PRAKTIKUM 1

SETUP ENVIRONMENT

# Tujuan Praktikum

* Mahasiswa mengetahui *tools* yang akan digunakan selama perkuliahan
* Mahasiswa mengetahui cara membuat akun dan repositori di Github
* Mahasiswa mengetahui cara melakukan *version control* dengan menggunakan Github

# Dasar Teori

Praktikum ini mencakup beberapa dasar teori yang melibatkan alat-alat kunci dalam pengembangan web modern. Pertama, XAMPP atau Laragon digunakan sebagai platform pengembangan lokal, menyediakan server Apache, MySQL, dan PHP untuk memungkinkan mahasiswa mengembangkan serta menguji aplikasi web secara efisien sebelum diterapkan di lingkungan produksi. Selanjutnya, NPM (Node Package Manager) adalah alat manajemen paket untuk JavaScript yang memfasilitasi pengelolaan dependensi proyek dan eksekusi skrip. Composer, sebagai manajer dependensi untuk PHP, memainkan peran penting dalam mengelola pustaka dan dependensi proyek PHP dengan mudah.

Visual Studio Code, sebagai editor kode sumber, membantu mahasiswa dalam mengembangkan aplikasi web dengan menyediakan fitur-fitur seperti penyusunan kode, debugging, dan integrasi dengan ekstensi untuk berbagai bahasa dan platform. Terakhir, Git dan GitHub digunakan untuk mengelola versi kode dan kolaborasi tim. Mahasiswa perlu memahami konsep dasar Git, termasuk commit, branch, pull request, dan merge, untuk efektif berkolaborasi dalam pengembangan proyek. Pemahaman mendalam tentang setiap alat ini menjadi kunci bagi mahasiswa untuk berhasil mengintegrasikan dan mengoptimalkan penggunaannya dalam pengembangan web dan kerja kolaboratif.

# Persiapan Alat dan Bahan

* Seperangkat PC dan Koneksi Internet
* Web Browser, contoh : Microsoft Edge, Google Chrome
* Aplikasi Web pada Semester 1

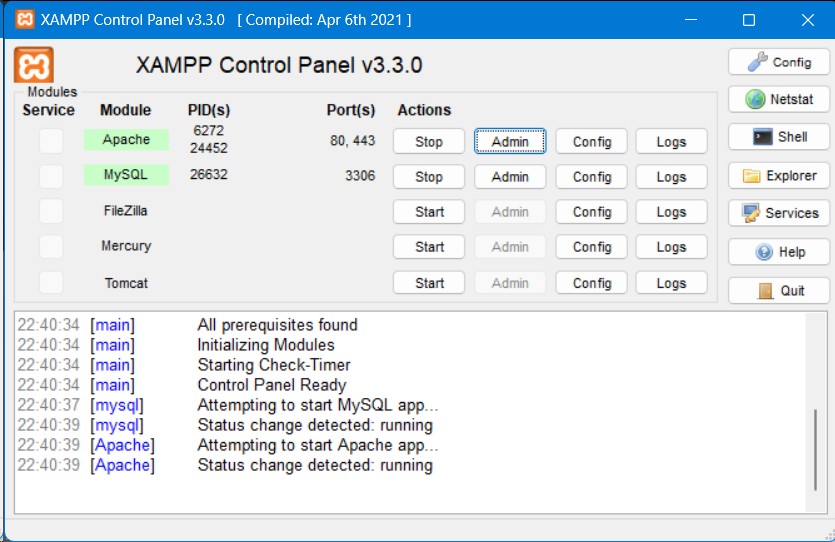
# Kegiatan Praktikum

Dalam Praktikum ini, fokus utamanya adalah pada instalasi alat-alat yang dibutuhkan dan pembuatan akun GitHub. Kegiatan dimulai dengan langkah-langkah untuk menginstal tools seperti XAMPP/Laragon, NPM, Composer, Visual Studio Code, dan Git. Mahasiswa akan diberikan panduan langkah-demi-langkah untuk memastikan bahwa setiap alat terkonfigurasi dengan benar di lingkungan kerja mereka. Selanjutnya, kegiatan praktikum melibatkan proses pembuatan akun GitHub, di mana mahasiswa akan diajak untuk membuat akun, mengonfigurasi profil, dan memahami dasar-dasar penggunaan platform GitHub.

