



Instalasi Alat Bantu & Pengantar Algoritma



Pendahuluan & Kontrak Kuliah

Mata kuliah ini bukan sekadar belajar membuat website, melainkan mempersiapkan Anda untuk memiliki kompetensi standar **Junior Web Developer**. Struktur kurikulum yang digunakan telah mengacu pada **SKKNI (Standar Kompetensi Kerja Nasional Indonesia) Nomor 282 Tahun 2016**.

- Link **Keputusan Menteri Ketenagakerjaan Nomor 282 Tahun 2016**
<https://jdih.kemnaker.go.id/peraturan/detail/1381/keputusan-menaker-nomor-282-tahun-2016>
- Link **Pengembang Web Pratama (Junior Web Developer)**
<https://proglat.kemnaker.go.id/programs/4afe93df-edf5-4cf9-bcbb-6813ffb0b2db/versions/1>

Berikut adalah Unit Kompetensi yang akan kita pelajari, yang diambil langsung dari standar sertifikasi Junior Web Developer:

| Kode Unit | Judul Unit Kompetensi (SKKNI) |
|-----------------|---|
| J.620100.005.02 | Mengimplementasikan User Interface |
| J.620100.010.01 | Menerapkan Perintah Eksekusi Pemrograman |
| J.620100.015.01 | Menyusun Fungsi & Sumber Daya yang Rapi |
| J.620100.016.01 | Menulis Kode sesuai Guidelines |
| J.620100.017.02 | Mengimplementasikan Pemrograman Terstruktur |
| J.620100.019.02 | Menggunakan Library Pre-existing |

Jika Anda lulus mata kuliah ini dengan baik, Anda telah memenuhi prasyarat pengetahuan, keterampilan, dan sikap kerja untuk mengikuti **Sertifikasi Kompetensi Junior Web Developer** dari BNSP atau program seperti VSGA (Vocational School Graduate Academy).

Contoh **Sertifikasi Kompetensi Junior Web Developer** adalah sebagai berikut:



Tata Tertib & Kehadiran Perkuliahan

- **Syarat Mengikuti UAS:** Wajib hadir minimal **70%** (batas maksimal Alfa adalah 3 kali dari total 14 kali pertemuan).
- **Sanksi Kehadiran Rendah (50%):** Kehilangan hak untuk melakukan *crosscheck* nilai kepada dosen bersangkutan.
- **Sanksi Pemalsuan Absen:** Apabila terbukti melakukan kehadiran palsu (titip absen), nilai mata kuliah otomatis menjadi **"D"**.
- **Ujian Tidak Lengkap:** Jika mahasiswa hanya mengikuti salah satu ujian utama (hanya UTS atau hanya UAS), nilai maksimal yang bisa didapatkan

adalah "C".

- Proses *crosscheck* nilai hanya akan dilayani sesuai jadwal yang telah ditentukan.
- Pemberian Tugas dan Kuis dapat dilakukan sewaktu-waktu oleh dosen tanpa pemberitahuan sebelumnya.
- Wajib berpakaian sopan serta menjaga kebersihan dan kerapian selama perkuliahan (teori maupun praktik).

Persyaratan Mengikuti Ujian (UTS & UAS)

- **Pengambilan Kartu Ujian:** Setiap mahasiswa wajib membawa Buku Pedoman Akademik sebagai syarat mengambil Kartu Ujian.
- **Syarat Absensi Ujian:** Kehadiran minimal **70%** dari total kehadiran dosen pengampu.
- **Syarat Bimbingan:** Wajib melakukan Bimbingan Akademik minimal 2 kali sebelum UTS dan 2 kali sebelum UAS (harus terlampir/tercatat di Buku Pedoman Akademik).
- **Syarat Administrasi:** Wajib melunasi pembayaran SPP perkuliahan semester berjalan sesuai catatan di Buku Pedoman Akademik.



Catatan Penting: Pemberian Surat Peringatan (SP 1, SP 2, dan SP 3) adalah wewenang Ketua Program Studi, yang diputuskan berdasarkan rekap absensi harian dari tim Akademik.

Skema & Skala Penilaian

Nilai akhir Anda akan ditentukan dari gabungan beberapa komponen dengan bobot berikut:

| Komponen Penilaian | Persentase Bobot |
|--------------------|------------------|
| Kehadiran | 10% |
| Perilaku | 10% |

| Komponen Penilaian | Persentase Bobot |
|--------------------|------------------|
| Tugas | 20% |
| UTS | 25% |
| UAS | 35% |

Total nilai akhir kemudian akan dikonversi menjadi huruf mutu dengan rentang berikut:

| Rentang Nilai Angka | Nilai Huruf |
|---------------------|-------------|
| 85 - 100 | A |
| 80 - 84,99 | A- |
| 75 - 79,99 | B+ |
| 70 - 74,99 | B |
| 65 - 69,99 | B- |
| 60 - 64,99 | C+ |
| 55 - 59,99 | C |
| 40 - 54,99 | D |
| 0 - 39,99 | E |



Pendahuluan Pemrograman Web

Pemrograman adalah sebuah proses yang mencakup kegiatan seperti analisis masalah dan penyusunan algoritma, yang kemudian diterjemahkan ke dalam bahasa pemrograman agar instruksi tersebut dapat dijalankan oleh komputer. Membuat program komputer serupa dengan menciptakan musik atau merancang rumah, sehingga pemrograman juga sering dianggap sebagai sebuah seni yang melibatkan kreativitas dan imajinasi

Mengapa Anda Harus Belajar Pemrograman?

- **Melatih Pemecahan Masalah:** Pemrograman mengajarkan cara berpikir dengan memecah masalah besar menjadi bagian-bagian kecil yang lebih

mudah diselesaikan.

- **Berdampak pada Karir dan Finansial:** Pertumbuhan karir di bidang ini terus menanjak, dan belajar pemrograman membuka peluang untuk mencapai kesuksesan finansial yang signifikan.
- **Mewujudkan Ide Menjadi Nyata:** Jika Anda memiliki gagasan untuk memperbaiki lingkungan kerja atau masalah sosial, Anda dapat menggunakan teknologi untuk mentransformasikan gagasan tersebut menjadi aplikasi nyata.
- **Membangun Kepercayaan Diri:** Memiliki kemampuan untuk menyelesaikan sesuatu yang sangat kompleks dapat meningkatkan rasa percaya diri.

Banyak yang mengira bahwa untuk menjadi *programmer* harus jago matematika tingkat lanjut. Faktanya, matematika kompleks hanya sangat dibutuhkan jika Anda membuat *game* 3D, pemrosesan gambar, atau bahasa tingkat rendah (*low-level*). Untuk bidang lainnya, yang paling penting adalah memiliki logika yang baik dan kemampuan memecahkan masalah secara sistematis. Selain itu, siapapun bisa menjadi *programmer*, termasuk mereka yang berasal dari latar belakang non-IT atau anak-anak, selama ada kemauan keras untuk belajar. Kemampuan minimal yang wajib dikuasai justru adalah Bahasa Inggris, khususnya pada kemampuan membaca (*Reading*) dan mendengar (*Listening*), untuk mempelajari dokumentasi resmi.

Sejarah Internet dan Lahirnya World Wide Web (WWW)

Sejarah web sangat erat dengan perkembangan internet, yang berawal dari era Perang Dingin pada tahun 1966. Saat itu, Departemen Pertahanan Amerika Serikat melalui lembaga ARPA mendirikan proyek **ARPANET** untuk menciptakan jaringan komputer desentralisasi sebagai ketahanan militer. Karena pesatnya perkembangan universitas yang ingin bergabung, ARPANET dipecah menjadi MILNET (untuk militer) dan ARPANET (untuk sipil/universitas), yang gabungannya kemudian dikenal sebagai Internet.

Pada tahun 1970, protokol **TCP/IP** dikembangkan oleh Vinton Cerf dan Robert Kahn agar berbagai komputer dapat saling berkomunikasi. Kemudian, pada tahun 1983 diperkenalkan **DNS (Domain Name System)** yang memungkinkan alamat server diakses dengan nama domain yang mudah diingat (seperti *www.example.com*), alih-alih angka IP yang rumit.

Puncak kelahiran web terjadi pada tahun 1989 ketika ilmuwan Tim Berners-Lee di CERN menciptakan konsep **World Wide Web (WWW)**. Ia menciptakan tiga teknologi kunci yang menjadi landasan web hingga hari ini:

1. **HTML (Hypertext Markup Language)**: Bahasa untuk membuat struktur dan tampilan halaman web.
2. **HTTP (Hypertext Transfer Protocol)**: Protokol pengiriman data antar perangkat di internet.
3. **URL (Uniform Resource Locator)**: Alamat unik untuk mengidentifikasi setiap halaman web.

Generasi Web:

- **Web 1.0**: Era awal web statis, dimana pembuat memberi informasi dan pengguna hanya sebagai pembaca pasif.
- **Web 2.0**: Era jejaring sosial (seperti Friendster, Facebook, Twitter), dimana pengguna dapat berpartisipasi aktif berbagi konten.
- **Web 3.0**: Era integrasi Kecerdasan Buatan (AI), *Metaverse*, *Augmented Reality* (AR), dan *Internet of Things* (IoT) dimana internet dapat memahami konten dan perangkat berkomunikasi langsung dengan web

Arsitektur dan Klasifikasi Website

Website adalah kumpulan halaman web terkait yang memuat teks, gambar, video, atau elemen interaktif, dan dapat diakses melalui internet. Dalam sistem kerjanya, terdapat arsitektur **Client-Server**:

- **Client Side (Front-end)**: Pengguna mengakses *browser* dan memasukkan URL. Semua pengolahan tampilan fisik web diterjemahkan secara langsung oleh *browser* pengguna.
- **Server Side (Back-end)**: Server akan menerima permintaan (*HTTP Request*) dari *client*, memproses data (seperti mengambil dari database), lalu mengirimkan kembali balasannya (*HTTP Response*) ke *browser* untuk ditampilkan.

Agar web bisa mengudara, ia memerlukan **Web Hosting** (penyewaan tempat menyimpan file/database web) dan **Domain** (nama unik alamat web).

Jenis Website Berdasarkan Sifatnya:

1. **Web Statis:** Kontennya tetap, menggunakan HTML murni tanpa database, dan pengguna tidak bisa berinteraksi dengan sistem (contoh: portofolio, *landing page*).
2. **Web Dinamis:** Menampilkan informasi yang bisa berinteraksi dengan pengguna menggunakan form dan database (contoh: situs berita, portal kampus).
3. **Web Interaktif:** Memungkinkan pengguna saling berinteraksi secara aktif satu sama lain (contoh: jejaring sosial dan *game online*).

Secara kepemilikan, website bisa berupa **Web Pribadi** (blog, portofolio), **Web Bisnis/Perusahaan** (menampilkan identitas *brand*), **Web Komunitas** (forum diskusi), dan **Web E-Commerce** (platform jual-beli *online*).

Spesialisasi Programmer

Berdasarkan jenis aplikasi yang dibuat, *programmer* terbagi menjadi beberapa profesi:

1. **Web Programmer:** Membuat website dari yang berukuran kecil hingga sistem informasi yang sangat kompleks. Dibagi lagi menjadi *Frontend Developer* (fokus pada tampilan web dengan HTML/CSS/JS) dan *Backend Developer* (fokus di balik layar mengatur server, aplikasi, dan database). Ahli yang menguasai keduanya disebut *Fullstack Developer*.
2. **Mobile Programmer:** Menciptakan perangkat lunak untuk ponsel pintar, tablet, atau *smartwatch*, yang terbagi ke spesialisasi sistem operasi Android atau iOS.
3. **Desktop Programmer:** Membuat aplikasi yang diinstal langsung di komputer (seperti editor gambar atau aplikasi kasir internal) yang dapat berjalan sangat cepat tanpa dependensi *browser*.
4. **Hardware/Embedded System Programmer:** Memprogram perangkat keras dan mikrokontroler menggunakan bahasa tingkat rendah (Assembly) atau C/C++ untuk menghubungkan alat dengan sensor secara fisik

Bahasa Pemrograman dan Kerangka Kerja (Framework)

Bahasa pemrograman digolongkan menjadi *Machine Language* (kode biner 0 dan 1), *Assembly Language* (bahasa tingkat rendah untuk perangkat keras), dan *High-Level Language* (bahasa tingkat tinggi yang dekat dengan bahasa manusia seperti Python, JavaScript, Java).

1. Bahasa Front-end Web (Client-side)

- **HTML:** Membangun kerangka dan struktur konten web.
- **CSS:** Mendesain tampilan web (warna, tata letak, efek visual).
- **JavaScript:** Bahasa serbaguna yang menambahkan interaktivitas, animasi, dan logika pada *front-end* secara *real-time*.
- **TypeScript:** Versi perluasan JavaScript dengan fitur tipe data statis agar penulisan kode proyek kompleks lebih aman.

2. Bahasa Back-end Web (Server-side)

- **PHP:** Sangat populer, infrastrukturnya murah, dan terintegrasi baik dengan database. *Framework* populer: **Laravel** (memiliki struktur MVC, ORM Eloquent, dan keamanan mumpuni) serta **CodeIgniter** (ringan, minimalis, dan sangat cepat).
- **Python:** Mudah ditulis dan dibaca, sangat tangguh. *Framework* populer: **Django** (kerangka kerja lengkap dengan arsitektur MVT dan antarmuka admin bawaan) serta **Flask** (fleksibel, minimalis, dan cocok untuk API RESTful).
- **JavaScript (Node.js):** Memungkinkan JavaScript dijalankan di *server*, memiliki arsitektur *asynchronous non-blocking* sehingga handal menangani banyak permintaan sekaligus. *Framework* populer: **Express.js**.
- **Ruby:** Sangat produktif menggunakan *framework Ruby on Rails* yang menganut filosofi "konvensi lebih dari konfigurasi".
- **Java:** Tangguh dan stabil untuk aplikasi skala korporat besar (*enterprise*) berskala tinggi.

3. Database dan Bahasa Query

- **SQL:** Bahasa standar untuk memanipulasi database relasional (seperti MySQL).
- **NoSQL:** Basis data non-relasional, contohnya MongoDB yang menggunakan format mirip JSON.

- **GraphQL:** Bahasa *query* modern agar aplikasi dapat meminta spesifik data yang dibutuhkan saja, sehingga beban *server* berkurang.



