



MJ CREATIVE

**VueJs,  
NodeJs,  
Expressjs**

PEMROGRAMAN WEB

# **MODUL PRAKTIKUM**

Mujiyanto, M.KOM

Be a Web Developer Like a Boss

## Daftar Isi

1. ....	0
<i>Daftar Isi</i> .....	1
<b>BAB 1 . Mengatur Proyek di GitHub</b> .....	2
<b>A. Pendahuluan</b> .....	2
<b>B. Persiapan</b> .....	2
C. Langkah 1: Membuat Akun GitHub .....	2
D. Langkah 2: Membuat Repository Baru di GitHub .....	2
E. Langkah 3: Mengatur Git di Komputer .....	2
F. Langkah 4: Menghubungkan GitHub dengan Visual Studio Code.....	3
G. Langkah 5: Login ke GitHub dari Visual Studio Code.....	3
H. Langkah 6: Memastikan Proyek Ter-clone dengan Benar .....	3
<b>BAB 2 . Instalasi Node.js</b> .....	4
Langkah 1: Mengunduh Node.js .....	4
Langkah 2: Menginstal Node.js di Windows.....	4
Langkah 3: Menginstal Node.js di macOS.....	4
Langkah 4: Menginstal Node.js di Linux.....	4
Langkah 5: Memeriksa Instalasi Node.js dan npm .....	5
<b>BAB 3. Express.js dan Vue.js</b> .....	6
Persiapan.....	6
Bagian 1: Instalasi dan Setup Express.js .....	6
Langkah 1: Membuat Proyek Express.js .....	6
Langkah 2: Membuat Server Express.js .....	6
Langkah 3: Membuat API untuk Profil Perusahaan.....	7
Bagian 2: Instalasi dan Setup Vue.js .....	7
Langkah 1: Membuat Proyek Vue.js .....	7
Langkah 2: Mengatur Struktur Proyek Vue.js .....	8
Langkah 3: Mengatur Routing di Vue.js .....	9
Langkah 4: Mengatur App.vue untuk Navigasi .....	10
Langkah 5: Menghubungkan Frontend dengan Backend .....	10
Langkah 6: Menjalankan Proyek Vue.js .....	11

# BAB 1 . Mengatur Proyek di GitHub

## A. Pendahuluan

Dalam tutorial ini, kita akan belajar bagaimana membuat dan mengatur proyek di GitHub serta melakukan clone proyek tersebut di Visual Studio Code (VS Code). Tutorial ini dirancang untuk pemula yang belum memiliki pengalaman dengan Git dan GitHub.

## B. Persiapan

Sebelum kita mulai, pastikan kamu telah menginstal:

1. [Git](#)
2. [Visual Studio Code](#)

Jika belum, silakan unduh dan instal terlebih dahulu.

## C. Langkah 1: Membuat Akun GitHub

1. Buka [GitHub](#) di browser.
2. Klik tombol "Sign up" untuk membuat akun baru.
3. Ikuti petunjuk pendaftaran dengan mengisi informasi yang diperlukan seperti email, username, dan password.

## D. Langkah 2: Membuat Repository Baru di GitHub

1. Login ke akun GitHub kamu.
2. Klik ikon '+' di pojok kanan atas dan pilih "New repository".
3. Isi nama repository (misalnya: proyek-pertama) dan deskripsi (opsional).
4. Pilih opsi "Public" agar repository dapat diakses oleh semua orang.
5. Centang opsi "Initialize this repository with a README".
6. Klik tombol "Create repository".

## E. Langkah 3: Mengatur Git di Komputer

1. Buka terminal atau command prompt.
2. Atur username Git dengan perintah berikut:

```
git config --global user.name "NamaKamu"
```

3. Atur email Git dengan perintah berikut:

```
git config --global user.email "emailkamu@example.com"
```

4. Pastikan pengaturan berhasil dengan perintah:

```
git config --global --list
```

Kamu akan melihat username dan email yang telah diset.