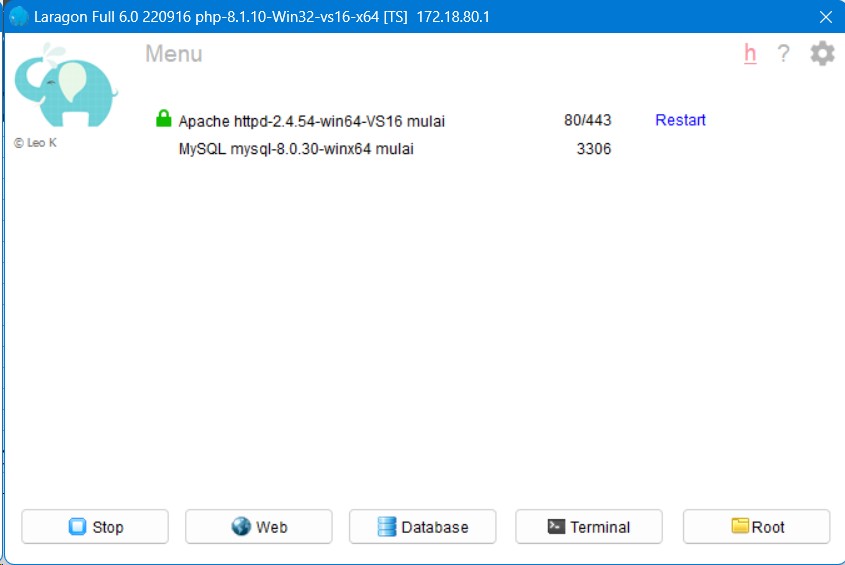
Setelah instalasi dan pembuatan akun selesai, fokus selanjutnya adalah memperkenalkan GitHub sebagai sistem kontrol versi (version control). Mahasiswa akan diajak untuk membuat repositori di GitHub, melakukan commit, membuat branch, dan menggabungkan perubahan melalui pull request. Kegiatan ini dirancang untuk memberikan pemahaman praktis tentang bagaimana menggunakan GitHub sebagai alat kolaborasi dan version control dalam pengembangan perangkat lunak. Dengan demikian, mahasiswa akan dapat merasakan langsung bagaimana alat-alat tersebut berinteraksi dan mendukung proses pengembangan aplikasi web.

## Persiapan Praktikum

1. Periksa instalasi Local Development Server (XAMPP) Anda, Pastikan aplikasi dapat berjalan dengan baik ! Jika Anda tidak menggunakan XAMPP, sebutkan Local Development Server yang Anda gunakan ! Selain itu, periksa juga versi PHP dari XAMPP Anda, Pastikan berada di PHP versi 8 !



*Gambar 1 Aplikasi XAMPP*



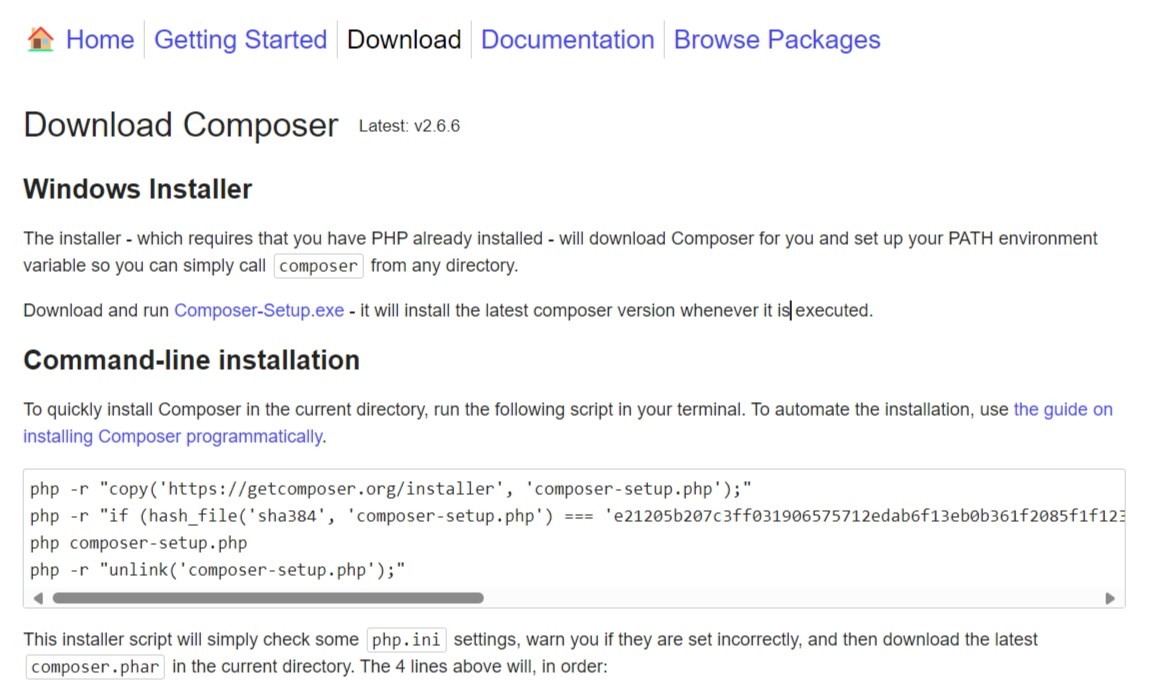
*Gambar 2 Aplikasi Laragon*

Periksa versi PHP Anda ! Bagaimana cara melakukannya ?

