

SUMMARY MATERI WEEK 6

* **Full Stack Web Developer Career Path**

1. **Definisi dan Scope Pengembangan Full Stack**

Merujuk pada pengembangan seluruh aplikasi secara end-to-end, dari front-end, back-end, bahkan hingga client-side.

1. **Scope Penting Full Stack Development**

* *Front-End Development*: HTML, CSS, Javascript.

Framework🡪 React, Vue Js, AngularJs

* *Back-End Development*: Node Js, Python, Ruby, Java, PHP, C#.

Framework🡪 Express Js, Spring, Laravel, Rails

* *Database Management*: MySQL, PostgreSQL, MongoDB.

Framework🡪 *SQL*: Oracle, PostgreSQL, MySQL; *No SQL*: MongoDB, Redis

* *Integration of Front-End & Back-End*
* *Version Control & Collaboration*: GitHub, GitLab
* *Mobile Development*: React Native, Flutter

1. **Pengembangan Aplikasi End to End**

* Perencanaan dan Analisis
* Desain
* Pengembangan Front-End
* Pengembangan Back-End
* Integrasi dan Pengujian
* Pemeliharaan dan Peningkatan

1. **Manfaat Version Control Untuk Berkolaborasi**

* Rekam perubahan
* Pencatatan riwayat
* Pemecahan konflik
* Pemulihan mudah

1. **Tools Set**

* IDE – Code Editor: Visual Studio Code
* Version Control – Repository: GitHub, GitLab, Bitbucket
* Version Control – Git Tools: Sourcetree, GitLens
* DBMS: Oracle, MySQL, dll
* API: Postman, Swagger
* Test dan Debugging: Jest, Mocha Chai, JUnit
* Mobile Development: Flutter, dll
* Layanan Cloud: AWS, Google Cloud, Azure
* CI/CD: Jenkins, circleci
* Desain UI/UX: Figma, Sketch
* **SDLC & Design Thinking Implementation**

1. **Pengertian SDLC**

SDLC (Siklus Hidup Pengembangan Perangkat Lunak) adalah rangkaian proses yang terstruktur dan metodologi yang digunakan untuk mengembangkan perangkat lunak dari awal hingga selesai.

1. **Siklus SDLC**

* Perencanaan dan Analisis
* Desain
* Pengembangan
* Pengujian
* Penerapan
* Pemeliharaan

1. **Manfaat Penggunaan SDLC**

* Prediktabilitas dan Pengendalian Proyek
* Memenuhi Kebutuhan Pengguna
* Peningkatan Kualitas Perangkat Lunak
* Penghematan Biaya dan Waktu
* Pengelolaan Risiko yang Lebih Baik
* Meningkatkan Pengawasan dan Evaluasi
* Eﬁsiensi Tim dan Kolaborasi
* Peningkatan Dokumentasi

1. **Model-Model SDLC**

* *Waterfall Model*🡪 model yang linier dan berurutan, setiap tahap dalam model ini harus selesai sebelum memulai tahap berikutnya.
* *V-Shaped Model*🡪 terkait erat dengan model waterfall, tetapi menekankan pada pengujian setiap tahap pengembangan.
* *Prototype Model*🡪 bertujuan untuk menciptakan prototipe atau contoh awal sebelum mengembangkan versi ﬁnalnya.
* *Spiral Model*🡪 membangun pada inkrementasi sebelumnya, menghasilkan perangkat lunak yang semakin berkembang dengan ﬁtur yang lebih banyak setiap siklusnya.
* *Iterative Incremental Model*🡪 melibatkan pengulangan siklus pembangunan dan peningkatan perangkat lunak dalam tahapan-tahapan kecil.
* *Big Bang Model*🡪 kurang terstruktur, di mana semua tahapan pengembangan dilakukan tanpa perencanaan yang detail.
* *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.

1. **Steps Design Thinking**

* Empathize: Understand User Needs
* Deﬁne: Deﬁne the Problem
* Ideate: Generate Ideas
* Prototype: Build Quick and Iterative Solutions
* Test: Gather User Feedback
* Implement: Develop the Software
* **Basic Git & Collaborating Using Git**

1. **Sejarah Singkat Terminal**

Dengan perkembangan teknologi dan perangkat lunak, terminal tetap menjadi alat penting bagi para pengembang perangkat lunak, administrator sistem, dan pengguna teknis lainnya. Meskipun antarmuka grafis semakin canggih dan populer, terminal tetap memberikan fleksibilitas dan kekuatan untuk melakukan tugas-tugas khusus dan otomatisasi dalam lingkungan komputer modern.