#### **F. Langkah 4: Menghubungkan GitHub dengan Visual Studio Code**

1. Buka Visual Studio Code.
2. Jika belum menginstal, instal ekstensi GitHub di VS Code dengan cara:
  - Klik ikon Extensions di sidebar kiri.
  - Cari "GitHub Pull Requests and Issues" dan klik "Install".
3. Klik ikon Source Control di sidebar kiri.
4. Klik ikon tiga titik di pojok kanan atas panel Source Control dan pilih "Clone".
5. Masukkan URL repository GitHub yang telah kamu buat. URL bisa ditemukan di halaman repository GitHub (misalnya: <https://github.com/username/proyek-pertama.git>).
6. Pilih lokasi di komputer untuk menyimpan repository yang di-clone.

#### **G. Langkah 5: Login ke GitHub dari Visual Studio Code**

1. Setelah memilih lokasi untuk clone repository, akan muncul pop-up untuk login ke GitHub.
2. Klik "Sign in with GitHub".
3. Browser akan terbuka dan meminta izin untuk mengakses akun GitHub. Klik "Authorize Visual Studio Code".
4. Kembali ke Visual Studio Code, kamu akan diminta untuk memasukkan token autentikasi. Ikuti petunjuk untuk menyelesaikan proses login.

#### **H. Langkah 6: Memastikan Proyek Ter-clone dengan Benar**

1. Setelah login, Visual Studio Code akan mulai meng-clone repository.
2. Setelah selesai, repository akan terbuka di VS Code.
3. Kamu bisa melihat semua file dari repository GitHub di panel Explorer di sebelah kiri.

## BAB 2 . Instalasi Node.js

Node.js adalah runtime JavaScript yang memungkinkan kamu menjalankan kode JavaScript di luar browser. Pada tutorial ini, kita akan belajar cara menginstal Node.js di komputer. Tutorial ini dirancang untuk pemula yang belum memiliki pengalaman dengan Node.js.

### Langkah 1: Mengunduh Node.js

1. Buka [situs resmi Node.js](#).
2. Kamu akan melihat dua versi Node.js yang tersedia untuk diunduh: LTS (Long Term Support) dan Current. Untuk pemula, disarankan untuk mengunduh versi LTS karena lebih stabil.
3. Klik tombol "LTS" untuk mengunduh installer Node.js sesuai dengan sistem operasi kamu (Windows, macOS, atau Linux).

### Langkah 2: Menginstal Node.js di Windows

1. Buka file installer yang telah diunduh (`node-vxx.x.x-x64.msi`).
2. Ikuti petunjuk instalasi dengan mengklik "Next".
3. Baca dan setuju lisensi dengan mencentang "I accept the terms in the License Agreement", lalu klik "Next".
4. Biarkan pengaturan instalasi default dan klik "Next".
5. Pada bagian Tools for Native Modules, centang "Automatically install the necessary tools...". Ini akan membantu jika kamu perlu mengkompilasi modul asli di masa mendatang.
6. Klik "Install" dan tunggu hingga proses instalasi selesai.
7. Klik "Finish" setelah instalasi selesai.

### Langkah 3: Menginstal Node.js di macOS

1. Buka file installer yang telah diunduh (`node-vxx.x.x.pkg`).
2. Ikuti petunjuk instalasi dengan mengklik "Continue".
3. Baca dan setuju lisensi dengan mengklik "Agree".
4. Klik "Install" untuk memulai proses instalasi.
5. Masukkan kata sandi administrator jika diminta.
6. Klik "Close" setelah instalasi selesai.

### Langkah 4: Menginstal Node.js di Linux

1. Buka terminal.
2. Tambahkan repository NodeSource menggunakan perintah berikut (sesuaikan dengan versi Node.js yang ingin diinstal):

```
curl -fsSL https://deb.nodesource.com/setup_lts.x | sudo -E -
```

3. Instal Node.js dengan perintah berikut:

```
sudo apt-get install -y nodejs
```

## Langkah 5: Memeriksa Instalasi Node.js dan npm

Setelah instalasi selesai, kita perlu memastikan bahwa Node.js dan npm (Node Package Manager) telah terinstal dengan benar.