|  |
| --- |
| Menjalankan perintah php　-v pada cmd maka hasilnya seperti ini |

## Instalasi Composer

1. Unduh Composer pada situs berikut [Unduh Composer](https://getcomposer.org/download/) ! Unduhlah sesuai dengan sistem operasi Anda !

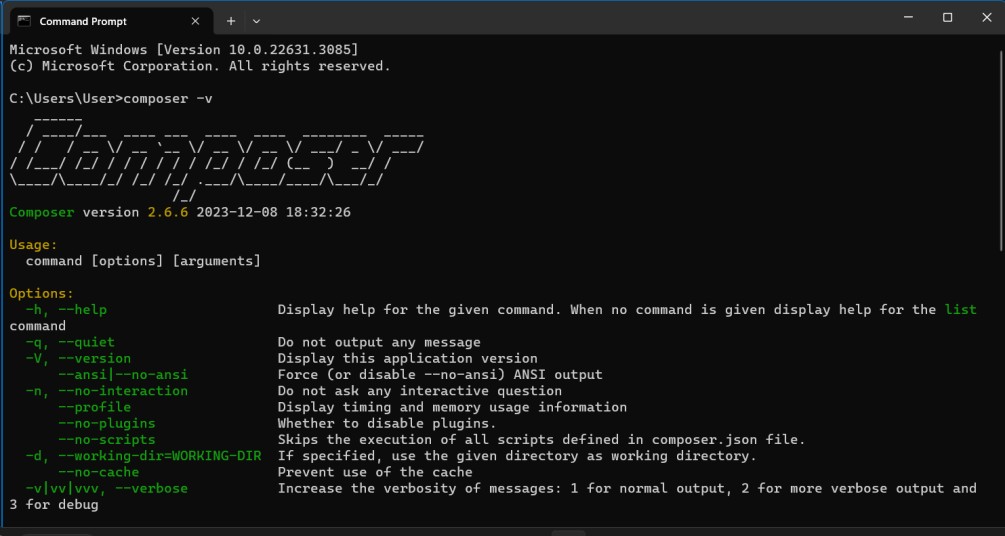


*Gambar 3 Unduh Composer*

1. Jelaskan proses instalasi composer nya !

|  |
| --- |
| Buka halaman resmi Composer.  Klik Composer-Setup.exe untuk mengunduh installer.  Buka file Composer-Setup.exe yang telah diunduh.  Ikuti langkah-langkah pada wizard instalasi:  Pilih lokasi PHP.exe (biasanya di C:\xampp\php\php.exe jika menggunakan XAMPP).  Klik Next sampai selesai. |

1. Pastikan composer sudah terinstal, salah satu cara untuk mengecek apakah composer sudah terinstal adalah dengan mengetik perintah **composer -v** pada terminal CMD !



*Gambar 4 Cek Instalasi Composer*

## Instalasi NPM

1. Unduh Node.js pada situs berikut [Unduh Node.js](https://nodejs.org/en) ! Unduhlah sesuai dengan sistem operasi Anda !