1. **Command Line Dasar**

Bagian 1: Navigasi File dan Direktori

|  |  |  |  |
| --- | --- | --- | --- |
| **Command** | **Windows** | **Linux / macOS** | **Deskripsi** |
| List Files | dir | ls | Menampilkan daftar file dan direktori dalam folder saat ini |
| Change Directory | cd | cd | Berpindah direktori kerja saat ini |
| Make Directory | mkdir | mkdir | Membuat direktori baru |
| Remove Directory | rmdir | rm -r | Menghapus direktori |
| Delete Files | del | rm | Menghapus file |
| Copy Files | copy | cp | Menyalin file |
| Move / Rename | move | mv | Memindahkan atau mengganti nama file/direktori |

Bagian 2: Operasi Dasar Lainnya

|  |  |  |  |
| --- | --- | --- | --- |
| **Command** | **Windows** | **Linux / macOS** | **Deskripsi** |
| Display File | type | cat | Menampilkan isi file |
| Display Text | echo | echo | Menampilkan teks atau mengaktifkan/mematikan echoing command |
| List Processes | tasklist | ps | Menampilkan daftar proses yang sedang berjalan |
| Terminate Process | taskkill | kill | Menghentikan proses yang sedang berjalan |

Bagian 3: Perintah Tambahan untuk Administrasi dan Jaringan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | **Command** | **Windows** | **Linux / macOS** | **Deskripsi** | | Print Directory | N/A | pwd | Menampilkan direktori kerja saat ini | | Create File | N/A | touch | Membuat file kosong atau memperbarui timestamp | | Change Permissions | N/A | chmod | Mengubah izin akses file | | Change Ownership | N/A | chown | Mengubah kepemilikan file | | Search in Files | N/A | grep | Mencari pola dalam file | | Display Manual | N/A | man | Menampilkan halaman manual untuk suatu perintah | | Execute as Admin | N/A | sudo | Menjalankan perintah dengan hak administrator | | Ping | N/A | ping | Mengirim permintaan ICMP untuk mengecek konektivitas jaringan | | Network Config | ipconfig | ifconfig | Menampilkan konfigurasi antarmuka jaringan | | Secure Shell | N/A | ssh | Menghubungkan ke host jarak jauh menggunakan SSH | |

1. **Memahami Version Control Git**

* *Sistem Kontrol Versi Terpusat (Centralized Version Control System)*
* Menggunakan satu server pusat untuk menyimpan semua file dan riwayat versi dari proyek.
* Klien (pengguna) bekerja dengan salinan lokal dari file, tetapi harus terhubung ke server pusat untuk mengambil dan menyimpan perubahan.
* *Sistem Kontrol Versi Terdistribusi (Distributed Version Control System)*
* Setiap pengguna memiliki salinan lengkap dari seluruh repository di komputer lokal mereka, termasuk semua riwayat versi.
* Tidak memerlukan koneksi terus-menerus ke server pusat karena perubahan disimpan secara lokal dan dapat disinkronkan nanti.

1. **Pengertian Git**

Git adalah sistem kontrol versi terdistribusi yang memungkinkan pengembang perangkat lunak untuk melacak perubahan dalam kode mereka, berkolaborasi dengan anggota tim, dan mengelola revisi kode secara efektif.

1. **Dasar-Dasar Command Git**

|  |  |  |
| --- | --- | --- |
| **Command** | **Deskripsi** | **Contoh Penggunaan** |
| git init | Inisialisasi repository Git baru di direktori lokal. Membuat folder .git. | git init (di dalam folder proyek) |
| git clone | Menggandakan repository yang sudah ada dari server ke lokal. | git clone https://github.com/user/repo.git |
| git status | Menampilkan status perubahan file di working directory (modifikasi, staged, dll.) | git status |
| git add | Menambahkan perubahan file ke **staging area** untuk commit. | git add file.txt atau git add . (semua file) |
| git commit | Menyimpan perubahan yang ada di staging area ke repository dengan pesan commit. | git commit -m "Pesan commit" |
| git push | Mengirim commit yang ada di repository lokal ke remote repository (seperti GitHub). | |  |  |  | | --- | --- | --- | |  |  |  |  |  |  |  | | --- | --- | --- | |  |  | git push origin main | |
| git pull | Mengambil perubahan terbaru dari remote repository dan menggabungkannya dengan repository lokal. | git pull origin main |
| git branch | Melihat, membuat, atau menghapus cabang (branch) pada repository. | git branch (melihat), git branch nama-branch (membuat) |
| git checkout | Berpindah ke cabang lain atau ke commit tertentu. | git checkout nama-branch |
| git merge | Menggabungkan cabang lain ke cabang saat ini. | git merge nama-branch |
| git log | Menampilkan riwayat commit dalam repository. | git log |
| git remote | Mengelola daftar remote repository yang dikaitkan dengan repository lokal. | git remote -v (melihat remote), git remote add origin https://github.com/user/repo.git |
| git fetch | Mengambil perubahan dari remote repository tanpa langsung menggabungkannya. | git fetch origin |
| git diff | Menampilkan perbedaan antara file yang ada di working directory dan repository. | git diff atau git diff file.txt |
| git reset | Mengembalikan perubahan baik di staging area atau commit ke status sebelumnya. | git reset --hard HEAD (ke commit sebelumnya) |

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |