

**Nama** : I Ngurah Komang Agus  
**Suryadiyatmika. S**  
**Class** : IT Fullstack Developer  
**Batch** : SIB 7

### Soal

1. Silahkan teman teman melakukan proses instalasi beberapa tools dibawah ini:

- Git
- Visual Studio Code
- Browser

Setelah melakukan instalasi, buat sebuah summary untuk apa yang sudah dipelajari dalam materi pengenalan software engineer sebelumnya pada google docs.

2. Buatlah repositori GitHub baru dengan nama "finpro-msib-7-kelompok X"

Contoh: finpro msib 7 kelompok 1

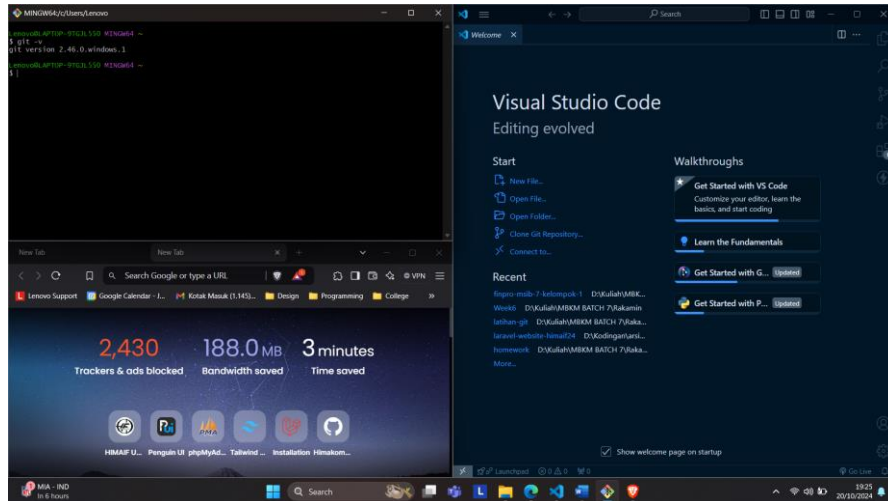
*\*1 kelompok hanya membuat 1 repositori saja*

- Undang teman-teman kelompokmu sebagai kolaborator pada repositori tersebut
- Setiap peserta meng-clone repositori ke mesin lokal masing-masing
- Setiap peserta harus membuat cabang untuk pekerjaannya. Misal: doni-homework, nita-homework, etc
- Masukkan Homework dari minggu sebelumnya ke branch lokalmu.
- Setelah melakukan perubahan, teman-teman harus melakukan commit dengan pesan commit yang deskriptif
- Lakukan push and pull ke branch homework dari branch lokalmu.
- Pastikan konflik sudah terselesaikan sebelum melakukan pull request

Setelah selesai, bisa teman teman zip hasil soal pertama dan kedua, lalu upload ke dalam LMS Rakamin.

## JAWABAN

1. Tools yang telah diinstall yaitu git dengan version 2.46.0, lalu untuk text editor yaitu Visual Studio Code version 1.94.2.0 dan untuk browser yang digunakan yaitu Brave.



### **Summary “Introduction to Software Engineering**

#### **a. Full Stack Developer Career Path**

Pengembangan full stack development memiliki arti pada pengembangan seluruh aplikasi secara **end to end**, mulai dari sisi depan (*front end*) dan sisi belakang (*back end*) dan bahkan hingga ke sisi client (*client-side*). Dalam pengembangan full stack, penting adanya penguasaan terkait dengan

##### **2. Front End Development**

Pengembangan pada sisi klien dengan dasar pengembangan meliputi HTML, CSS dan Javascript (Popular framework yang digunakan ada React JS, Vue JS, dan Angular JS)

##### **3. Back End Development**

Bertanggung jawab dalam proses permintaan dari pengguna, mengelola, dan menyimpan data di database, serta memberikan response berdasarkan request. Adapun dasar pengembangan Back End meliputi Bahasa pemrograman server-side, server framework dan database management.

##### **4. Database Management**

Bertanggung jawab dalam mengelola data dalam sebuah aplikasi atau sistem, sehingga lebih mudah diorganisir ketika data digunakan. Adapun dasar pengembangan meliputi database management system, tipe database, dan bahasa query.

##### **5. Integration of Front End dan Back End**

## 6. Version Control and Collaboration

## 7. Mobile Development

Konsep yang digunakan untuk membangun aplikasi yang dapat dijalankan di perangkat mobile, seperti smartphone dan tablet. Dasar pengembangan mobile development meliputi platform mobile (android, iOS, dan Windows Phone) dan IDE (integrated Development Environment).