Persiapan Perangkat dan Alat Pendukung (Tools)

Agar bisa mulai praktik, spesifikasi minimal yang dibutuhkan adalah PC/Laptop dengan prosesor Dual Core dan RAM 2GB (untuk pemrograman web). Untuk alat lunaknya, Anda membutuhkan:

Text Editor

Text editor adalah kanvas utama tempat Anda menulis baris-baris kode HTML, CSS, dan logika pemrograman. Beberapa pilihan yang bisa Anda gunakan:

Tempat Anda menulis kode.

- **Notepad / Wordpad:** Bawaan dasar Windows, namun fitur terbatas.
- **Notepad++:** Sangat populer di Windows, gratis, ringan, dan memiliki fitur pewarnaan kode (*syntax highlighting*).
- **Sublime Text:** Nyaman, kaya fitur, namun bersifat berbayar (ada versi uji coba/trial).
- **Visual Studio Code (VS Code)** adalah editor kode sumber (source code editor) yang ringan, gratis, dan *open-source* yang dikembangkan oleh Microsoft.

Menginstal Visual Studio Code

Download Visual Studio Code

Langkah pertama untuk memulai perjalanan Anda sebagai pengembang web adalah dengan menginstal Visual Studio Code. Akses [website resmi Visual Studio Code](#) dan klik tombol **Download** untuk platform Windows.

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

The image shows the download page for Visual Studio Code. It features three main sections: Windows, Linux, and Mac. Each section has a download button and a list of available installers and their supported architectures.

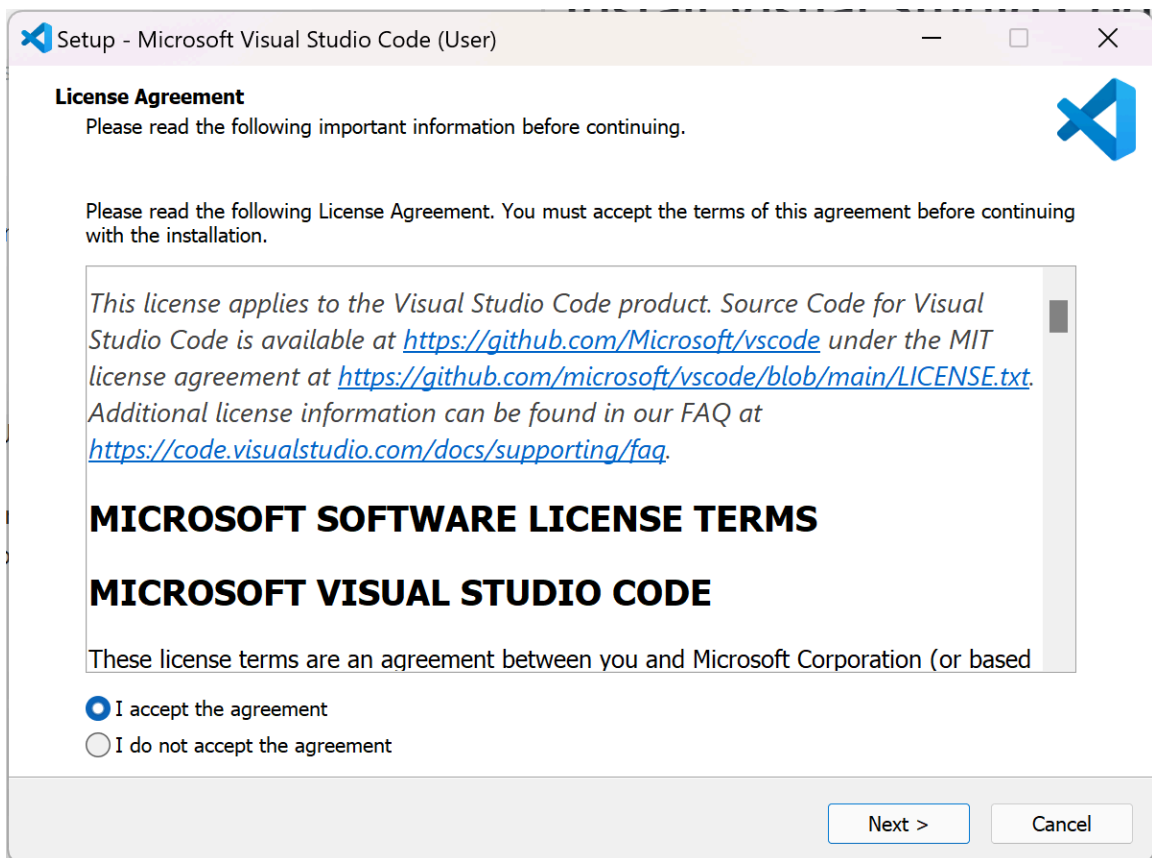
| Platform | OS/Version | Installer Type | Architectures |
|----------|-----------------------|------------------|--------------------------------------|
| Windows | Windows 10, 11 | User Installer | x64, x86, Arm64 |
| | | System Installer | x64, x86, Arm64 |
| | | .zip | x64, x86, Arm64 |
| | | CLI | x64, x86, Arm64 |
| Linux | Debian, Ubuntu | .deb | x64, Arm32, Arm64 |
| | | .rpm | x64, Arm32, Arm64 |
| | Red Hat, Fedora, SUSE | .rpm | x64, Arm32, Arm64 |
| | | .tar.gz | x64, Arm32, Arm64 |
| | Snap | Snap | Snap Store |
| | | CLI | x64, Arm32, Arm64 |
| Mac | macOS 10.11+ | .zip | Intel chip, Apple silicon, Universal |
| | | CLI | Intel chip, Apple silicon |

Setelah itu, tunggu hingga proses download selesai.

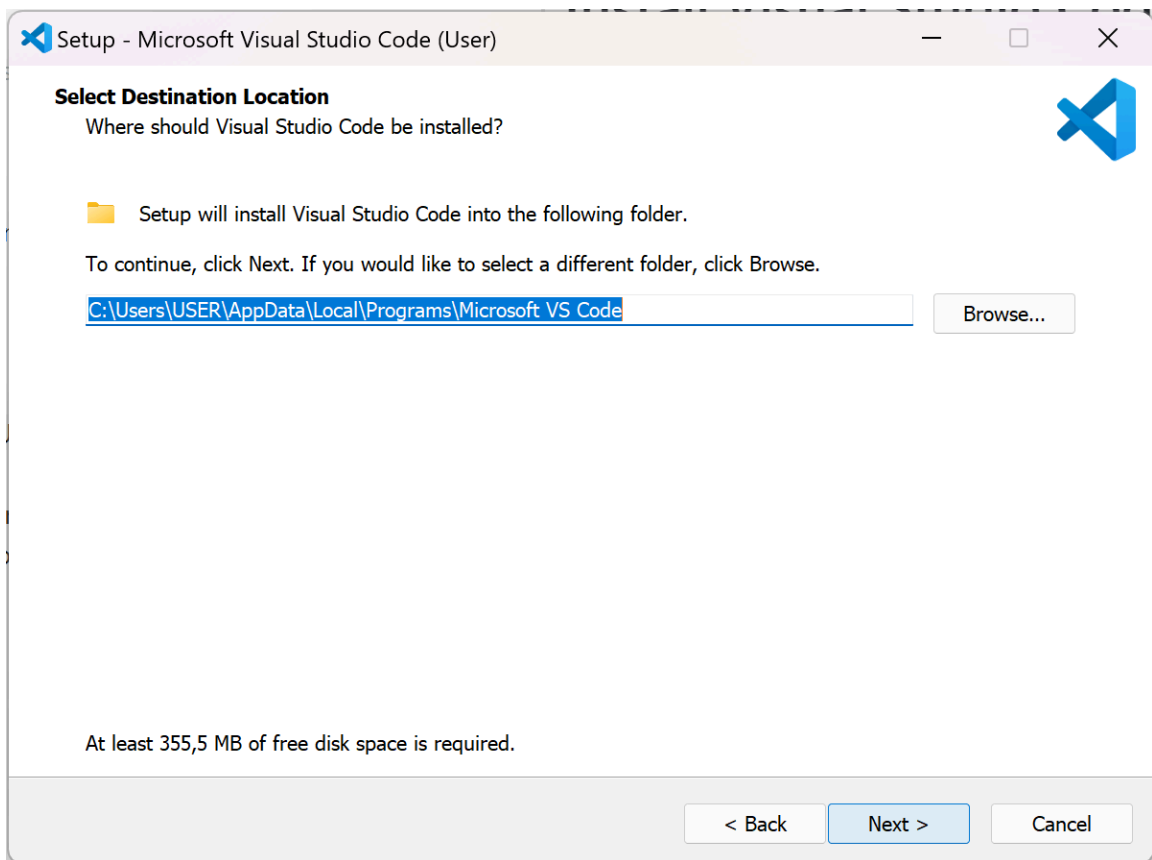
Proses Instalasi

Setelah file installer `VSCodeUserSetup-x64-1.82.2` berhasil diunduh, klik dua kali untuk memulai proses instalasi.

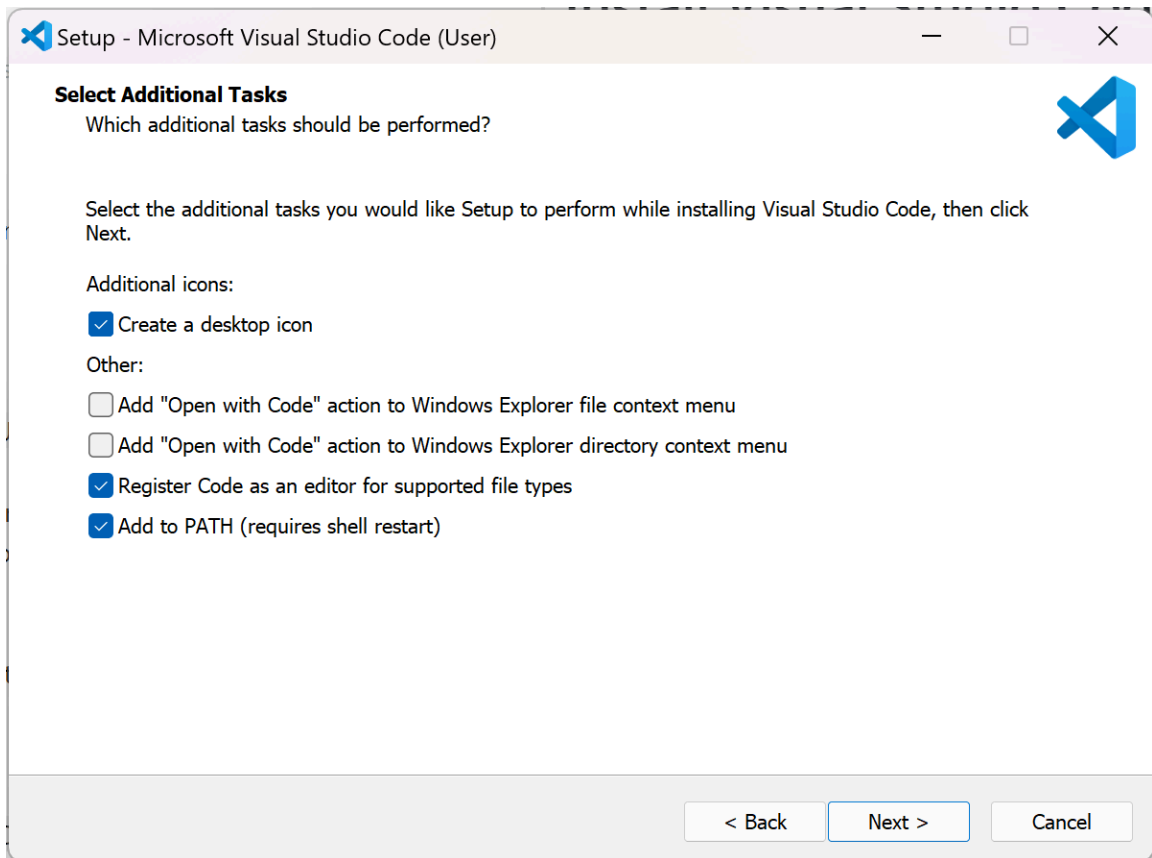
1. Pada halaman pertama instalasi, Anda akan diminta untuk menerima **License Agreement**. Pilih opsi **I accept the agreement** dan klik **Next**.



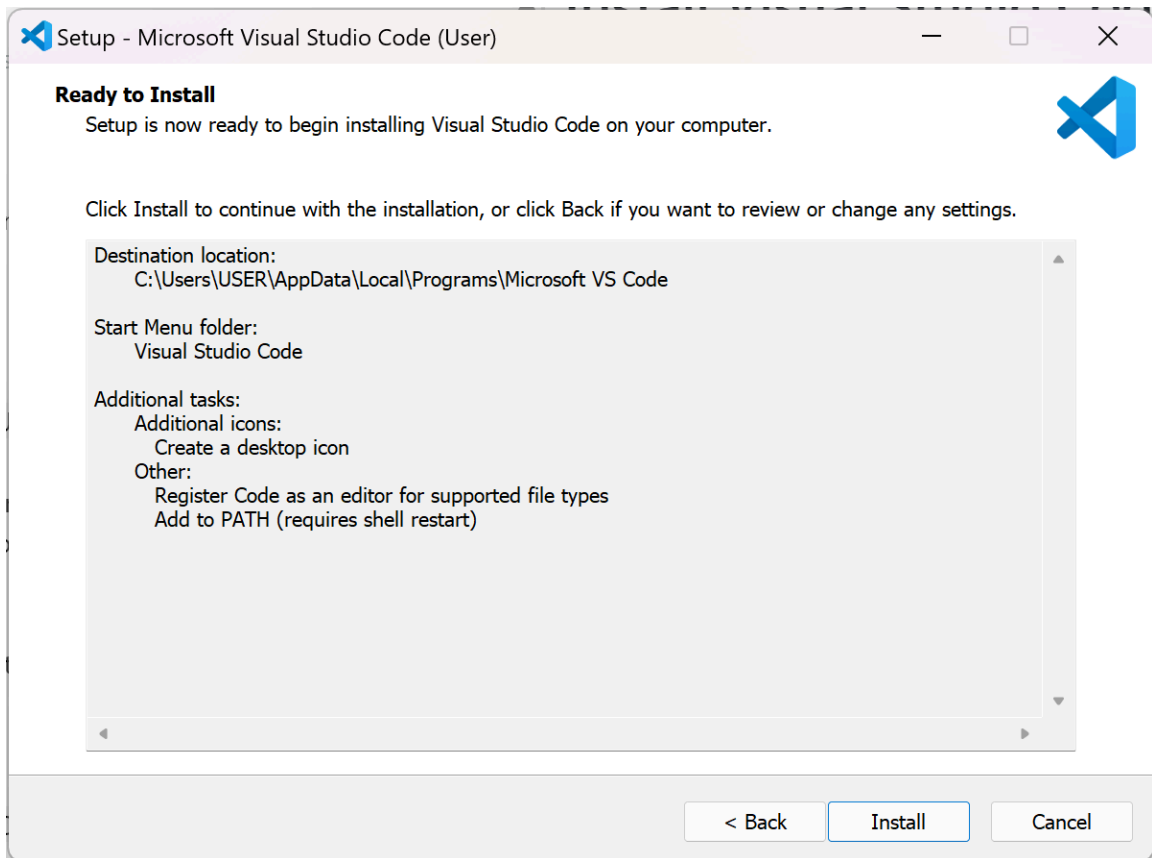
2. Pilih direktori tempat Anda ingin menginstal VS Code. Default-nya adalah di `C:\Program Files\Microsoft VS Code`. Anda bisa menggunakan pengaturan default atau menyesuaikan direktori sesuai keinginan Anda. Klik **Next**.