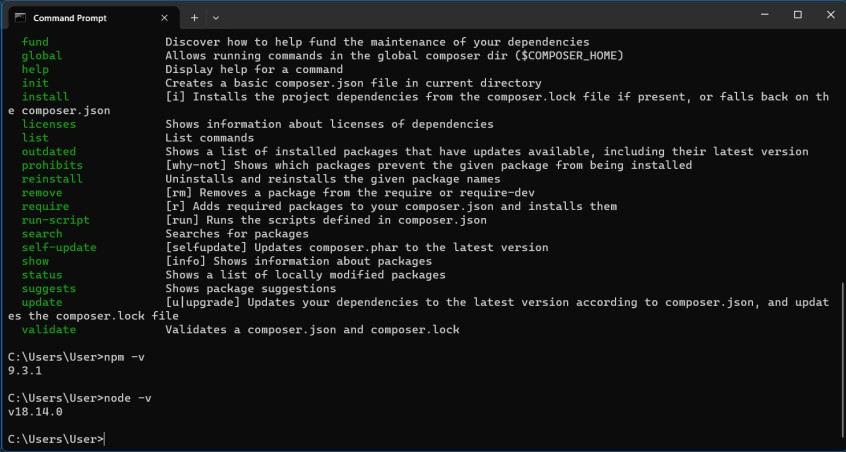
*Gambar 5 Unduh Node.js*

Disarankan untuk mengunduh versi **20.11.0 LTS**

1. Jelaskan proses instalasi Node.js !

|  |
| --- |
| Buka situs resmi Node.js: https://nodejs.org/.  Pilih versi LTS (Long-Term Support) untuk stabilitas atau versi terbaru untuk fitur terbaru.  Unduh file Windows Installer (.msi).  Buka file .msi yang telah diunduh.  Ikuti wizard instalasi:  Pilih Next sampai selesai.  Centang opsi Automatically install necessary tools jika ingin menggunakan fitur tambahan seperti npm. |

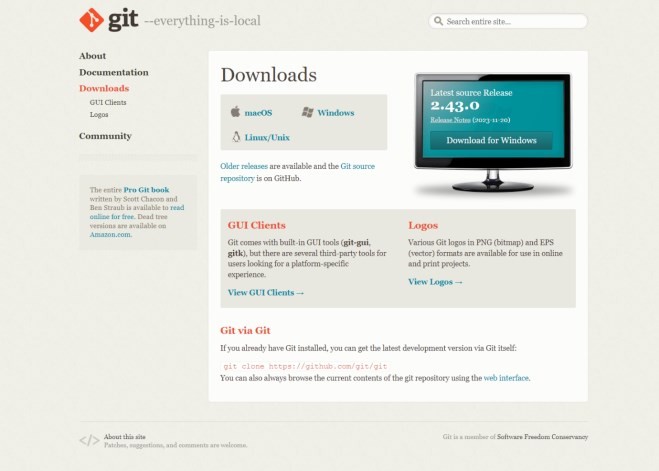
1. Pastikan node.js dan npm sudah terinstal, salah satu cara untuk mengecek instalasi nya adalah dengan mengetik perintah **node -v** dan **npm -v** pada terminal CMD !



*Gambar 6 Cek Instalasi Node.js*

## Instalasi Git

1. Unduh Git pada situs berikut [Unduh Git](https://git-scm.com/download) ! Unduhlah sesuai dengan sistem operasi Anda !

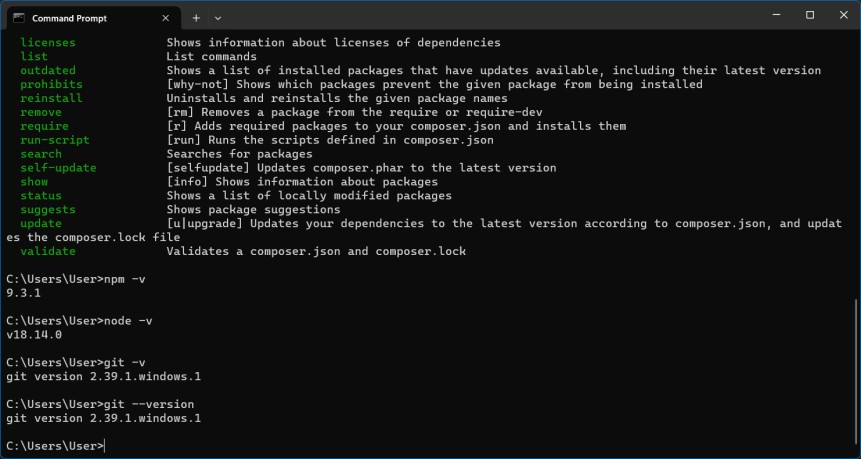


*Gambar 7 Unduh Git*

1. Jelaskan proses instalasi Git nya !

|  |
| --- |
| Unduh Installer Git  Kunjungi situs resmi: https://git-scm.com/downloads.  Pilih Windows dan unduh file .exe.  Pilih lokasi instalasi (biarkan default jika tidak yakin).  Pada opsi Adjusting your PATH environment, pilih Git from the command line and also from 3rd-party software (disarankan).  Pilih editor teks default (misalnya VS Code atau Nano).  Pilih opsi default lainnya kecuali ada preferensi khusus.  Klik Install dan tunggu hingga proses selesai. |

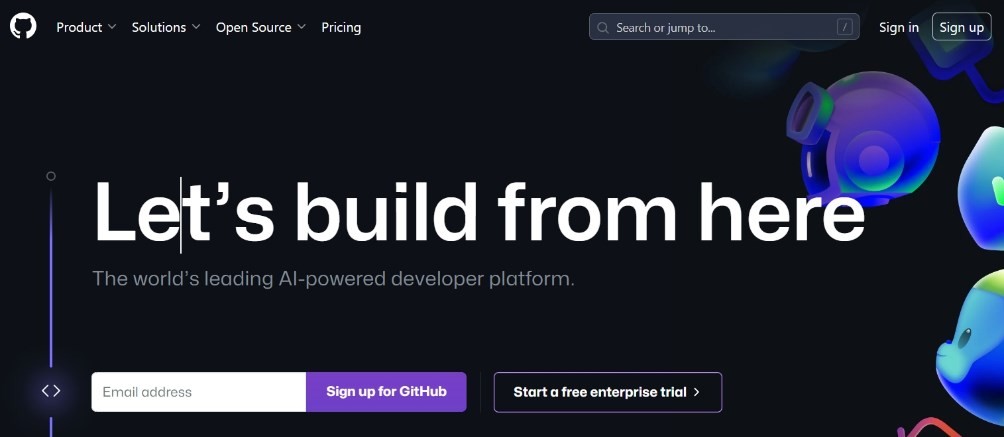
1. Pastikan Git sudah terinstal, salah satu cara untuk mengecek instalasi nya adalah dengan mengetik perintah **git -v** atau **git –version** pada terminal CMD !



*Gambar 8 Cek Instalasi Git*

## Membuat Akun Github

1. Kunjungi situs [Github!](https://github.com/) Lalu buatlah akun baru jika belum mempunyai akun !



*Gambar 9 Situs Github*

1. **Apabila Anda belum memiliki akun**, jelaskan proses pembuatan akunnya ! **Apabila Anda sudah memiliki Akun,** berikan tangkapan layar halaman dashboard Anda !

|  |
| --- |
|  |

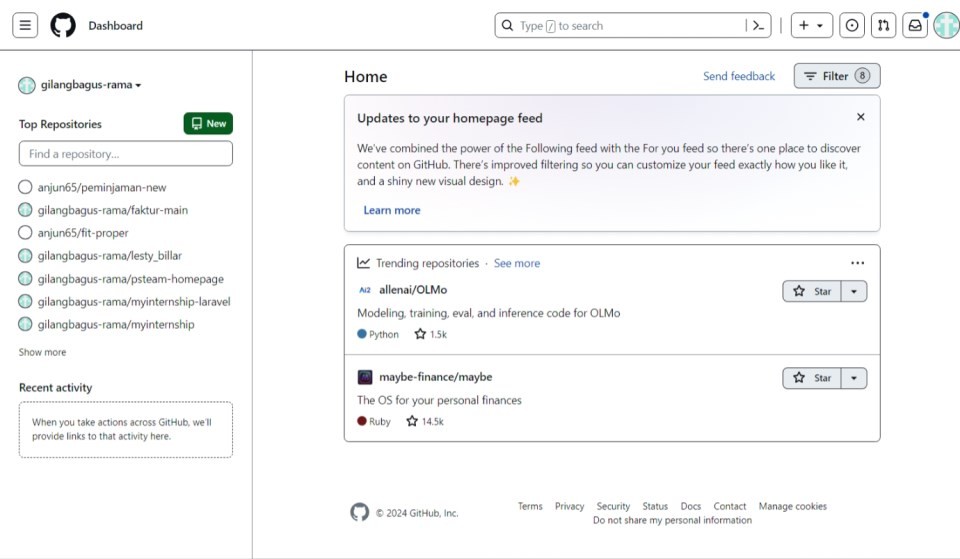
1. Anda sekarang (atau sudah memiliki) akun Github ! Berikan Username serta E-Mail yang terdaftar di akun Github Anda !

E-Mail Github : Berkatsiallagan201105@gmail.com

Username Github : TnKatt

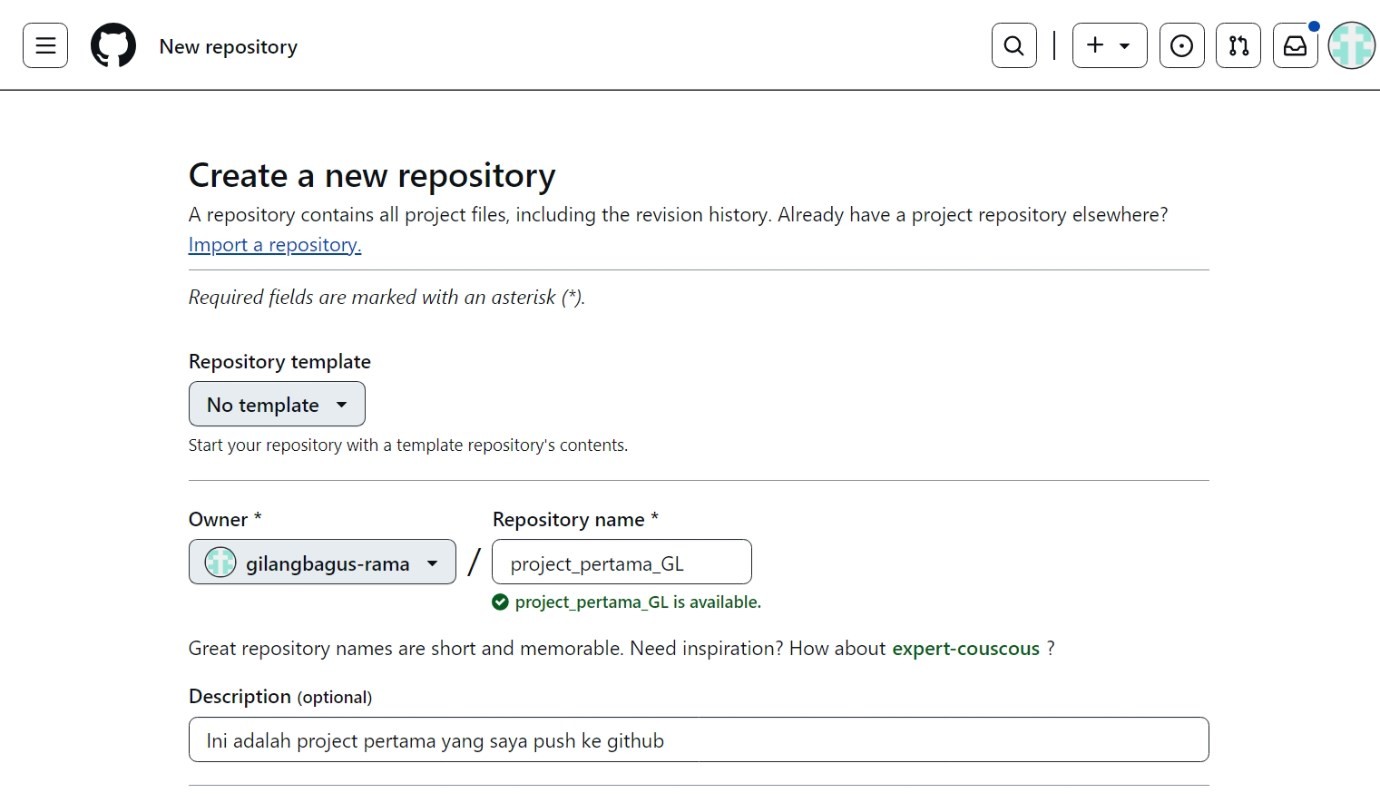
## Membuat Repositori

1. Login ke akun Github Anda !

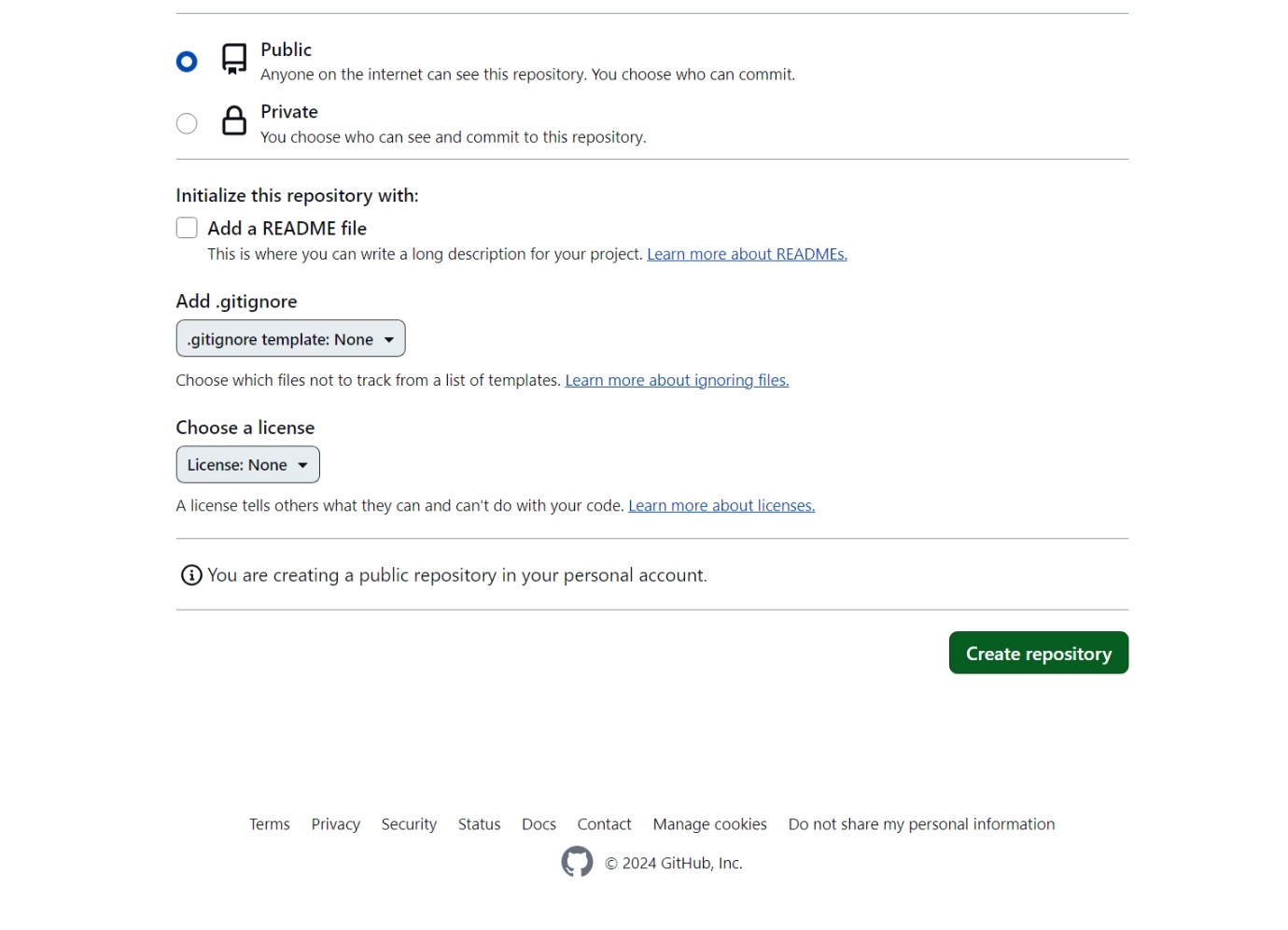


*Gambar 10 Halaman Dashboard User Github*

1. Pada Sidebar, bagian Repoisitori, klik New. Lalu isi dengan data yang Anda inginkan !

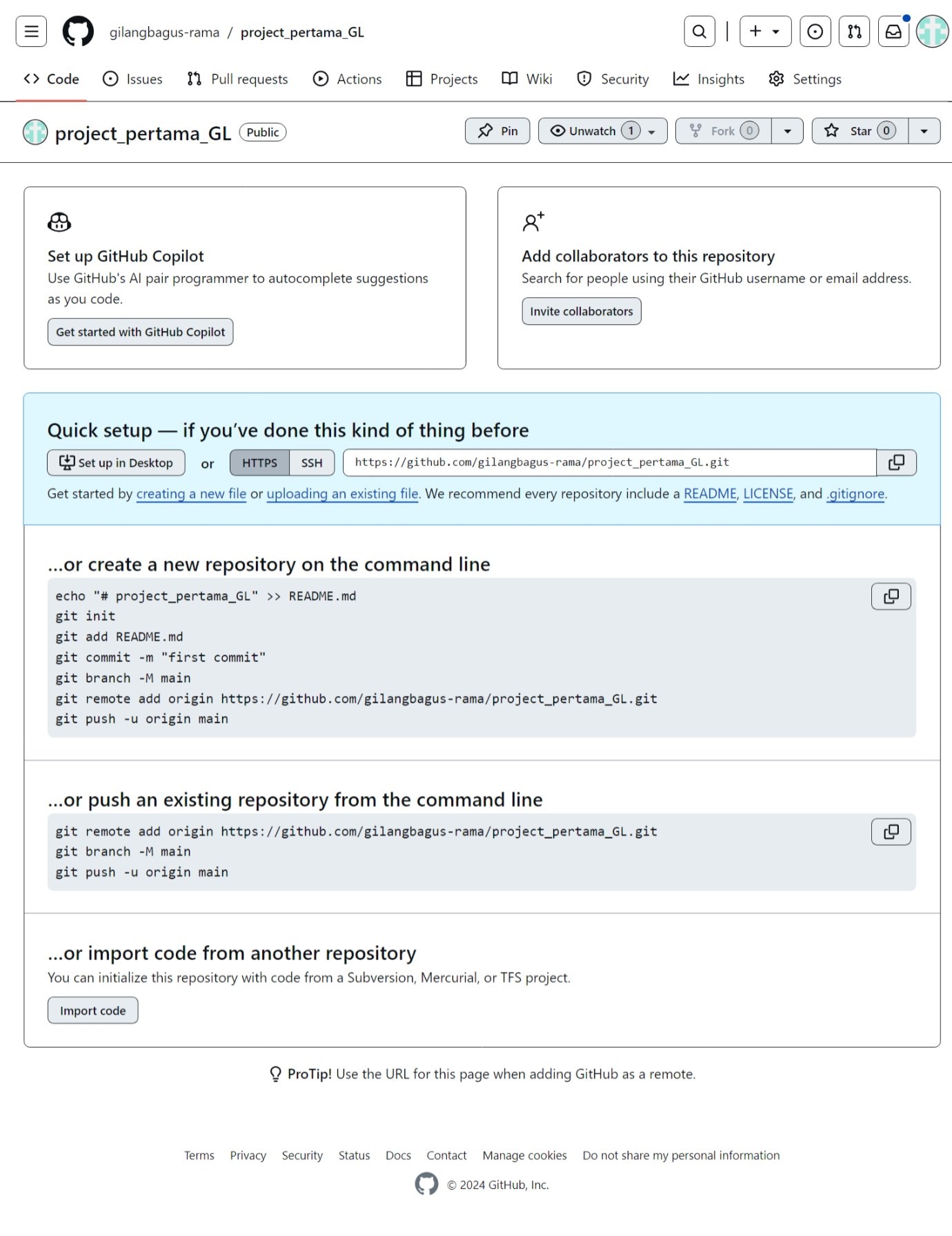


*Gambar 11 Membuat Repositori Baru (1)*



*Gambar 12 Membuat Repositori Baru (2)*

1. Repositori baru berhasil dibuat !



*Gambar 13 Repositori Baru berhasil dibuat*

1. ApakahAnda berhasil membuat repositori baru ? Jika tidak jelaskan kendala yang Anda alami, Jika berhasil berikan Tangkapan Layar Repositori Anda!

|  |
| --- |
|  |

Sekarang Anda sudah mengerti cara instalasi tools yang akan digunakan selama kegiatan perkuliahan Pemrograman Berbasis Objek. Anda juga sudah memiliki Akun Github serta berhasil membuat Repositori pertama Anda !

## Tugas Tambahan

1. Anda sudah banyak membuat project web selama semester 1 yang lalu. Pilih salah satu project web Anda, kemudian *push* project tersebut ke Repositori Github yang sudah Anda buat ! Jelaskan secara singkat Langkah - Langkah nya !

