

Presentasi

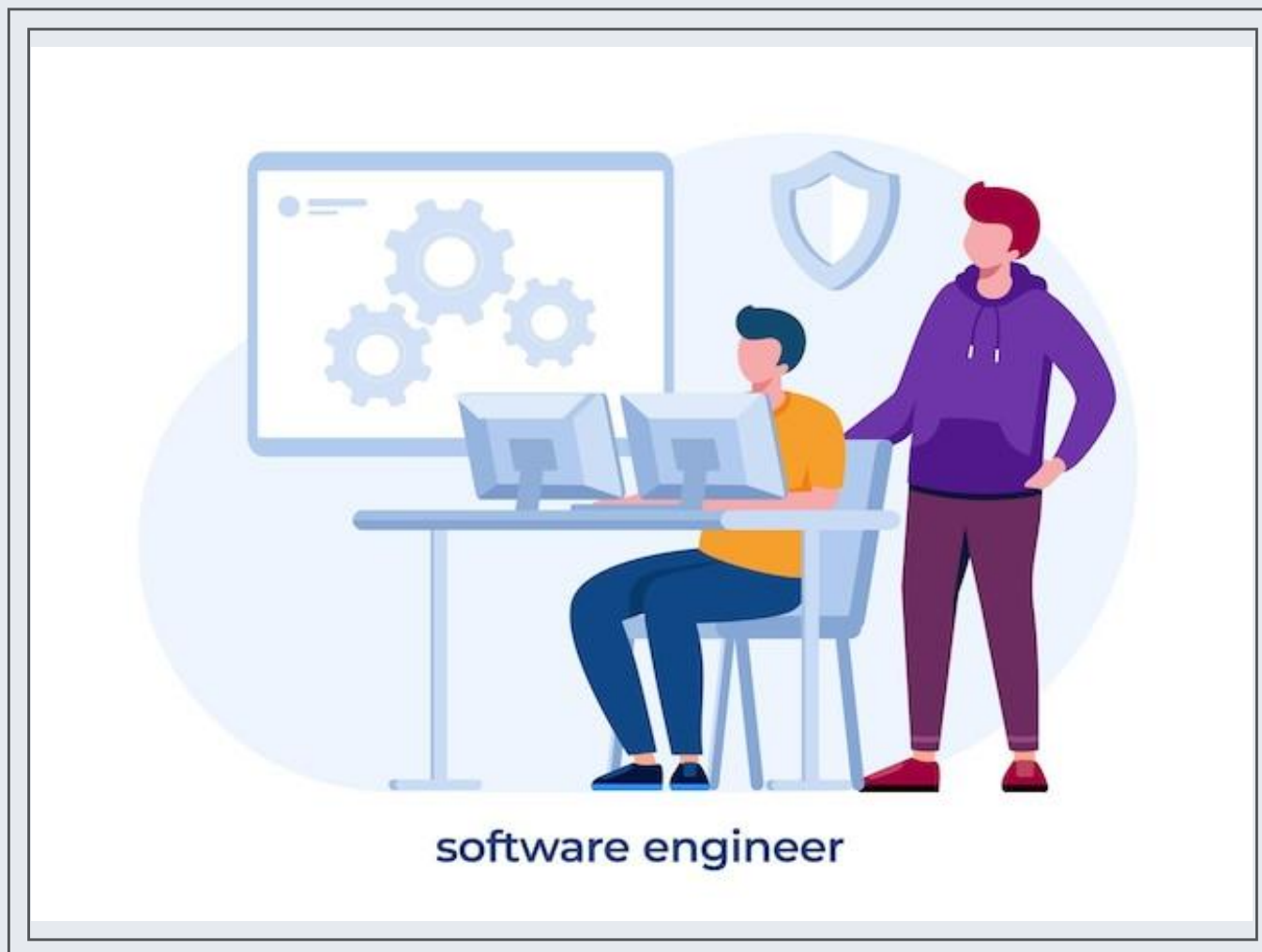
Introduction To


Software Engineering


Yohanes Serpiyanto Elo
IT Full Stack Developer
MSIB Batch 7

Universitas Bina Sarana Informatika | Teknologi Informasi | 2024

Pentingnya Software Engineering



 **Software Engineering** atau rekayasa perangkat lunak adalah disiplin ilmu yang berfokus pada penerapan prinsip-prinsip teknik dalam pengembangan perangkat lunak.

 Proses ini mencakup perancangan, pengembangan, pengujian, dan pemeliharaan perangkat lunak yang memenuhi spesifikasi pengguna dan standar kualitas tertentu.



Pentingnya Software Engineering

Software engineering diperlukan untuk menangani kompleksitas proyek pengembangan perangkat lunak modern yang terus berkembang. Dengan pendekatan yang terstruktur, tim pengembang dapat:

- Mengurangi risiko kesalahan atau kegagalan perangkat lunak.
- Meningkatkan efisiensi dan kualitas.
- Memastikan perangkat lunak dapat dipelihara dan diperbarui dengan mudah.