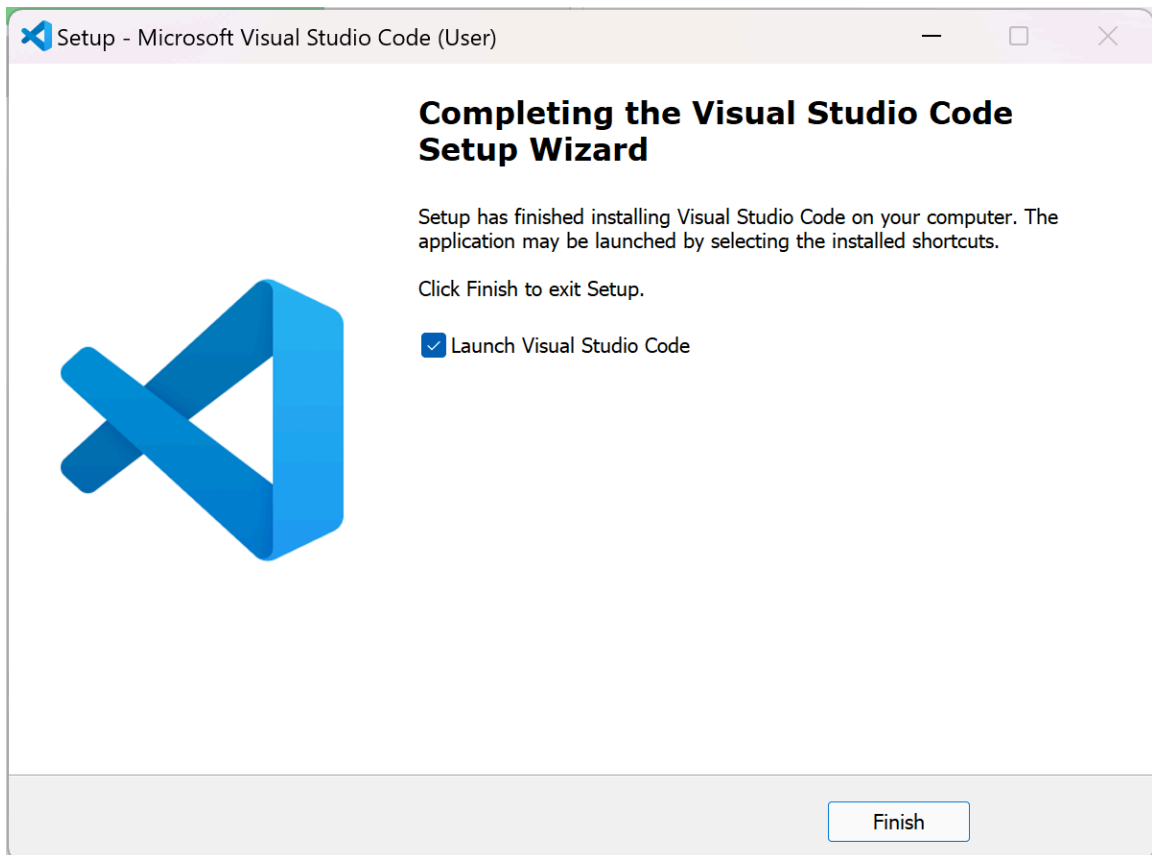
3. Pada halaman **Select Additional Tasks**, Anda dapat mencentang opsi **Create a desktop icon** jika Anda menginginkan shortcut di desktop. Lalu klik **Next**.



4. Pada halaman **Ready to Install**, klik **Install** dan tunggu hingga proses selesai.



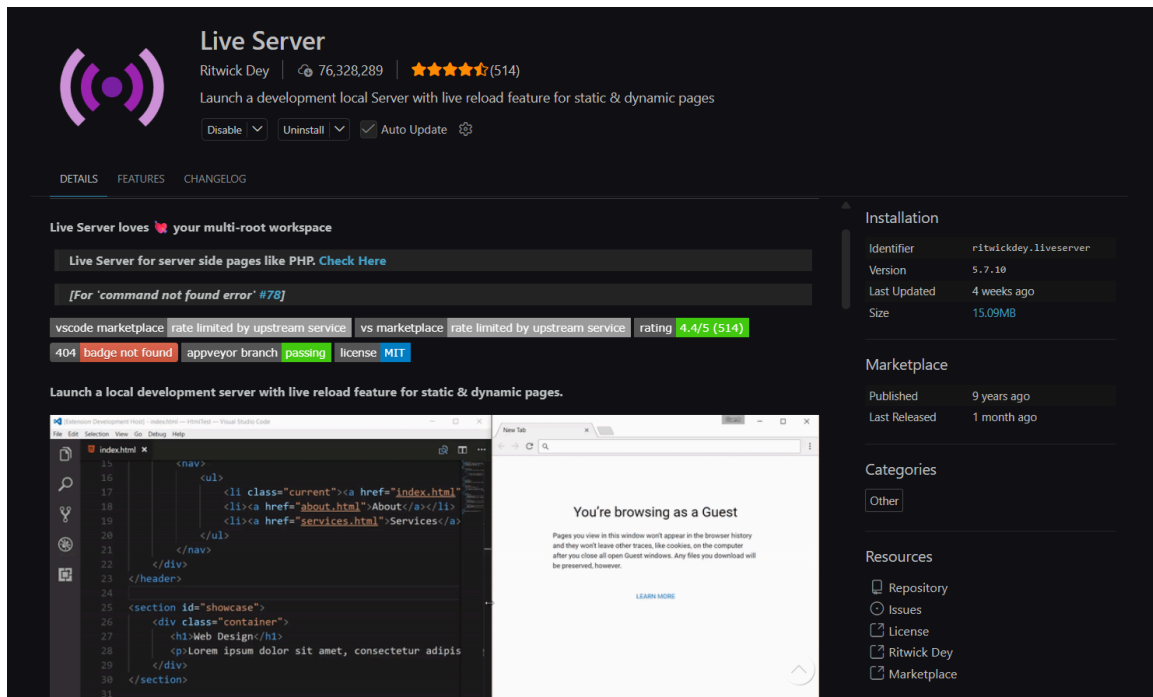
5. Setelah instalasi selesai, klik **Finish** untuk menutup installer.



10 VS Code Extension untuk Optimalkan Efisiensi Pengembangan

Dengan bantuan ekstensi yang tepat, VS Code bisa meningkatkan efisiensi dan mengoptimalkan pengembangan web. Berikut adalah 10 VS Code *extension* yang bisa mengoptimalkan efisiensi pengembangan:

1. Live Server

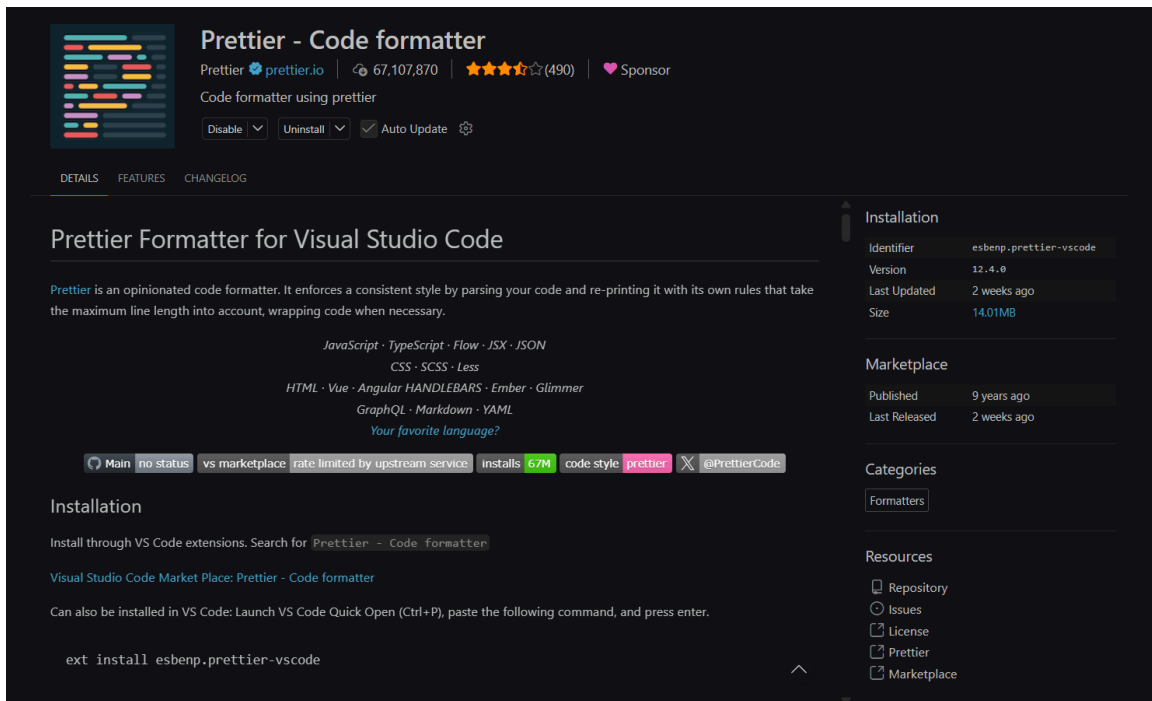


Pertama adalah Live Server. Ekstensi ini memungkinkan pengembangan *real-time* dengan memuat ulang halaman web secara otomatis setiap kali kamu menyimpan perubahan pada kodemu.

Ekstensi ini cocok bagi pengembang yang bekerja pada proyek HTML, CSS, dan JavaScript. Live Server khususnya cocok bagi pengembang yang ingin melihat perubahan secara langsung di *browser*.

Fungsi utama Live Server adalah membuka *server* lokal dan melakukan *live reload* pada *browser*. Secara keseluruhan, ini memudahkan pengembangan interaktif dan cepat.

2. Prettier - Code formatter

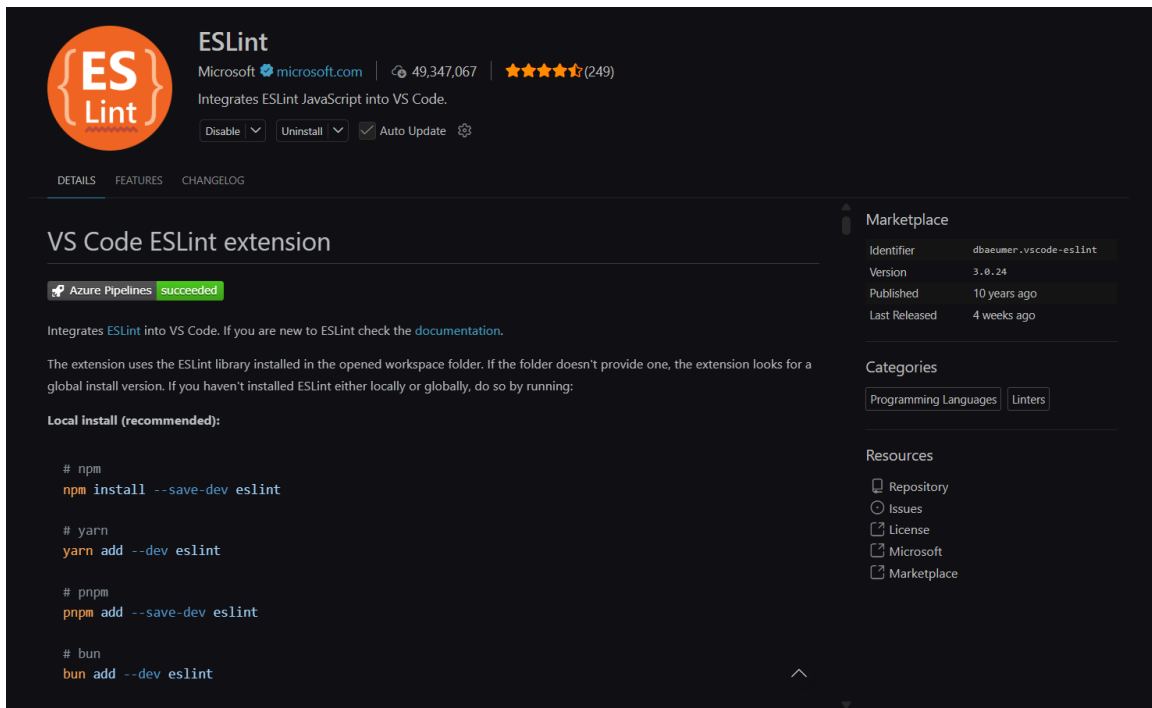


Berikutnya adalah Prettier. Ekstensi ini membantu menjaga konsistensi gaya kode. Hal ini dilakukan dengan secara otomatis memformat kode sesuai dengan aturan yang ditetapkan.

Prettier cocok untuk pengembang yang bekerja dalam tim atau proyek dimana konsistensi gaya kode adalah prioritas.

Secara keseluruhan, Prettier berfungsi untuk memformat kode secara otomatis. Tujuannya adalah untuk menjaga konsistensi dan meningkatkan keterbacaan.

