

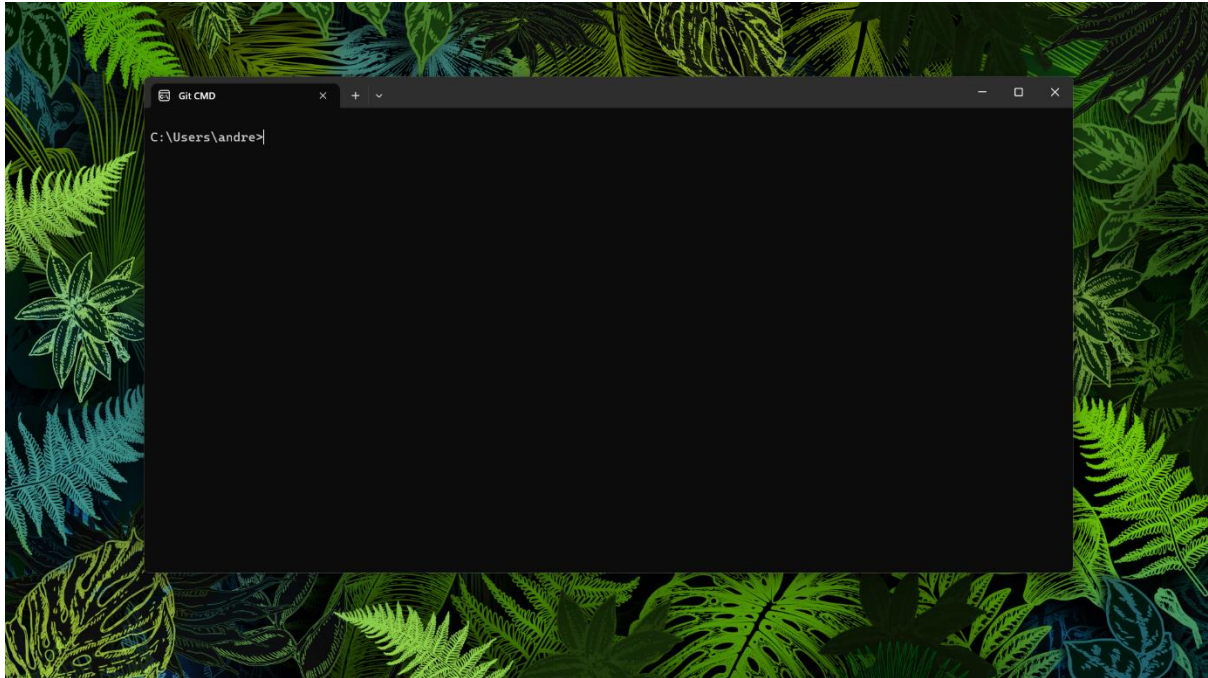
KELOMPOK 2

Nama : Oktavius Theo Andreas RIyadi

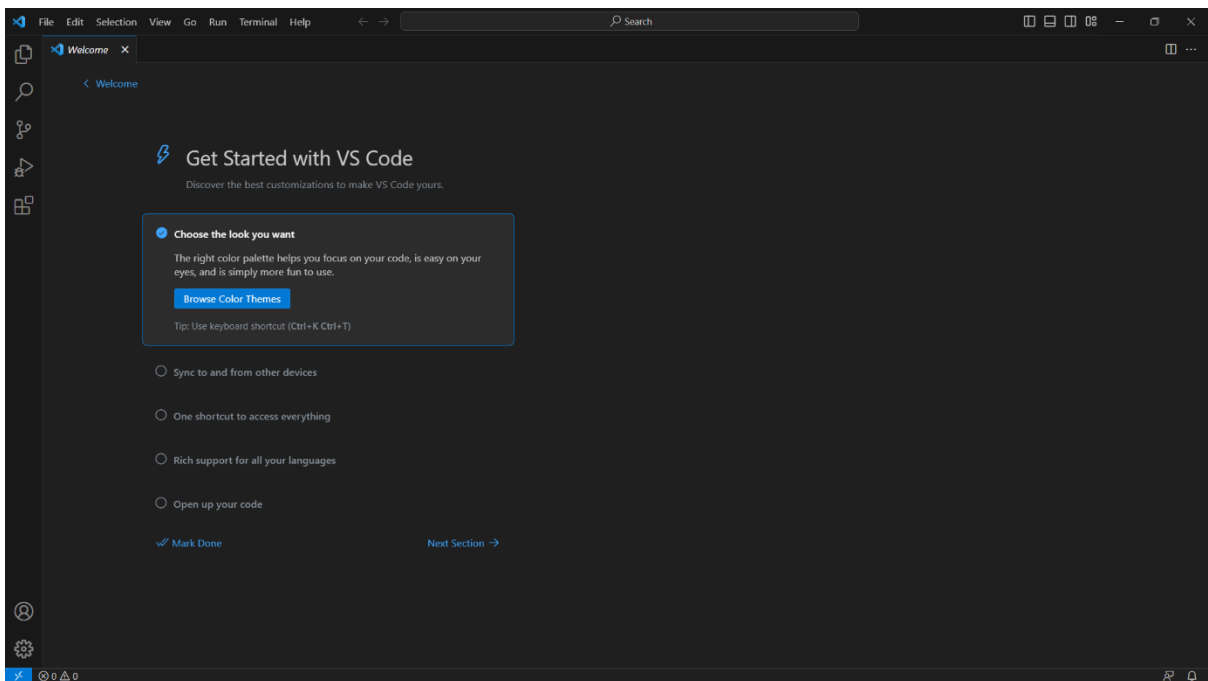
Email : andreas.or80@gmail.com

Soal 1

Screenshot ## Git



Screenshot ## Visual Studio Code



KELOMPOK 2

Nama : Oktavius Theo Andreas RIyadi

Email : andreas.or80@gmail.com

Create ## Summary

SDLC (Software Development Life Cycle)

SDLC (Software Development Life Cycle) adalah proses yang sistematis untuk mengembangkan dan memelihara perangkat lunak. SDLC terdiri dari beberapa fase, mulai dari perencanaan hingga pemeliharaan. Setiap fase memiliki tugas dan tanggung jawabnya masing-masing.

Fase-fase SDLC

- Perencanaan

Fase perencanaan adalah fase awal SDLC. Pada fase ini, tim pengembang akan menentukan tujuan dan persyaratan perangkat lunak. Tim pengembang juga akan mengumpulkan data dan informasi yang diperlukan untuk pengembangan perangkat lunak.

- Analisis

Fase analisis adalah fase lanjutan dari fase perencanaan. Pada fase ini, tim pengembang akan menganalisis persyaratan perangkat lunak. Tim pengembang akan melakukan analisis kebutuhan pengguna, analisis fungsionalitas perangkat lunak, dan analisis non-fungsionalitas perangkat lunak.

- Desain

Fase desain adalah fase di mana tim pengembang akan membuat desain perangkat lunak. Desain perangkat lunak terdiri dari desain arsitektur, desain antarmuka pengguna, dan desain database.

- Implementasi

KELOMPOK 2

Nama : Oktavius Theo Andreas RIyadi

Email : andreas.or80@gmail.com

Fase implementasi adalah fase di mana tim pengembang akan membangun perangkat lunak. Tim pengembang akan menggunakan berbagai bahasa pemrograman dan alat untuk membangun perangkat lunak.

- Pengujian

Fase pengujian adalah fase di mana tim pengembang akan menguji perangkat lunak untuk memastikan bahwa perangkat lunak memenuhi persyaratan. Pengujian perangkat lunak dapat dilakukan secara manual atau menggunakan alat pengujian otomatis.

- Pemeliharaan

Fase pemeliharaan adalah fase terakhir dari SDLC. Pada fase ini, tim pengembang akan melakukan pemeliharaan perangkat lunak. Pemeliharaan perangkat lunak dapat berupa perbaikan bug, penambahan fitur baru, atau modifikasi perangkat lunak.