Software Development Life Cycle (SDLC)

SDLC adalah proses standar dalam pengembangan perangkat lunak yang terdiri dari beberapa tahapan penting:

★ Requirement Gathering & Analysis

Proses ini melibatkan pemahaman kebutuhan pengguna dan mendokumentasikannya sebagai spesifikasi teknis yang jelas.

★ System Design

Berdasarkan spesifikasi yang terkumpul, arsitektur sistem dan elemen-elemen penting dirancang. Desain ini mencakup pengaturan data, alur kerja, antarmuka pengguna, serta teknologi yang akan digunakan.

★ Implementation (Coding):

Dalam tahap ini, pengembang menulis kode berdasarkan desain yang telah dibuat. Teknologi yang digunakan tergantung pada bahasa pemrograman dan framework yang dipilih.

★ Testing

Perangkat lunak diuji untuk menemukan dan memperbaiki bug, serta memastikan bahwa perangkat lunak memenuhi spesifikasi awal. Metode testing bisa berupa unit testing, integration testing, hingga user acceptance testing.

★ Deployment

Setelah pengujian selesai, perangkat lunak dirilis ke lingkungan produksi, di mana pengguna akhir dapat menggunakannya.

★ Maintenance

Setelah perangkat lunak diluncurkan, perbaikan dan penambahan fitur tetap diperlukan untuk menjaga fungsionalitas dan relevansinya.



Full-Stack Development

Full-stack developer adalah pengembang yang mampu bekerja baik di sisi **frontend** (antarmuka pengguna) maupun **backend** (logika bisnis, database, dan API). Mereka memiliki keahlian di berbagai teknologi, antara lain:

- **Frontend:** HTML, CSS, JavaScript, dan framework modern seperti React.js, Angular, atau Vue.js.
- **Backend:** Bahasa pemrograman server seperti Node.js, Python (Django/Flask), Ruby on Rails, atau PHP. Mereka juga menggunakan database seperti MySQL, PostgreSQL, atau MongoDB.
- **API Development:** Membuat dan mengintegrasikan API (Application Programming Interface) untuk komunikasi antara frontend dan backend.
- **DevOps Knowledge:** Memahami infrastruktur dan penerapan otomatisasi deployment menggunakan alat seperti Docker, Kubernetes, dan cloud services (AWS, Azure).



Collaborative Development (Kolaborasi Pengembangan)

Dalam pengembangan perangkat lunak, kolaborasi menjadi aspek yang sangat penting, terutama dalam tim besar. Alat seperti **GitHub** dan **GitLab** menyediakan platform untuk kolaborasi melalui kontrol versi. Beberapa aktivitas penting dalam kolaborasi adalah:

- ① Menyinkronkan pekerjaan anggota tim melalui branch.
- ② Melakukan **Code Review** sebelum penggabungan untuk memastikan kualitas kode.
- ③ Memperbaiki konflik yang muncul saat melakukan merge.

Praktik kolaborasi ini meningkatkan kualitas dan stabilitas perangkat lunak, serta mengurangi risiko kesalahan yang mungkin terjadi akibat penggabungan kode dari banyak orang.

Design Thinking dalam Pengembangan Perangkat Lunak

Pendekatan **Design Thinking** digunakan untuk memecahkan masalah secara kreatif dengan fokus utama pada pengguna. Tahap-tahapnya adalah:

- ❖ **Empathize:** Memahami pengguna dan kebutuhannya dengan mendalam.
- ❖ **Define:** Mendefinisikan masalah secara jelas berdasarkan analisis dari tahap sebelumnya.
- ❖ **Ideate:** Menciptakan berbagai ide solusi yang inovatif.
- **Prototype:** Membuat prototipe sederhana dari solusi yang diusulkan untuk diuji.
- **Test:** Mencoba prototipe pada pengguna nyata dan mengumpulkan umpan balik untuk perbaikan.



Tools yang Sering Digunakan dalam Software Engineering

Berikut beberapa alat yang umum digunakan dalam pengembangan perangkat lunak:

- ❖ **Git:** Untuk version control dan kolaborasi kode.

- **Visual Studio Code:** Editor teks populer yang mendukung banyak bahasa pemrograman dan memiliki banyak ekstensi untuk meningkatkan produktivitas.

- ❖ **Browser:** Alat penting untuk menguji aplikasi web dan menganalisis performa frontend.

- **Docker:** Alat untuk membuat, mengirim, dan menjalankan aplikasi dalam kontainer untuk memastikan lingkungan pengembangan dan produksi konsisten.

- **CI/CD Tools:** Jenkins, CircleCI, dan TravisCI digunakan untuk otomatisasi build, testing, dan deployment.



Terima Kasih

Virtium Indonesia | Universitas Bina Sarana Informatika 2024