3. ESLint

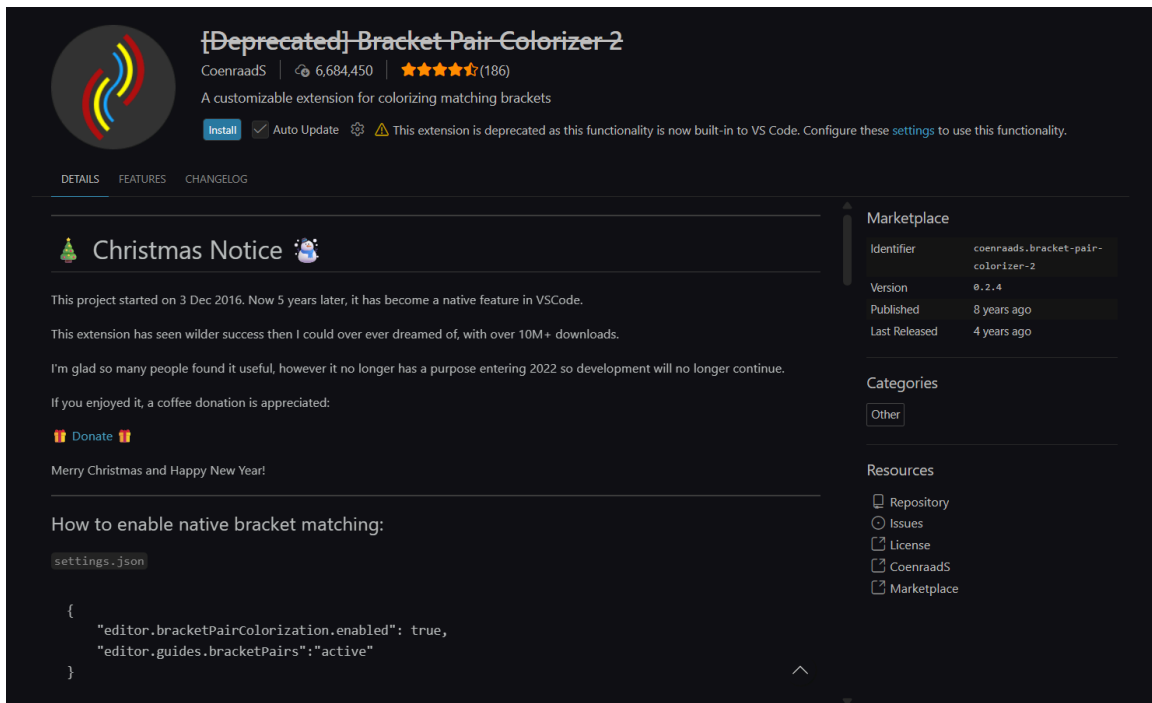


VS Code *extension* berikutnya ESLint. Ekstensi ini adalah alat linting yang membantu pengembang mengidentifikasi dan memperbaiki masalah dalam kode JavaScript.

Beberapa kesalahan yang bisa diperbaiki antara lain masalah sintaks atau pola kode yang tidak efisien. Oleh karena itu, ini cocok bagi pengembang JavaScript yang ingin memastikan kode mereka bersih, efisien, dan mengikuti *best practices*.

Fungsi utama ESLint adalah untuk menyoroti kesalahan sintaks dan masalah gaya kode. Hal ini memungkinkan pengembangan yang lebih aman dan terorganisir.

4. Bracket Pair Colorizer

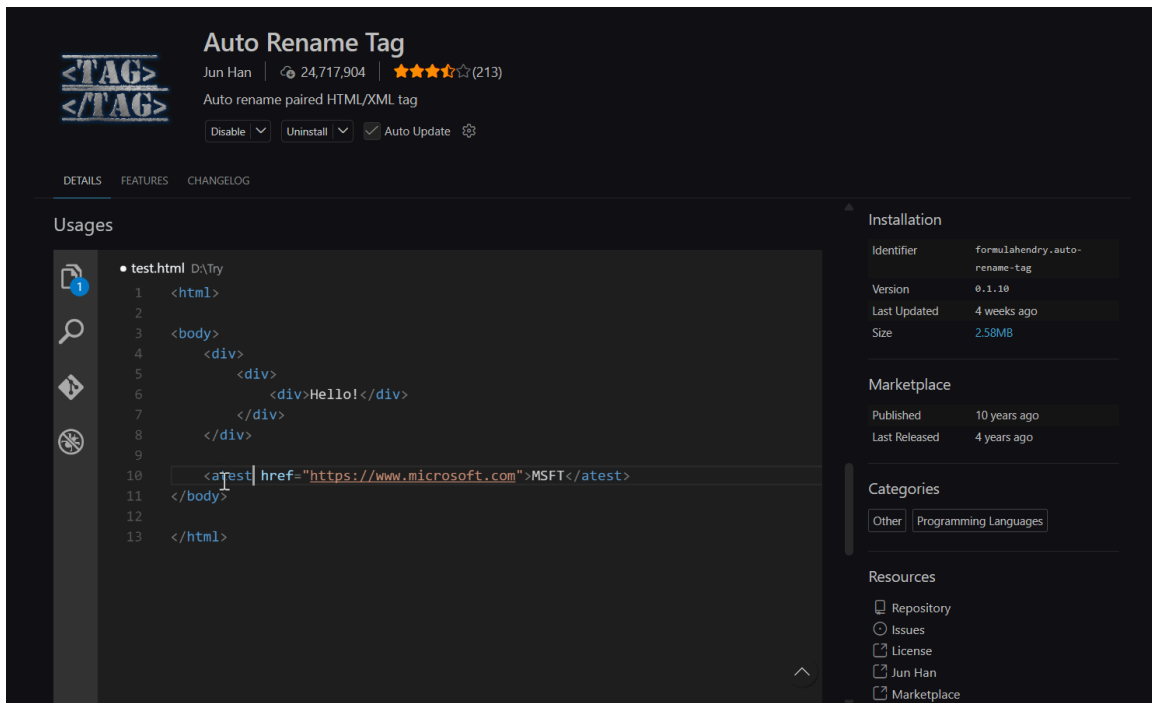


Berikutnya adalah Bracket Pair Colorizer. Ekstensi ini mewarnai pasangan kurung di kode pengembang. Pewarnaan membuat pengembang lebih mudah untuk mengikuti struktur blok kode.

Ekstensi ini cocok bagi pengembang yang bekerja dengan kode yang kompleks. Ini tentunya juga sesuai dengan *developer* yang ingin mempercepat pemahaman terhadap struktur kode.

Fungsi utamanya adalah untuk memudahkan navigasi dan pemahaman blok kode dengan visualisasi yang lebih jelas.

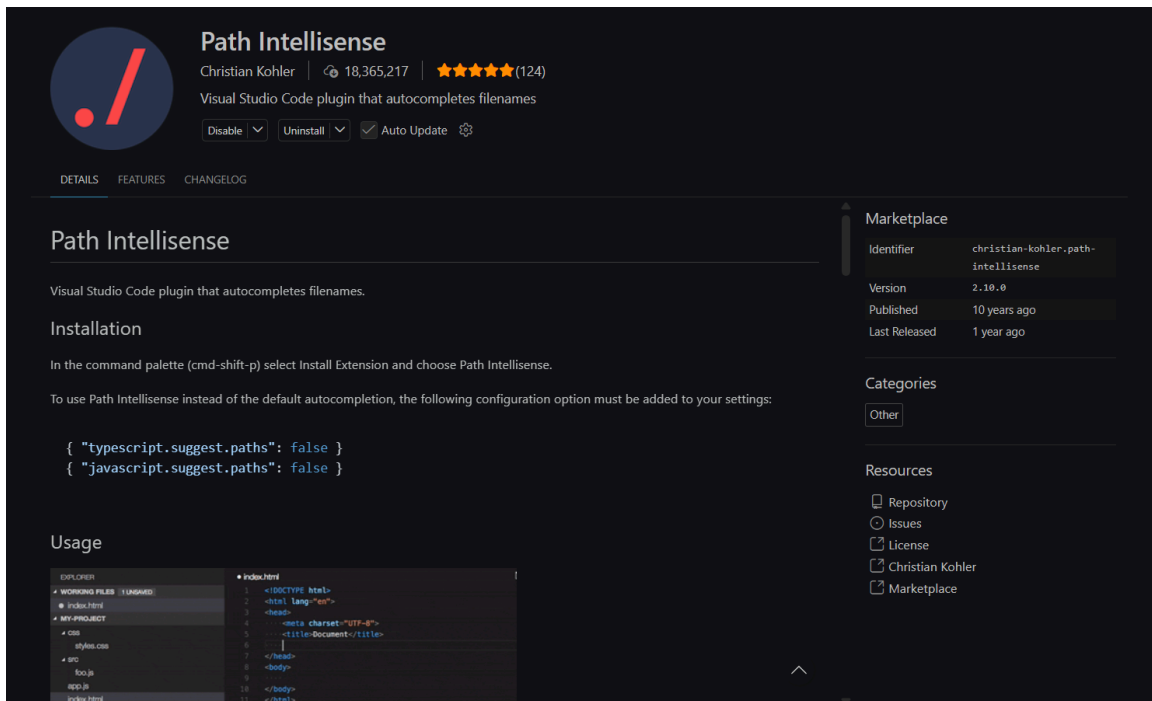
5. Auto Rename Tag



VS Code *extension* berikutnya adalah Auto Rename Tag. Ini secara otomatis mengganti nama pasangan *tag* pembuka dan penutup di HTML dan XML saat kamu mengedit salah satunya.

Ini cocok untuk pengembang web yang sering melakukan pengeditan pada file HTML atau XML. Sebab, ekstensi ini bisa mempercepat proses pengeditan dengan mengurangi kesalahan dan memastikan konsistensi dalam *tag*.

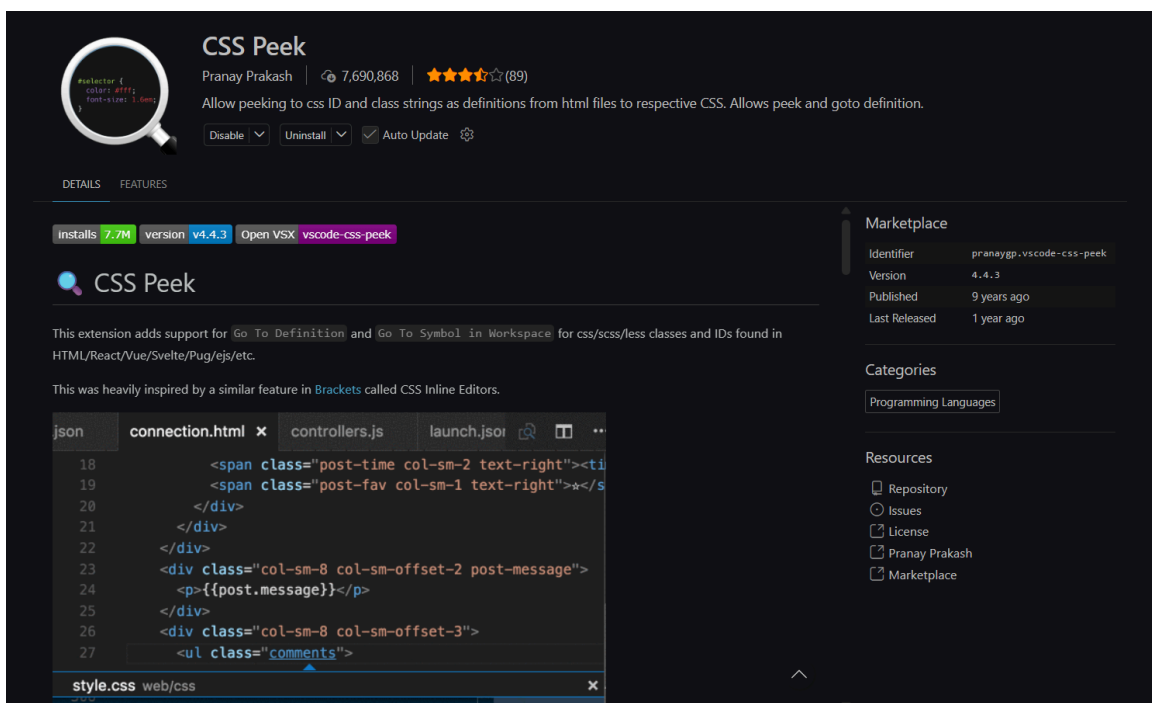
6. Path Intellisense



Selanjutnya adalah Path Intellisense. Ekstensi ini menambahkan *autocompletion* untuk jalur file dalam proyekmu. Hal ini memudahkan penautan file dan modul.

Ini sangat cocok bagi pengembang yang sering bekerja dengan file dan modul dalam proyek besar. Sebab, ini bisa mempercepat penulisan kode dengan menyediakan saran otomatis untuk jalur file dan modul.

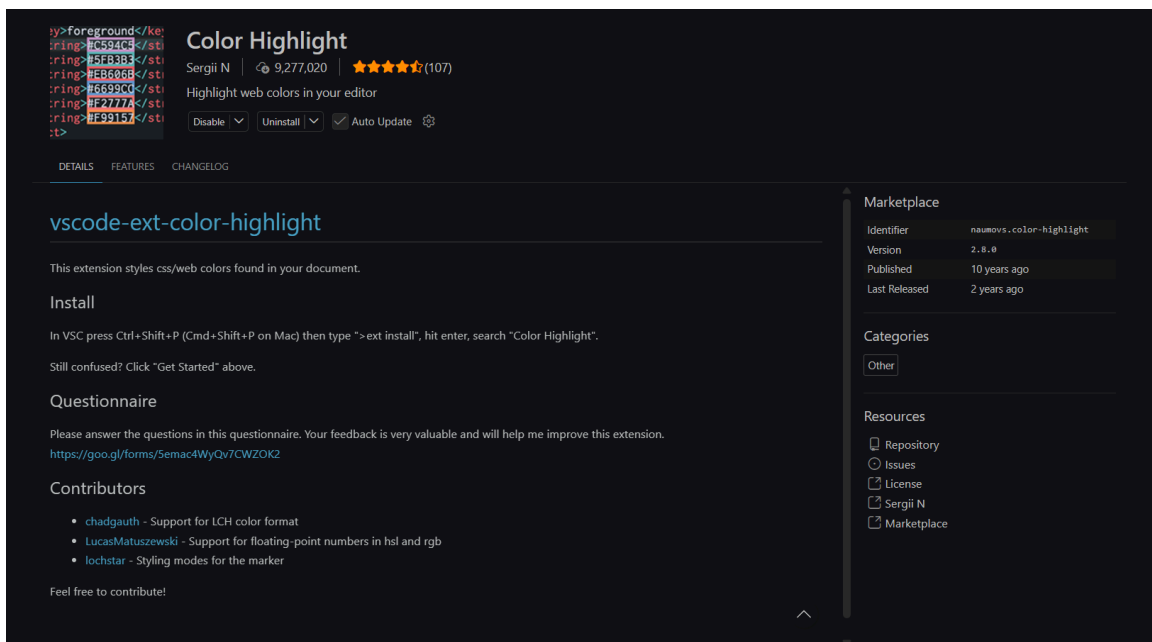
7. CSS Peek



VS Code *extension* selanjutnya adalah CSS Peek. Hal ini memungkinkanmu untuk melihat dan langsung melompat ke definisi CSS dari kelas, ID, atau tag HTML yang diacu dalam kode.

CSS Peek sangat cocok bagi pengembang yang sering harus beralih antara file HTML dan CSS. Hal ini karena CSS Peek bisa mempercepat pengembangan dan *debugging* dengan mengintegrasikan pengalaman penulisan HTML dan CSS.

8. Color Highlight



VS Code *extension* selanjutnya adalah menyoroti sintaks warna dalam kodemu dengan warna sebenarnya. Hal ini memudahkan pemilihan dan penyesuaian warna.

Color Highlight juga cocok bagi desainer dan pengembang yang bekerja dengan tata letak dan desain visual. Sebab, ekstensi ini mampu mempercepat pemilihan warna dan meningkatkan konsistensi visual di dalam kode.

9. Debugger for Chrome

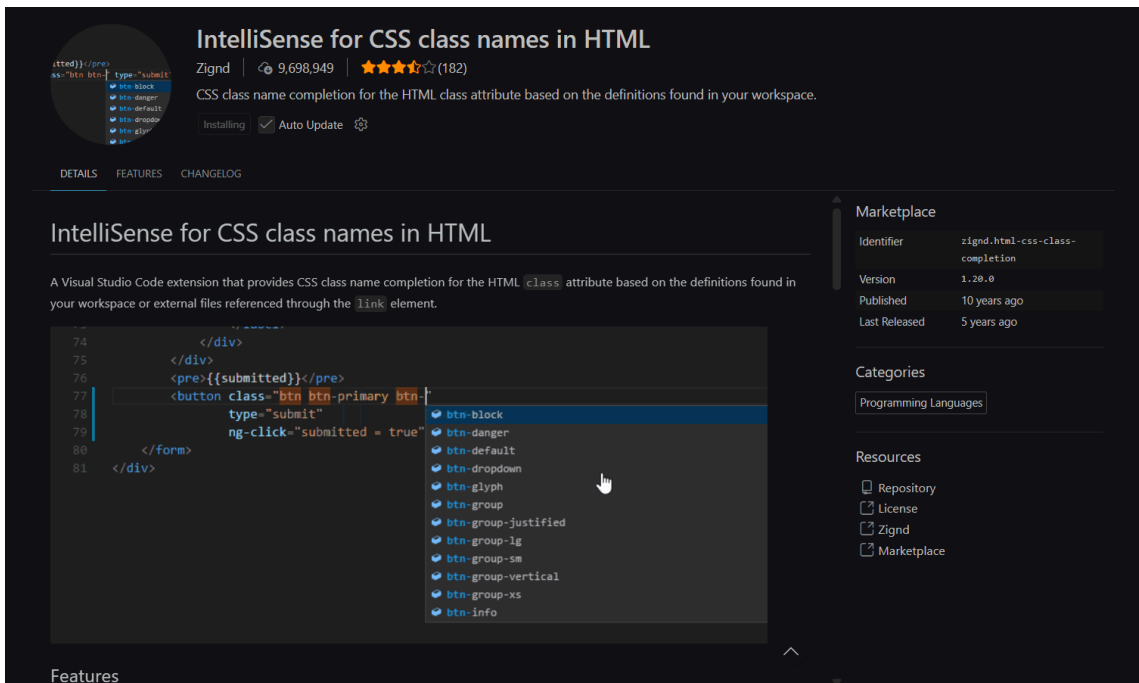


Berikutnya adalah Debugger for Chrome. Hal ini memungkinkan *debugging* kode JavaScript langsung dari VS Code dengan menggunakan Chrome sebagai lingkungan eksekusi.

Ini cocok digunakan oleh pengembang yang menggunakan Chrome untuk pengujian dan ingin integrasi *debugging* secara mulus.

Fungsi utama Debugger for Chrome adalah memperbaiki *bugs* secara efektif. Hal ini dilakukan dengan menggunakan fitur *breakpoint*, *step-through*, dan *inspection*.

10. IntelliSense for CSS class names in HTML



VS Code *extension* terakhir adalah IntelliSense for CSS *class names* in HTML. Ekstensi ini bisa menambahkan *autocompletion* untuk nama kelas CSS yang tersedia saat mengedit file HTML. Fungsi tersebut bisa meningkatkan kecepatan dan akurasi.

Lebih lanjut, ekstensi ini cocok bagi pengembang yang bekerja pada *stylesheet* kompleks dan ingin integrasi yang lebih baik dengan HTML.

Secara keseluruhan, IntelliSense for CSS bisa memudahkan pengelolaan dan pemakaian kelas CSS dengan menyediakan saran otomatis saat penulisan HTML.

Itulah 10 daftar VS Code *extension* yang bisa mempermudah proses pengembangan webmu. Masing-masing ekstensi ini dirancang untuk membuat proses pengembangan menjadi lebih efisien.

Web Browser

Digunakan untuk menampilkan hasil kode (seperti Chrome, Firefox, Safari, Edge, Opera, Maxthon). Tambahkan alat eksistensi pengembang seperti **Chrome Developer Tools** atau **Firebug** (pada Firefox) agar Anda bisa memeriksa elemen web dan melakukan perbaikan *error* sementara langsung dari layar.

Local Web Server

Agar dapat menjalankan bahasa server (PHP) dan database tanpa internet, gunakan *software* simulasi *localhost*. Aplikasi paket terpopuler adalah **XAMPP** atau WAMP Server. *Langkah Instalasi XAMPP*: Unduh aplikasi *installer* di apachefriends.org, jalankan `xampp-win32-x.x.x-installer.exe`, pilih lokasi folder `D:\xampp`, centang *Install Apache as service* dan *MySQL as service*, klik *Install* lalu *Finish*. Buka XAMPP Control Panel, klik tombol *Start* pada Apache dan MySQL hingga berstatus hijau. Terakhir, buka *browser* dan ketikkan `http://localhost`.

Laragon

Laragon telah lama menjadi pilihan favorit bagi para pengembang web yang membutuhkan lingkungan pengembangan lokal yang ringan dan fleksibel. Namun, perubahan kebijakan oleh pengembangnya, Leo Khoa, membuat banyak pengguna terkejut karena Laragon kini beralih menjadi berbayar, bertentangan dengan slogan awal "Free Forever".

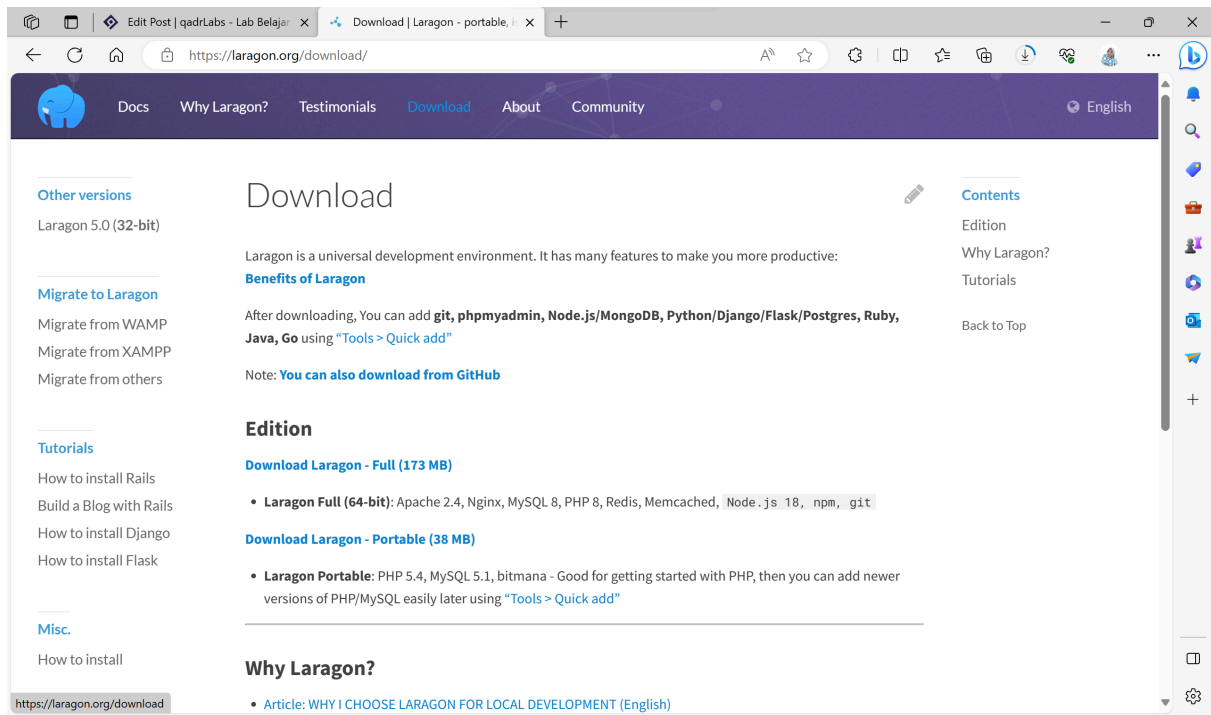
Perubahan Lisensi Laragon

Ketika pertama kali dirilis, Laragon dipromosikan sebagai alat yang gratis untuk digunakan selamanya. Platform ini dirancang untuk memberikan solusi pengembangan lokal dengan dukungan PHP, MySQL, Node.js, dan berbagai teknologi lainnya. Namun, pada akhir 2024, Leo Khoa mengumumkan perubahan besar: mulai versi 7, Laragon akan berbayar.

Keputusan ini diambil karena faktor ekonomi dan tantangan yang dihadapi pengembang. Dalam sebuah diskusi di GitHub, Leo mengungkapkan bahwa ia harus mencari cara untuk menopang kehidupannya, sehingga Laragon pun beralih menjadi proyek berbayar.

Download Laragon

Untuk lingkungan server lokal, Laragon adalah pilihan yang sempurna karena ringan dan mudah digunakan. Anda bisa mengunduhnya dari website resmi Laragon. Klik menu **Download** dan pilih versi **Laragon - Full** dari daftar pilihan.



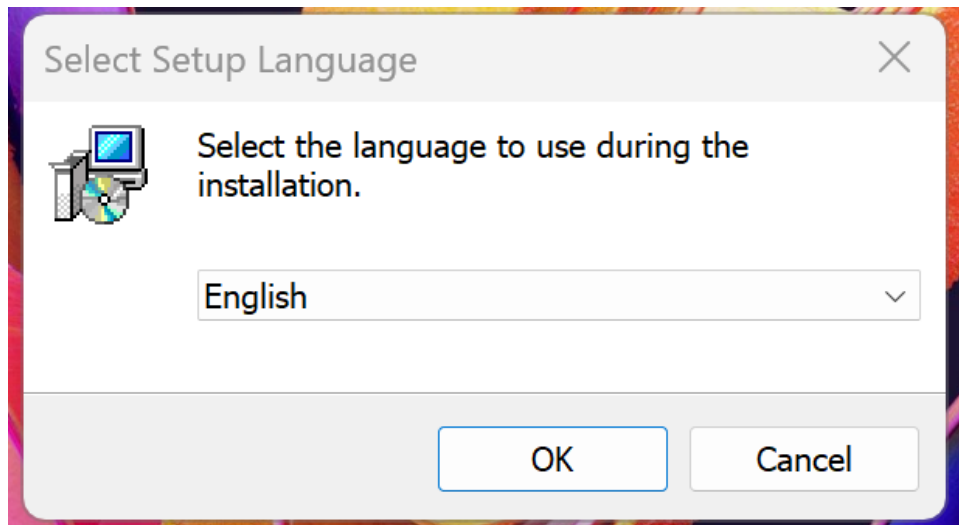
Keterangan:

- Setelah laragon versi 7 rilis, file yang didownload di halaman download laragon adalah laragon versi 7. Berdasarkan diskusi di repositori laragon, laragon versi 7 **tidak lagi gratis** dan menggunakan **Paid Licensing model**.
- Apabila ingin menggunakan **laragon versi gratis**, teman-teman bisa download langsung di link github, yaitu <https://github.com/leokhoa/laragon/releases/download/6.0.0/laragon-wamp.exe>

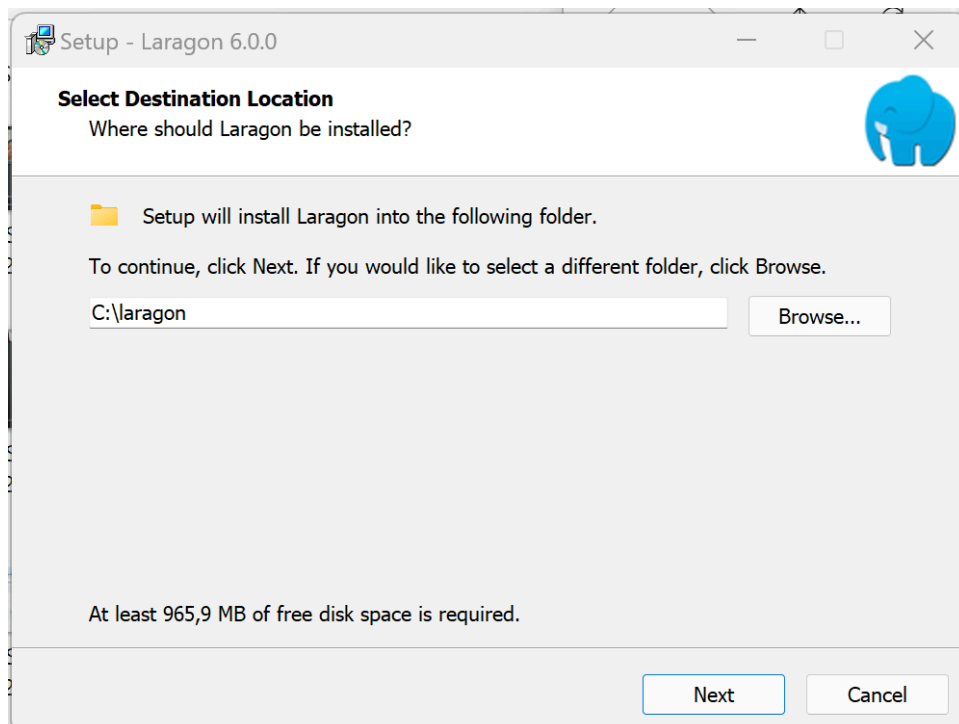
Proses Instalasi Laragon

Setelah file `laragon-wamp.exe` selesai diunduh, klik dua kali untuk memulai instalasi. Berikut adalah langkah-langkah instalasi Laragon:

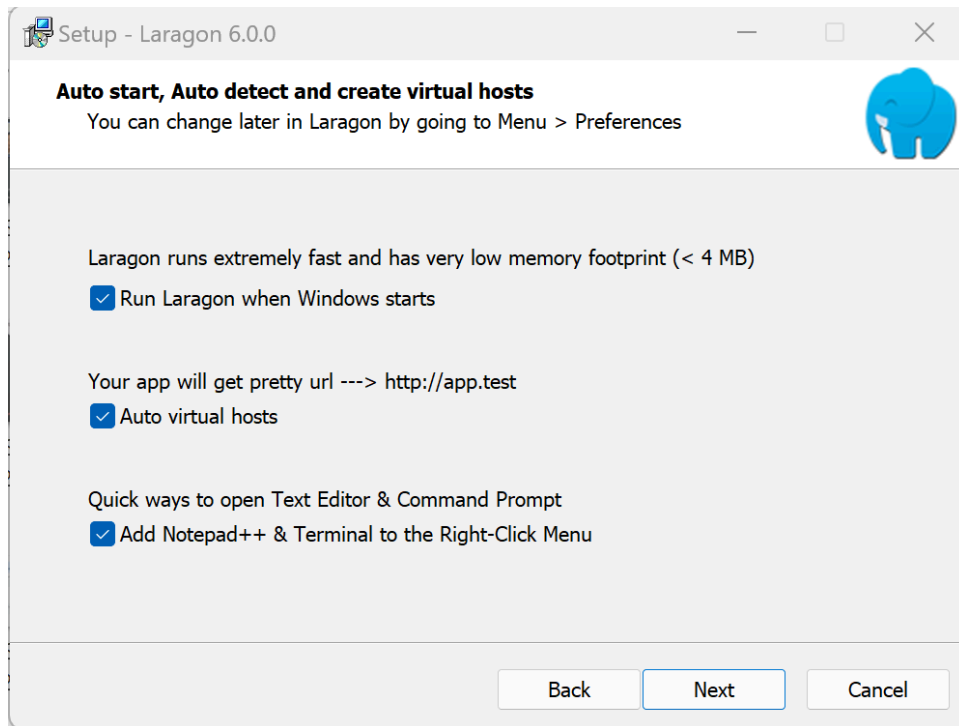
1. Pilih bahasa instalasi (misalnya, **English**), lalu klik **Next**.



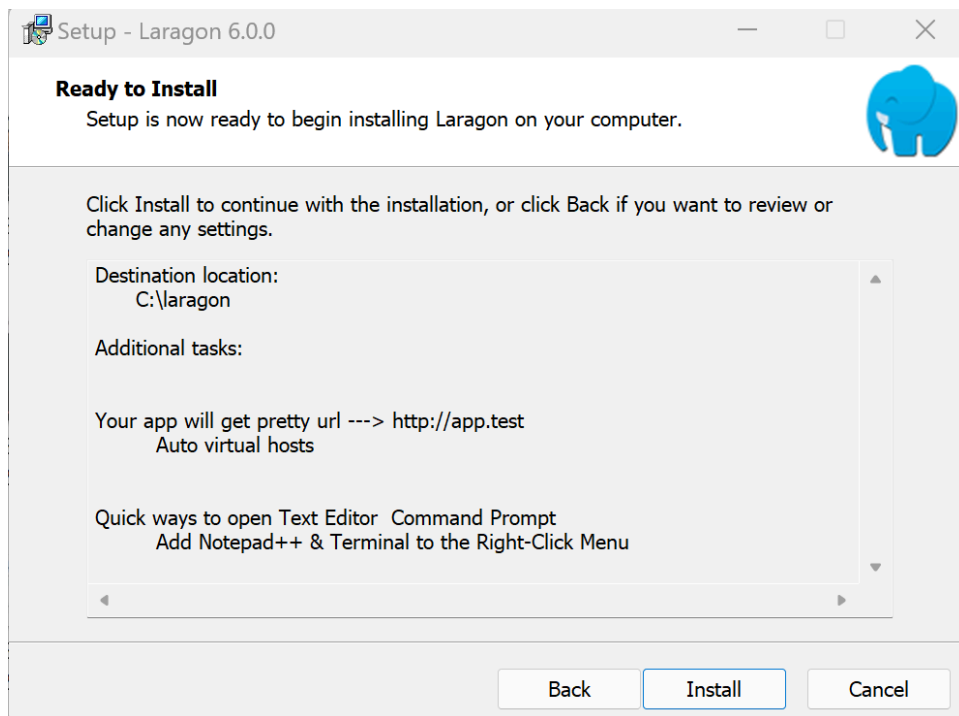
2. Pilih lokasi instalasi Laragon, default-nya adalah `C:\Laragon` . Klik **Next** untuk melanjutkan.



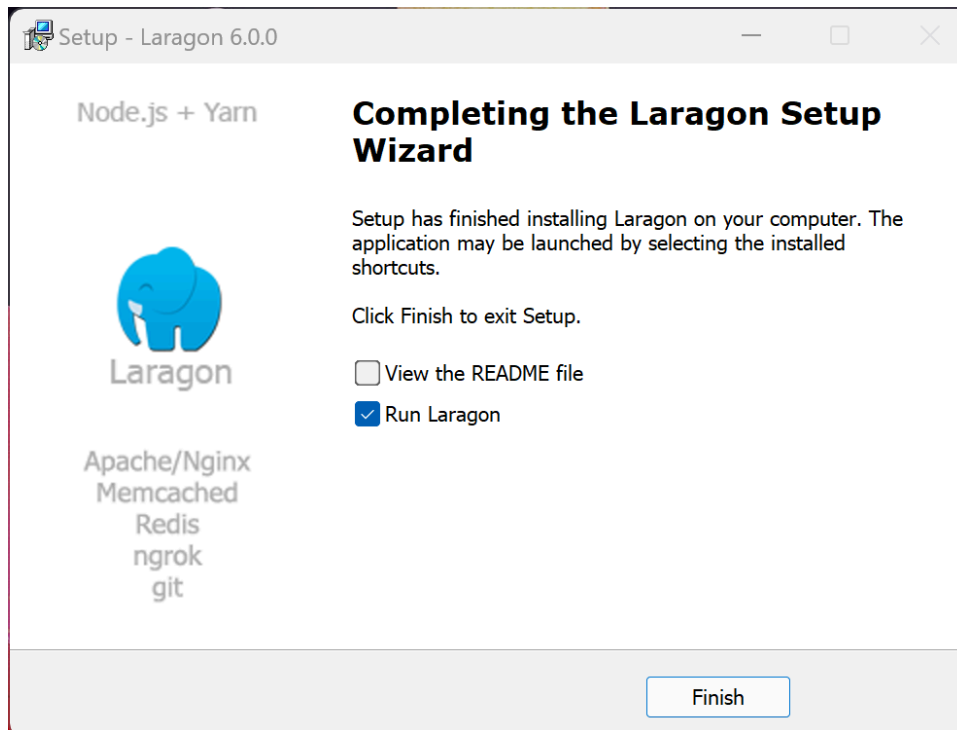
3. Anda akan melihat opsi konfigurasi seperti autostart pada saat Windows mulai, dan menambahkan Notepad++ serta terminal ke Laragon. Anda bisa memilih opsi sesuai preferensi, lalu klik **Next**.



4. Pada halaman **Ready to Install**, klik **Install** untuk memulai proses instalasi Laragon.

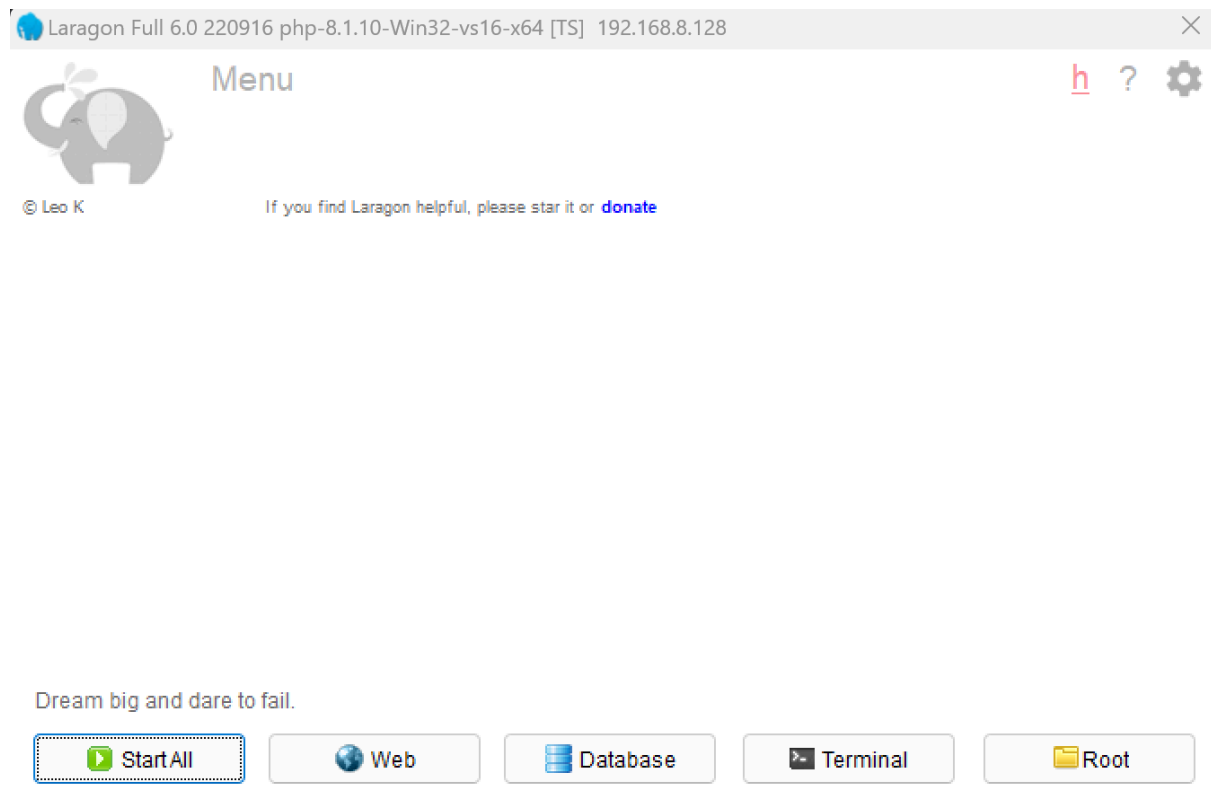


5. Tunggu hingga proses instalasi selesai. Setelah itu, klik **Finish** untuk menutup installer dan membuka Laragon.



Menjalankan Laragon

Setelah Laragon terbuka, Anda akan melihat tampilan antarmuka Laragon yang intuitif dan user-friendly.



Untuk memulai layanan seperti Apache dan MySQL, Anda cukup klik **Start All**. Laragon akan menjalankan semua layanan yang diperlukan untuk pengembangan aplikasi web, termasuk Apache, MySQL, dan PHP.

Membuka folder project di visual studio code

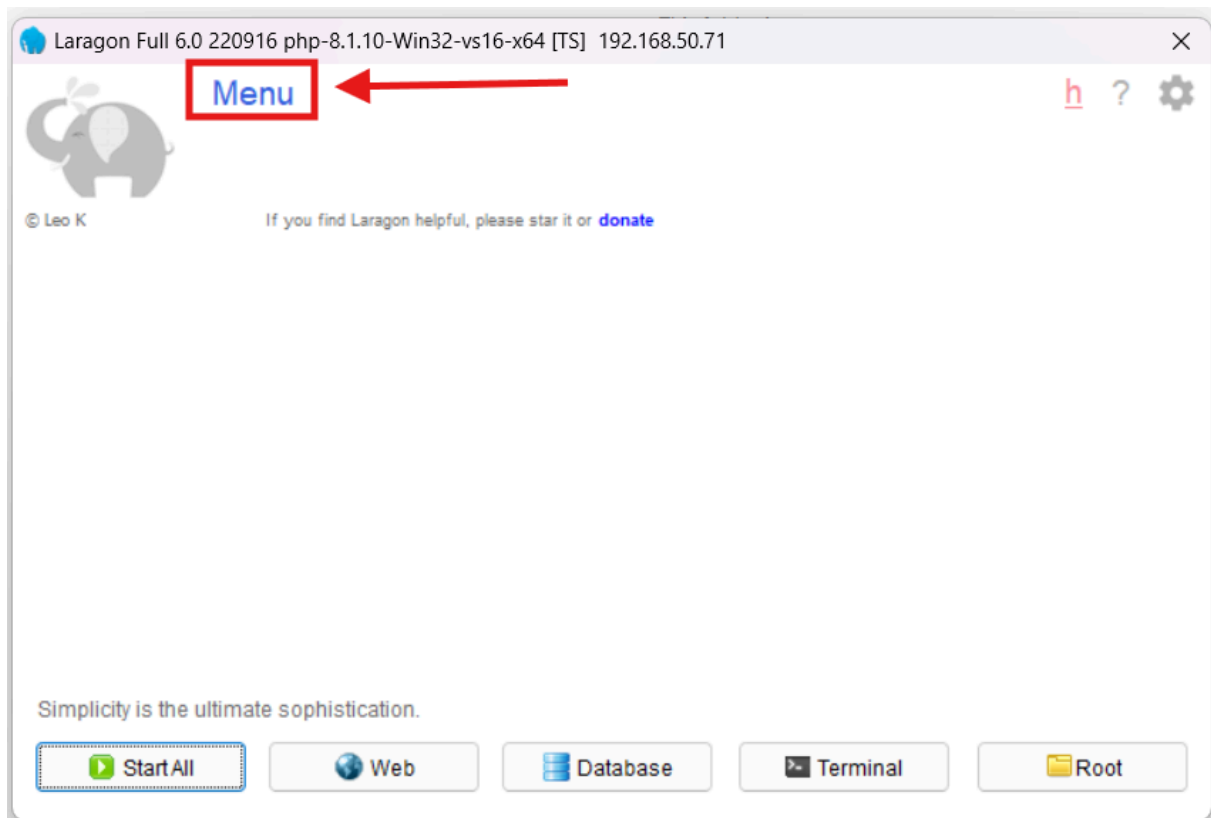
Setelah menginstal VS Code dan Laragon, langkah selanjutnya adalah menghubungkan keduanya agar Anda bisa bekerja dengan smooth dalam satu environment development. Anda bisa membuka direktori project di Visual Studio Code dengan cara sederhana:

1. Buka Laragon, klik **Root**. Ini akan membuka direktori `root` di mana proyek Anda berada, yaitu `C:\laragon\www`. Sebagai contoh di sini kita akan buat direktori project baru, di mana pada real project nanti kita buat langsung menggunakan command `composer` apabila kita mengembangkan project `laravel`, `codeigniter` ataupun framework php lainnya. Di direktori `root`, kita bisa buat direktori project baru dengan nama `sample-app`.
2. Buka Visual Studio Code, klik menu **File > Open Folder** dan pilih folder yang baru saja Anda buat di direktori `root` Laragon, yaitu `C:\laragon\www\sample-app`.

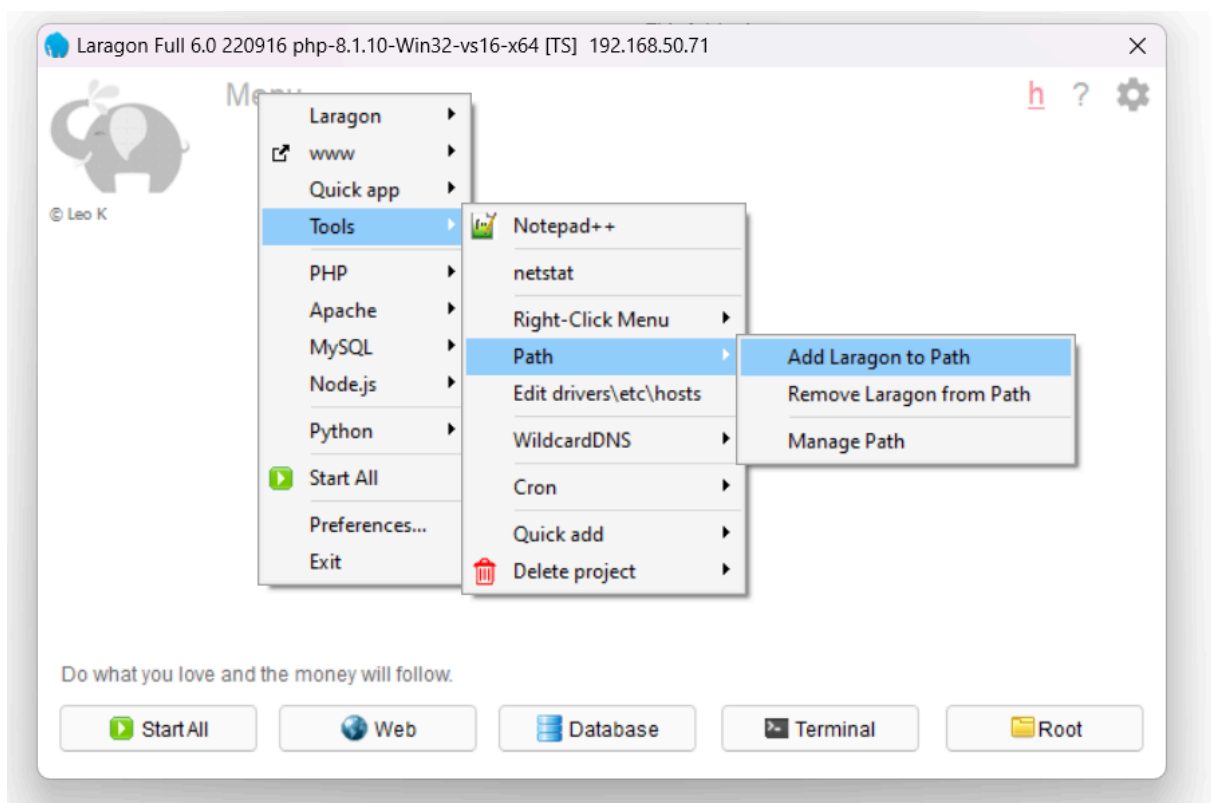
Konfigurasi Awal

Pada tahapan ini kita akan setup environment variable supaya kita bisa run command seperti `php`, `node`, dan `composer`, di mana command tersebut dapat kita gunakan ketika kita melakukan development project.

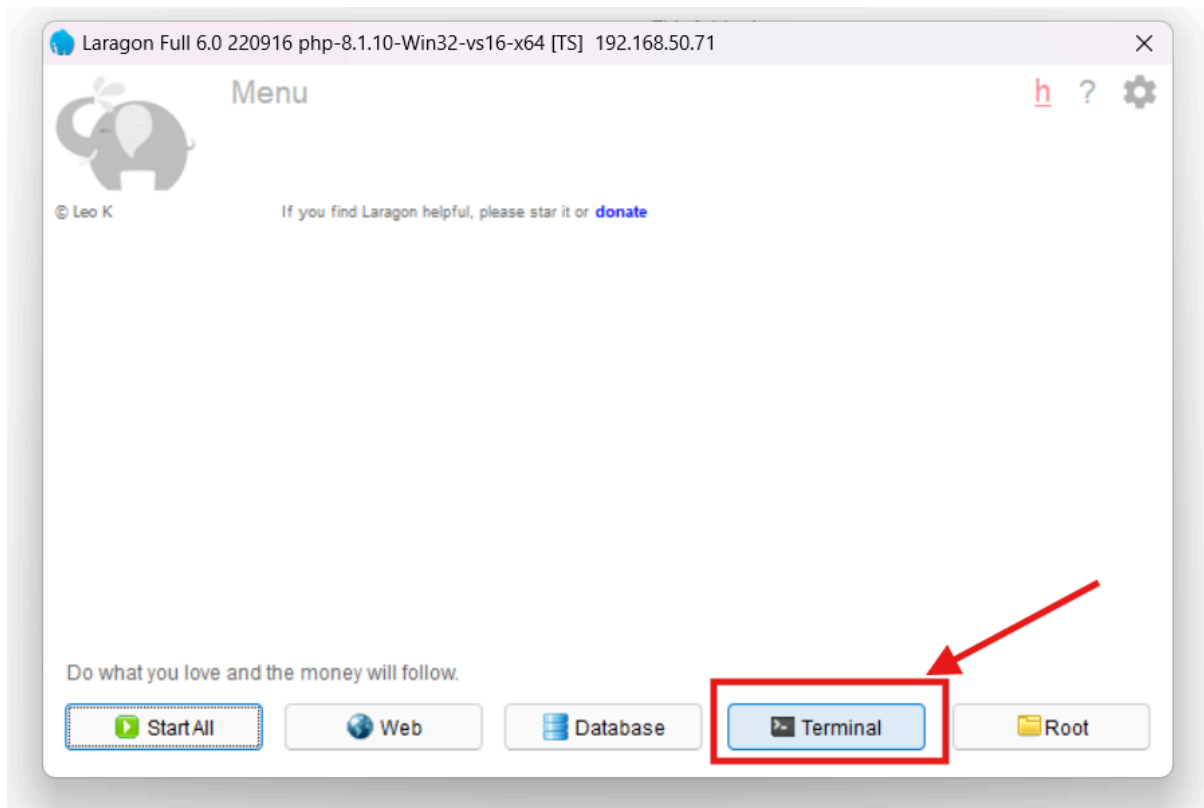
Pertama kita buka kembali laragon, lalu kita klik `menu` untuk membuka menu laragon.



Lalu selanjutnya kita tambahkan `path` laragon dengan klik menu `Tools > Path > Add Laragon to Path` .



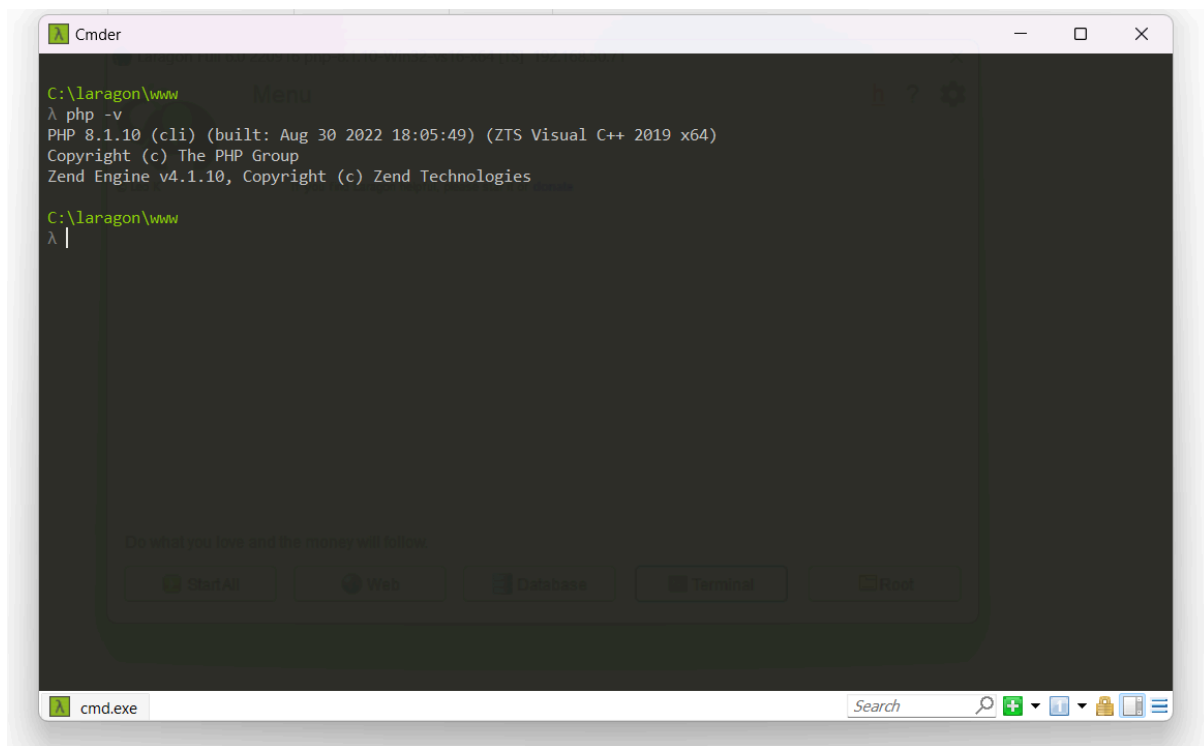
Sekarang kita coba run command untuk mengecek apakah laragon berhasil ditambahkan ke dalam path. Untuk mengecek apakah sudah berhasil, kita buka `terminal` atau `command prompt` dengan menekan menu `Terminal` di inteface laragon.



Setelah `terminal` terbuka, kita run command berikut ini untuk mengecek versi php yang terinstall di laragon.

```
php -v
```

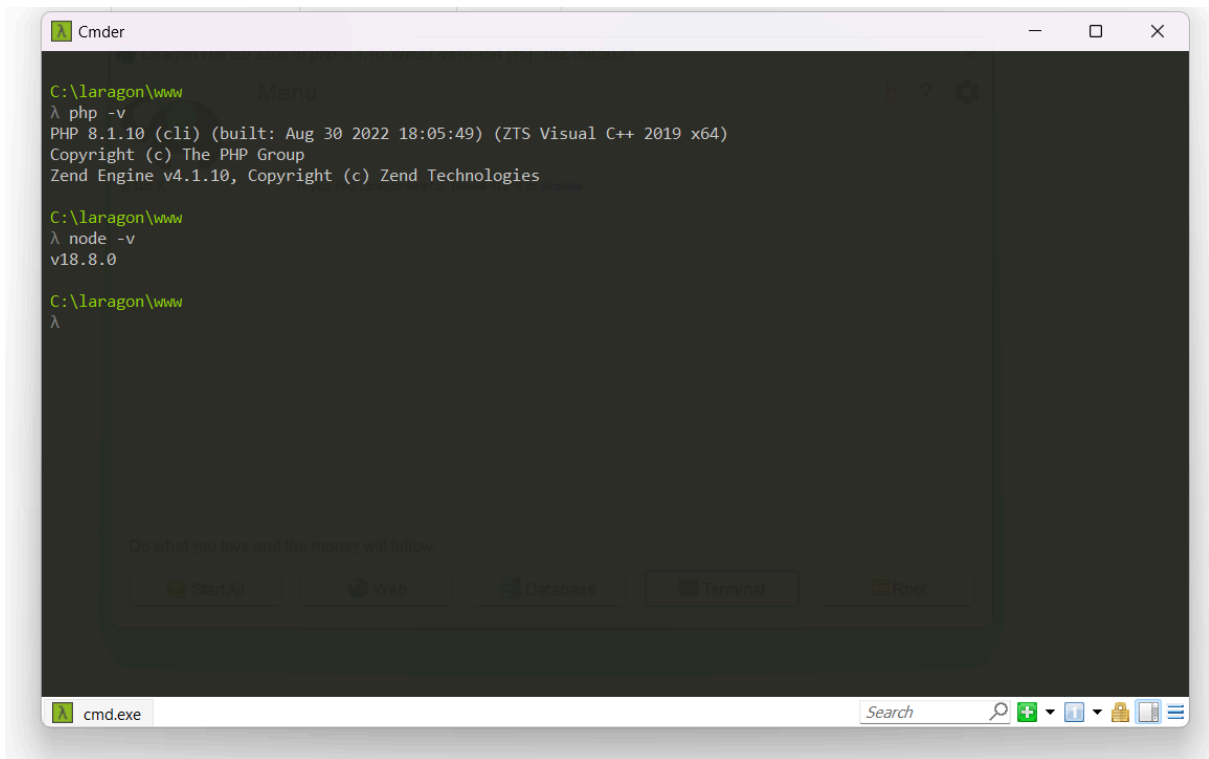
Selanjutnya kita bisa lihat versi php pada output yang ditampilkan seperti pada gambar berikut ini.



