terbaru dari branch di repository remote dan menggabungkannya ke branch lokal.

8. Branching

```
git branch
```

Menampilkan daftar branch dan menunjukan branch aktif sekarang.

9. Checkout Branching

```
git checkout damar-homework
```

Melakukan perpindahan branch ke branch yang dituju.

10. Merge

```
git merge main
```

Melakukan merger atau penggabungan dari branch saat ini ke branch yang dituju.

Masih banyak command line lainnya untuk menunjang pengembangan software, informasi lengkapnya dapat diakses melalui <https://education.github.com/git-cheat-sheet-education.pdf> atau <https://www.geeksforgeeks.org/git-cheat-sheet/>.

Untuk melakukan collaboration, berikut beberapa langkah mudahnya:

1. Lakukan Fork (untuk proyek open source) atau clone repository
2. Buat branch baru untuk media perubahan pada branch sendiri agar tidak mengganggu branch lainnya
3. Lakukan perubahan sesuai dengan yang diinginkan
4. Stage dan commit perubahan yang dilakukan sebelumnya
5. Push branch ke repository yang ada di github (repository remote)
6. Lakukan pull request dan review jika diperlukan
7. Lakukan merger ke branch utama

Link Github: <https://github.com/GusSurya031/finpro-msib-7-kelompok-1>

<https://github.com/damar-glh>

Terima Kasih 😊