SDLC untuk Pengembangan Web

SDLC juga dapat diterapkan untuk pengembangan web. Fase-fase SDLC untuk pengembangan web adalah sebagai berikut:

- Perencanaan

Pada fase perencanaan, tim pengembang akan menentukan tujuan dan persyaratan situs web. Tim pengembang juga akan mengumpulkan data dan informasi yang diperlukan untuk pengembangan situs web, seperti analisis kompetitor, analisis demografi pengguna, dan analisis tren industri.

- Analisis

Pada fase analisis, tim pengembang akan menganalisis persyaratan situs web. Tim pengembang akan melakukan analisis kebutuhan pengguna, analisis fungsionalitas situs web, dan analisis non-fungsionalitas situs web.

- Desain

Pada fase desain, tim pengembang akan membuat desain situs web. Desain situs web terdiri dari desain arsitektur, desain antarmuka pengguna, dan desain database.

- Implementasi

KELOMPOK 2

Nama : Oktavius Theo Andreas RIyadi

Email : andreas.or80@gmail.com

Pada fase implementasi, tim pengembang akan membangun situs web. Tim pengembang akan menggunakan berbagai bahasa pemrograman dan alat untuk membangun situs web.

- **Pengujian**

Pada fase pengujian, tim pengembang akan menguji situs web untuk memastikan bahwa situs web memenuhi persyaratan. Pengujian situs web dapat dilakukan secara manual atau menggunakan alat pengujian otomatis.

- **Pemeliharaan**

Pada fase pemeliharaan, tim pengembang akan melakukan pemeliharaan situs web.

Pemeliharaan situs web dapat berupa perbaikan bug, penambahan fitur baru, atau modifikasi situs web.

Git

Git adalah sistem kontrol versi yang populer digunakan untuk mengelola perubahan kode sumber. Git memungkinkan pengembang untuk melacak perubahan kode sumber, membuat versi kode sumber, dan bekerja sama dengan pengembang lain.

Fitur-fitur Git

- **Branching**

Branching adalah fitur Git yang memungkinkan pengembang untuk membuat cabang dari kode sumber utama. Cabang dapat digunakan untuk menguji perubahan kode sumber tanpa mempengaruhi kode sumber utama.

- **Merging**

Merging adalah fitur Git yang memungkinkan pengembang untuk menggabungkan perubahan kode sumber dari cabang ke kode sumber utama.

- **Pull Request**

Pull request adalah fitur Git yang memungkinkan pengembang untuk meminta pengembang lain untuk meninjau dan menyetujui perubahan kode sumber.

- **Commit**

KELOMPOK 2

Nama : Oktavius Theo Andreas RIyadi

Email : andreas.or80@gmail.com

Commit adalah tindakan menyimpan perubahan kode sumber ke repositori Git.

Cara Menggunakan Git

Untuk menggunakan Git, pengembang perlu menginstal Git di komputernya. Pengembang juga perlu membuat repositori Git untuk menyimpan kode sumbernya. Setelah repositori Git dibuat, pengembang dapat mulai menggunakan Git untuk mengelola perubahan kode sumbernya.

FSWDM (Front-End Software Development Methodology)

FSWDM (Front-End Software Development Methodology) adalah metodologi pengembangan perangkat lunak yang berfokus pada pengembangan antarmuka pengguna. FSWDM menggabungkan berbagai teknik dan pendekatan untuk memastikan bahwa antarmuka pengguna memenuhi persyaratan pengguna dan dapat digunakan dengan mudah.

Tujuan FSWDM

Tujuan FSWDM adalah untuk mengembangkan antarmuka pengguna yang:

- Mudah digunakan
- Efisien
- Efektivitas
- Estetis
- Fungsional
- Berorientasi pengguna

Tahap-tahap FSWDM

FSWDM terdiri dari beberapa tahap, yaitu:

- Planning

Tahap perencanaan adalah fase awal FSWDM. Pada tahap ini, tim pengembang akan menentukan tujuan dan persyaratan antarmuka pengguna. Tim pengembang juga akan mengumpulkan data dan informasi yang diperlukan untuk pengembangan antarmuka

KELOMPOK 2

Nama : Oktavius Theo Andreas RIyadi

Email : andreas.or80@gmail.com

pengguna, seperti analisis kebutuhan pengguna, analisis fungsionalitas antarmuka pengguna, dan analisis non-fungsionalitas antarmuka pengguna.