1. Buka terminal atau command prompt.
2. Periksa versi Node.js dengan perintah berikut:

```
node -v
```

Kamu akan melihat versi Node.js yang terinstal (misalnya: `v16.14.0`).

3. Periksa versi npm dengan perintah berikut:

```
npm -v
```

Kamu akan melihat versi npm yang terinstal (misalnya: `8.3.1`).

## BAB 3. Express.js dan Vue.js

Dalam tutorial ini, kita akan belajar cara menginstal Express.js untuk backend dan Vue.js untuk frontend. Kemudian, kita akan membuat aplikasi Single Page Application (SPA) sederhana untuk profil perusahaan yang terdiri dari 2-3 halaman. Tutorial ini dirancang untuk pemula yang ingin memahami dasar-dasar penggunaan Express.js dan Vue.js.

### Persiapan

Sebelum memulai, pastikan kamu telah menginstal Node.js dan npm di komputermu. Jika belum, silakan ikuti tutorial sebelumnya tentang instalasi Node.js.

## Bagian 1: Instalasi dan Setup Express.js

### Langkah 1: Membuat Proyek Express.js

1. Buka terminal atau command prompt.
2. Buat direktori baru untuk proyek backend, misalnya `company-profile-backend`:

```
mkdir company-profile-backend  
cd company-profile-backend
```

3. Inisialisasi proyek Node.js:

```
npm init -y
```

4. Instal Express.js:

```
npm install express
```

### Langkah 2: Membuat Server Express.js

1. Buat file baru bernama `server.js` di direktori proyek.
2. Tambahkan kode berikut ke dalam `server.js` untuk membuat server sederhana:

```
javascript  
  
const express = require('express');  
const app = express();  
const port = 3000;  
  
app.use(express.json());  
  
app.get('/', (req, res) => {  
  res.send('Welcome to Company Profile API');  
});
```

```
});  
  
app.listen(port, () => {  
  console.log(`Server is running on http://localhost:${port}`);  
});
```

3. Jalankan server dengan perintah berikut:

```
node server.js
```

4. Buka browser dan akses `http://localhost:3000`. Kamu akan melihat pesan "Welcome to Company Profile API".

### Langkah 3: Membuat API untuk Profil Perusahaan

1. Tambahkan endpoint untuk mendapatkan informasi profil perusahaan. Ubah file `server.js` menjadi:

```
javascript  
  
const express = require('express');  
const app = express();  
const port = 3000;  
  
app.use(express.json());  
  
const companyProfile = {  
  name: "PT GIT Solution",  
  description: "Solusi Teknologi Terbaik untuk Bisnis Anda",  
  services: ["Pengembangan Software", "Konsultasi IT", "Manajemen  
Proyek"]  
};  
  
app.get('/', (req, res) => {  
  res.send('Welcome to Company Profile API');  
});  
  
app.get('/profile', (req, res) => {  
  res.json(companyProfile);  
});  
  
app.listen(port, () => {  
  console.log(`Server is running on http://localhost:${port}`);  
});
```

2. Buka browser dan akses `http://localhost:3000/profile`. Kamu akan melihat data profil perusahaan dalam format JSON.

## Bagian 2: Instalasi dan Setup Vue.js

### Langkah 1: Membuat Proyek Vue.js

1. Buka terminal atau command prompt baru.



2. Buat direktori baru untuk proyek frontend, misalnya `company-profile-frontend`:

```
mkdir company-profile-frontend
cd company-profile-frontend
```

3. Inisialisasi proyek Vue.js menggunakan Vue CLI. Jika belum menginstal Vue CLI, instal dengan perintah berikut:

```
npm install -g @vue/cli
```

4. Buat proyek Vue.js baru:

```
vue create .
```

Ikuti petunjuk dan pilih pengaturan default.