Tahap-tahap dalam pengembangan aplikasi end-to-end

1. Perencanaan dan Analisis, merujuk pada analisis kebutuhan dan riset pasar
2. Desain, merujuk pada perancangan antarmuka dan pengalaman pengguna
3. Pengembangan Front End
4. Pengembangan Back End
5. Integrasi dan Pengujian
6. Pemeliharaan dan Peningkatan

Kolaborasi Efektif, dapat dilakukan dengan penggunaan version control atau pengendalian versi untuk melacak perubahan kode suatu aplikasi. Adapun salah satu contoh aplikasi yang dapat digunakan sebagai version control yaitu Git. Manfaat dari adanya version control ini yaitu:

- |                       |                      |
|-----------------------|----------------------|
| 1. Rekam perubahan    | 3. Pemecahan Konflik |
| 2. Pencatatan Riwayat | 4. Pemulihan Mudah   |

Tools yang dapat digunakan sebagai fullstack developer yaitu :

1. IDE – Code Editor (disarankan yaitu Visual Studio Code)
2. Version Control – Repository (Github, Gitlab, Bitbucket)
3. Version Control – Git Tools (Sourcetree, gitlens)
4. DBMS, database management system (PostgreSQL, MySQL, MongoDB, Oracle)
5. API (Postman, Swagger)
6. Testing dan Debugging (Jest, Mocha, chai, Junit5)
7. Mobile Development (react native, flutter)
8. Layanan cloud (AWS, Google Cloud, Azure)
9. CI/CD (Jenkins, circleci)
10. Desain UI/UX (Figma, Sketch)

## **b. SDLC & Design Thinking Implementation**

SDLC atau System Development Life Cycle atau Siklus Hidup Pengembangan Perangkat Lunak merupakan suatu rangkaian proses yang terstruktur dan metodologi dalam mengembangkan perangkat lunak dari awal hingga selesai.

Tahapan tersebut diantaranya:

1. Perencanaan dan Analisis  
Meliputi identifikasi masalah atau kebutuhan bisnis yang perlu diselesaikan.
2. Desain  
Mencakup perancangan arsitektur sistem, antarmuka pengguna dan desain database.
3. Pengembangan Produk  
Implementasi desain ke dalam bentuk kode untuk menghasilkan produk
4. Pengujian  
Mencakup tahapan untuk memverifikasi fungsionalitas, kinerja, keamanan, dan kualitas keseluruhan perangkat lunak.
5. Penerapan  
Implementasi desain ke dalam bentuk kode untuk menghasilkan produk
6. Pemeliharaan  
Memperbaiki bug, peningkatan fitur dan menjaga perangkat lunak agar sesuai dengan perubahan kebutuhan bisnis

Manfaat penggunaan SDLC:

1. Predikabilitas dan pengendalian proyek
2. Peningkatan kualitas perangkat lunak
3. Pengelolaan risiko yang lebih baik
4. Efisiensi Tim dan Kolaborasi
5. Memenuhi kebutuhan pengguna
6. Penghematan biaya dan waktu
7. Meningkatkan pengawasan dan Evaluasi
8. Peningkatan dokumentasi

Model-model SDLC:

1. Waterfall : Model SDLC dimana satu tahap harus selesai sebelum tahap berikutnya

2. V-Shaped : Model waterfall namun menekankan pada pengujian.
3. Prototype : Model pengembangan yang menciptakan prototype terlebih dahulu sebelum mengembangkan versi final
4. Spiral : Model elemen spiral dengan pendekatan incremental yaitu menghasilkan perangkat lunak yang semakin berkembang di setiap siklus
5. Iterative Incremental : Model yang membagi pembangunan dan peningkatan perangkat lunak ke dalam tahapan-tahapan kecil
6. Big Bang : Model yang kurang terstruktur, tahap pengembangan tanpa perencanaan detail
7. Agile : Model pendekatan kolaboratif dan iterative dengan fokus pengiriman perangkat lunak secara berkala dan inkremental

Tahapan dalam desain thinking

1. Empathize : Memahami kebutuhan pengguna
2. Define : Mendefinisikan permasalahan
3. Ideate : Merancang ide
4. Prototype : Merancang solusi cepat dan iterative
5. Test : Mengumpulkan feedback pengguna
6. Implement : Mengembangkan ke dalam bentuk perangkat lunak

### c. Basic Git & Collaborating Using Git

Terminal digunakan untuk memberikan fleksibilitas dan kekuatan untuk melakukan tugas-tugas khusus dan otomatisasi dalam lingkungan komputer modern. Contoh beberapa command line dasar:

Windows	Linux/macOS	Fungsi
'dir'	'ls'	List file yang ada pada direktori saat ini
'cd'	'cd'	Pindah direktori saat ini ke direktori lain
'mkdir'	'mkdir'	Membuat direktori baru

Version control seperti yang telah dijelaskan pada materi poin a yaitu mengenai **Full Stack Developer Career Path**, dijelaskan bahwa version control digunakan untuk

melacak setiap perubahan kode yang ada dalam sebuah proyek. Sistem control terbagi menjadi dua jenis, yaitu:

1. Sistem Kontrol Versi Terpusat, menyimpan seluruh proyek pada 1 repo (sebagai master)
2. Sistem Kontrol Versi Terdistribusi, setiap anggota menyimpan Salinan lengkap dari seluruh repo

Dasar-dasar command pada git

1. git init : inisialisasi direktori sebagai repository
2. git clone : duplikat repository
3. git status : cek status perubahan
4. git add : menambahkan file ke staging area
5. git commit : membuat commit dan menambahkan pesan sudah di-staging
6. git push : upload ke remote repository
7. git pull : mengambil repo remote ke versi local
8. git branch : menampilkan daftar cabang
9. git checkout: beralih ke cabang lain
10. git merge : menggabungkan cabang aktif
11. git log : menampilkan daftar Riwayat commit
12. git remote : menampilkan daftar remote repo yang terhubung ke repo local
13. git fetch : mengambil informasi terbaru dari remote tanpa perlu menggabungkan perubahan
14. git diff : menampilkan perbedaan file antara versi yang sudah di-staging
15. git reset : mengembalikan file yang sudah di-staging

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