- **Design**

Tahap desain adalah fase di mana tim pengembang akan membuat desain antarmuka pengguna. Desain antarmuka pengguna terdiri dari desain arsitektur, desain visual, dan desain interaksi.

- **Development**

Tahap pengembangan adalah fase di mana tim pengembang akan membangun antarmuka pengguna. Tim pengembang akan menggunakan berbagai bahasa pemrograman dan alat untuk membangun antarmuka pengguna.

- **Testing**

Tahap pengujian adalah fase di mana tim pengembang akan menguji antarmuka pengguna untuk memastikan bahwa antarmuka pengguna memenuhi persyaratan. Pengujian antarmuka pengguna dapat dilakukan secara manual atau menggunakan alat pengujian otomatis.

- **Deployment**

Tahap deployment adalah fase di mana antarmuka pengguna akan dipublikasikan untuk digunakan oleh pengguna.

Front End Developer

Front end developer adalah pengembang yang bertanggung jawab untuk mengembangkan antarmuka pengguna dari sebuah aplikasi atau situs web. Front end developer menggunakan berbagai bahasa pemrograman dan alat untuk membangun antarmuka pengguna, seperti HTML, CSS, dan JavaScript.

Back End Developer

KELOMPOK 2

Nama : Oktavius Theo Andreas RIyadi

Email : andreas.or80@gmail.com

Back end developer adalah pengembang yang bertanggung jawab untuk mengembangkan backend dari sebuah aplikasi atau situs web. Backend developer menggunakan berbagai bahasa pemrograman dan alat untuk membangun backend, seperti Java, Python, dan PHP.

Mobile Developer

Mobile developer adalah pengembang yang bertanggung jawab untuk mengembangkan aplikasi mobile. Mobile developer menggunakan berbagai bahasa pemrograman dan alat untuk membangun aplikasi mobile, seperti Android, iOS, dan Flutter.

Collaborating Using Git

Git adalah sistem kontrol versi yang populer digunakan untuk mengelola perubahan kode sumber. Git memungkinkan pengembang untuk bekerja sama dengan pengembang lain secara efektif.

Cara Berkolaborasi Menggunakan Git

Untuk berkolaborasi menggunakan Git, pengembang perlu membuat repositori Git bersama. Setelah repositori Git dibuat, pengembang dapat mulai membuat cabang untuk mengerjakan perubahan kode sumber. Perubahan kode sumber dapat dibagikan ke pengembang lain dengan menggunakan pull request. Pengembang lain dapat meninjau dan menyetujui perubahan kode sumber sebelum digabungkan ke kode sumber utama.

Prinsip-prinsip Berkolaborasi Menggunakan Git

Berikut adalah beberapa prinsip yang perlu diperhatikan saat berkolaborasi menggunakan Git:

- Komit perubahan kode sumber secara teratur
- Gunakan cabang untuk mengerjakan perubahan kode sumber
- Gunakan pull request untuk membagikan perubahan kode sumber
- Tinjau dan setujui perubahan kode sumber sebelum digabungkan
- Komunikasikan dengan pengembang lain secara efektif

KELOMPOK 2

Nama : Oktavius Theo Andreas RIyadi

Email : andreas.or80@gmail.com

Kesimpulan

SDLC, Git, FSWDM, front end developer, back end developer, dan mobile developer adalah hal-hal penting yang perlu diketahui oleh pengembang perangkat lunak. Dengan memahami hal-hal tersebut, pengembang dapat mengembangkan perangkat lunak yang berkualitas tinggi dan dapat digunakan oleh pengguna dengan baik.