|  |
| --- |
| 1. Saya menyiapkan project web yang sudah saya buat sebelumnya dan memastikan semua file tersimpan dalam satu folder, misalnya dengan nama "my-web-project".  2. Saya membuka GitHub di [https://github.com](https://github.com) dan login ke akun saya.  - Saya klik tombol "New Repository" (+ New).  - Saya memberi nama repository, misalnya "my-web-project".  - Saya memilih opsi "Public" agar bisa diakses oleh siapa saja atau "Private" jika ingin menyimpannya secara pribadi.  - Saya tidak mencentang "Initialize this repository with a README" agar bisa langsung menghubungkannya dengan project saya.  - Saya klik "Create Repository".  3. Saya membuka Terminal (Linux/Mac) atau Command Prompt (Windows), lalu masuk ke folder proyek saya dengan perintah:  cd path/to/my-web-project  4. Saya menginisialisasi Git di dalam folder proyek saya dengan menjalankan perintah:  git init  Ini membuat folder ".git" yang menandakan bahwa project saya sekarang dikelola dengan Git.  5. Saya menghubungkan project saya ke repository GitHub yang sudah saya buat sebelumnya dengan perintah:  git remote add origin https://github.com/username/my-web-project.git  (saya mengganti "username" dengan username GitHub saya).  6. Saya menambahkan semua file project ke dalam Git dan membuat commit pertama:  git add .  git commit -m "First commit - Upload project web"  7. Saya mengirim (push) project saya ke GitHub dengan perintah:  git branch -M main  git push -u origin main  Karena ini pertama kali saya menggunakan GitHub di perangkat ini, saya perlu login atau menggunakan Personal Access Token (PAT) sebagai pengganti password GitHub.  8. Setelah proses push selesai, saya membuka repository saya di GitHub dan melakukan refresh halaman. Saya melihat semua file project saya sudah berhasil terunggah ke repository GitHub.  Sekarang project web saya sudah tersimpan di GitHub dan bisa saya akses kapan saja! |

Halaman

Semoga dengan kesungguhan dan usaha yang Anda lakukan dalam mengerjakan soal ini, akan membawa hasil yang memuaskan.

Ingatlah untuk memperhatikan setiap detail dan menjawab dengan teliti. Jangan lupa untuk menikmati proses belajar dan mengerjakan soal ini. Semoga berhasil dan sukses selalu!

Halaman