## Langkah 2: Mengatur Struktur Proyek Vue.js

1. Hapus semua file di dalam folder `src/components`.
2. Buat tiga komponen baru untuk halaman SPA:
  - o `Home.vue`
  - o `About.vue`
  - o `Services.vue`
3. Tambahkan kode berikut ke masing-masing file komponen:

### Home.vue

```
vue

<template>
  <div>
    <h1>Welcome to PT GIT Solution</h1>
    <p>Solusi Teknologi Terbaik untuk Bisnis Anda</p>
  </div>
</template>

<script>
export default {
  name: 'Home'
}
</script>
```

### About.vue

```
vue

<template>
  <div>
    <h1>Tentang Kami</h1>
```

```
        <p>PT GIT Solution adalah perusahaan teknologi yang
menyediakan solusi IT terbaik untuk berbagai bisnis.</p>
```

```
    </div>
```

```
</template>
```

```
<script>
```

```
export default {
  name: 'About'
}
```

```
</script>
```

## Services.vue

```
vue
```

```
<template>
```

```
  <div>
```

```
    <h1>Layanan Kami</h1>
```

```
    <ul>
```

```
      <li>Pengembangan Software</li>
```

```
      <li>Konsultasi IT</li>
```

```
      <li>Manajemen Proyek</li>
```

```
    </ul>
```

```
  </div>
```

```
</template>
```

```
<script>
```

```
export default {
  name: 'Services'
}
```

```
</script>
```

## Langkah 3: Mengatur Routing di Vue.js

1. Buka file `src/router/index.js` dan ubah menjadi seperti berikut:

```
javascript
```

```
import Vue from 'vue'
```

```
import VueRouter from 'vue-router'
```

```
import Home from '../components/Home.vue'
```

```
import About from '../components/About.vue'
```

```
import Services from '../components/Services.vue'
```

```
Vue.use(VueRouter)
```

```
const routes = [
```

```
  {
```

```
    path: '/',
```

```
    name: 'Home',
```

```
    component: Home
```

```
  },
```

```
  {
```

```
    path: '/about',
```

```
    name: 'About',
```

```
    component: About
```

```
  },
```

```
  {
```

```
    path: '/services',
```

```

        name: 'Services',
        component: Services
      }
    ]

    const router = new VueRouter({
      mode: 'history',
      base: process.env.BASE_URL,
      routes
    })

    export default router

```

## Langkah 4: Mengatur App.vue untuk Navigasi

1. Buka file `src/App.vue` dan ubah menjadi seperti berikut:

```

vue

<template>
  <div id="app">
    <nav>
      <ul>
        <li><router-link to="/">Home</router-link></li>
        <li><router-link to="/about">About</router-link></li>
        <li><router-link to="/services">Services</router-
link></li>
      </ul>
    </nav>
    <router-view/>
  </div>
</template>

<script>
export default {
  name: 'App'
}
</script>

<style>
nav ul {
  list-style-type: none;
  padding: 0;
}
nav ul li {
  display: inline;
  margin-right: 10px;
}
</style>

```

## Langkah 5: Menghubungkan Frontend dengan Backend

1. Instal Axios untuk melakukan HTTP request:

```
npm install axios
```

2. Buka file `src/components/Home.vue` dan ubah menjadi seperti berikut untuk mengambil data dari backend:

```
vue

<template>
  <div>
    <h1>Welcome to {{ profile.name }}</h1>
    <p>{{ profile.description }}</p>
  </div>
</template>

<script>
import axios from 'axios'

export default {
  name: 'Home',
  data() {
    return {
      profile: {}
    }
  },
  created() {
    axios.get('http://localhost:3000/profile')
      .then(response => {
        this.profile = response.data
      })
      .catch(error => {
        console.log(error)
      })
  }
}
</script>
```

## Langkah 6: Menjalankan Proyek Vue.js

1. Jalankan proyek Vue.js dengan perintah berikut:

```
npm run serve
```

2. Buka browser dan akses `http://localhost:8080`. Kamu akan melihat halaman Home dengan data yang diambil dari backend.