Seperti yang kita lihat pada gambar di atas, versi php yang terinstall adalah PHP versi 8.1

Selanjutnya kita coba cek versi node js yang terinstall, kita buka kembali terminal dan run command berikut ini.

```
node -v
```



```
C:\laragon\www
λ php -v
PHP 8.1.10 (cli) (built: Aug 30 2022 18:05:49) (ZTS Visual C++ 2019 x64)
Copyright (c) The PHP Group
Zend Engine v4.1.10, Copyright (c) Zend Technologies

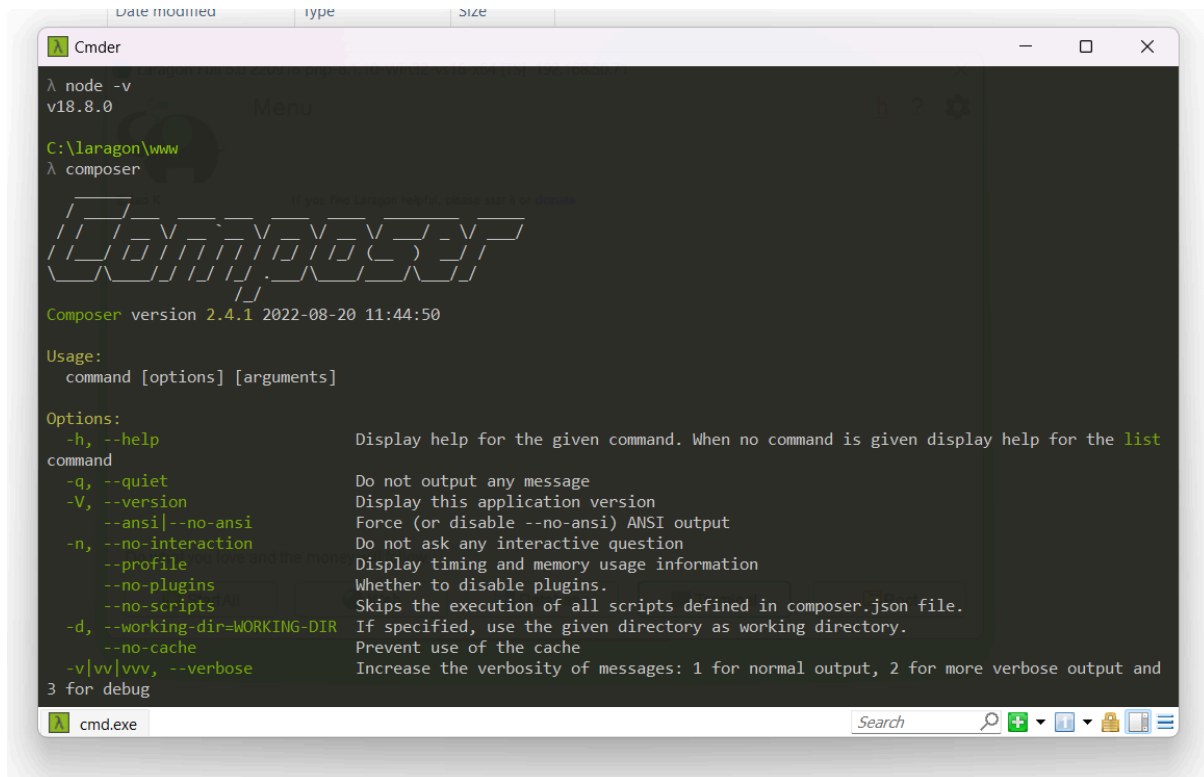
C:\laragon\www
λ node -v
v18.8.0

C:\laragon\www
λ
```

Seperti yang terlihat pada output yang ditampilkan di terminal, versi node js yang terinstall pada laragon adalah versi 18.8.0.

Dan yang terakhir kita cek apakah kita bisa run `composer`. Buka kembali terminal lalu run command berikut ini.

```
composer
```



```
λ node -v
v18.8.0

C:\laragon\www
λ composer

Composer version 2.4.1 2022-08-20 11:44:50

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display help for the given command. When no command is given display help for the list
  command
  -q, --quiet               Do not output any message
  -V, --version             Display this application version
  --ansi|--no-ansi         Force (or disable --no-ansi) ANSI output
  -n, --no-interaction     Do not ask any interactive question
  --profile                Display timing and memory usage information
  --no-plugins             Whether to disable plugins.
  --no-scripts             Skips the execution of all scripts defined in composer.json file.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache               Prevent use of the cache
  -v|vv|vvv, --verbose     Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and
  3 for debug
```

Seperti yang terlihat pada gambar di atas, kita bisa lihat output dari command composer.

Alat Tingkat Lanjut Lainnya:

- **Version Control System (Git):** Sistem perekam jejak untuk mengelola semua riwayat perubahan kode secara berkala, sangat esensial untuk kolaborasi tim kerja agar kode tidak tertimpa.
- **Command Line Interface (CLI):** Layar antarmuka berbasis teks hitam yang jauh lebih cepat memproses beban perintah dibandingkan antarmuka grafis (GUI) biasa.
- **Package Manager:** Alat otomatis seperti NPM, Composer, atau PIP untuk mengunduh pustaka (library) dan *software* dari pihak ketiga agar *developer* tidak perlu meracik fitur baru dari nol.
- **Aplikasi Converter & Desain Grafis:** Contohnya Format Factory (untuk mengubah format video agar mendukung *browser*), Adobe Photoshop, atau Dreamweaver.

Pengenalan Git

Git adalah software yang digunakan sebagai **sistem pengontrol versi** atau **Version Control System** dalam sebuah proyek perangkat lunak atau website.

Git dikembangkan oleh pembuat linux yaitu Linus Torvalds pada tahun 2005.

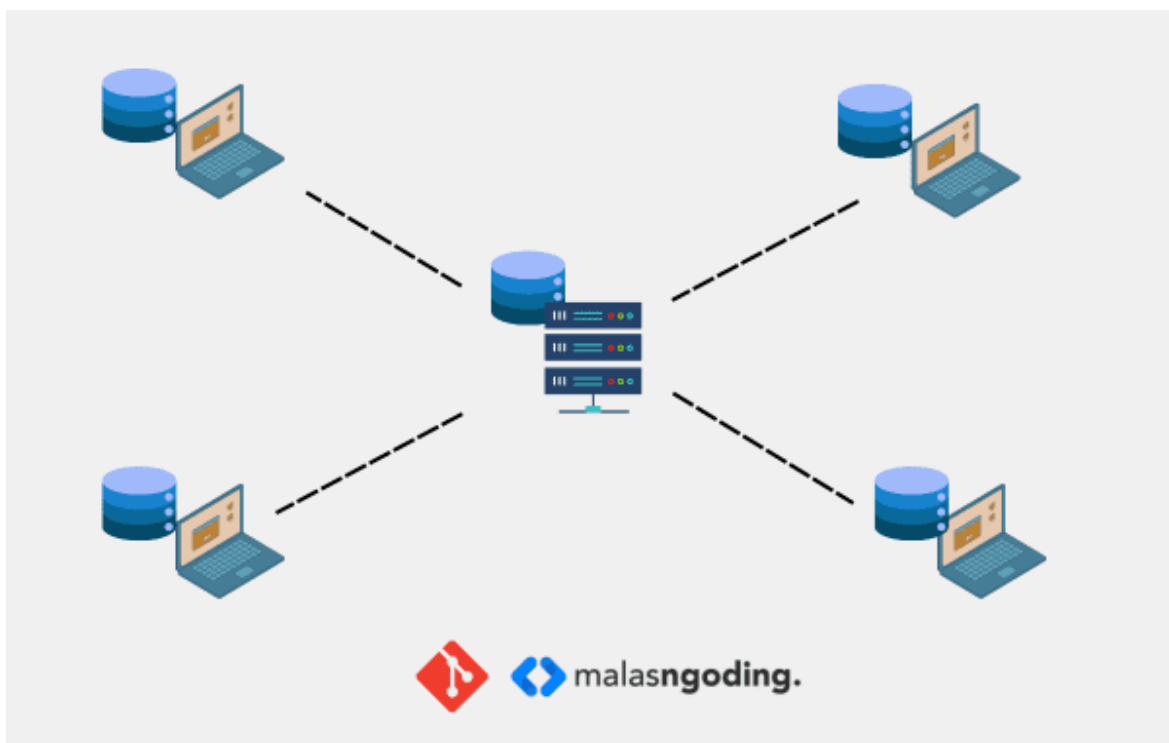
Git dirancang untuk memudahkan kolaborasi antar programmer dan memantau setiap riwayat perubahan pada suatu proyek.

Git juga dikenal dengan istilah **distributed revision control**. atau kontrol versi terdistribusi.

Nah apa lagi ini? apa maksudnya?

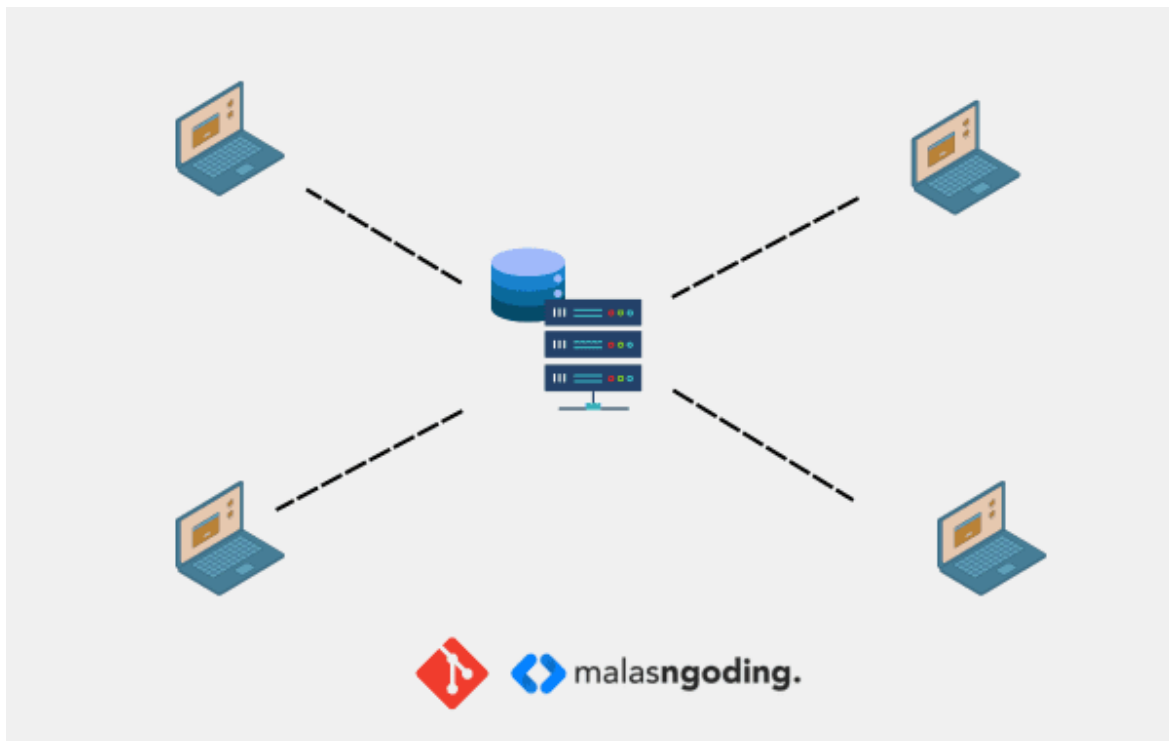
Artinya penyimpanan data yang dibuat Git tidak hanya tersimpan pada satu tempat. Melainkan pada setiap orang yang terhubung dalam sebuah proyek Git tersebut.

Sistem pengontrol versi terdistribusi



Berbeda dengan **VCS** versi lama, yaitu "**sistem pengontrol versi terpusat**". yang menyimpan semua *database* perubahan hanya pada 1 server.

Sistem pengontrol versi terpusat



Kekurangan dari sistem pengontrol versi terpusat adalah:

- Pengguna harus terkoneksi dengan jaringan internet agar dapat mengakses source code proyek.
- Lumayan beresiko karena semua database berada dalam 1 server. jika server tersebut bermasalah, maka pengguna akan kesulitan dalam mengakses data proyek.

Sehingga Git dibuat untuk menutupi kekurangan dari sistem pengontrol versi terpusat.

Bisa dikatakan Git merupakan sistem pengontrol versi terbaru saat ini.

Kegunaan Git

Sebagai Version Control System

Dari kata *version control system* bisa kita artikan bahwa dengan Git kita bisa mengatur versi-versi dari source code aplikasi atau website yang sedang kita bangun.

Secara sederhana, Git mencatat setiap perubahan dalam proyek perangkat lunak. baik itu saat membuat penambahan fitur, perbaikan bug atau membuat pengujian pada aplikasi atau website yang sedang teman-teman buat.

Mempermudah Team Work

Bukan hanya sebagai pengontrol versi, kita juga memiliki kegunaan lain yang bisa membantu programmer yang berkerja secara tim untuk menggabungkan coding mereka.

Misal kita membuat fitur pencarian pada website. dan si B membuat fitur CRUD pegawai, dan si C mendesain halaman depan websitenya.

Kita dan tim bisa menggabungkan setiap perubahan dengan melakukan **commit** pada **repository** proyek website kita.

Tentu saja akan sangat membantu programmer, dibandingkan harus menggabungkan coding-codingnya secara manual.

Memakan waktu yang lama pastinya.

Contoh Kegunaan Git

Saat mengerjakan sebuah pembuatan website, Git bisa menyimpan setiap riwayat perubahan pada source code yang teman-teman buat.

Misal teman-teman membuat fitur baru, katakanlah fitur pencarian artikel. nah, Git bisa menyimpan proyek website tersebut menjadi versi 1 dan versi 2.

Versi 1 yaitu proyek website utama. dan versi 2 adalah proyek website yang menggunakan fitur pencarian.

Sehingga pada saat teman-teman membutuhkan source code proyek website versi 1 yang tidak memiliki fitur pencarian, teman-teman masih bisa mengaksesnya kembali tanpa takut mengganggu proyek website versi lainnya.

Jadi teman-teman tidak perlu lagi memisah folder proyek website tersebut secara manual seperti cara lama yaitu dengan mem-backup file-file tersebut.

Contoh Kegunaan Git

| Name | Date modified |
|--------------------------------------|---------------|
| web_malasngodingx | 23/01/2023 |
| web_malasngoding5 | 23/01/2023 |
| web_malasngoding4 | 23/01/2023 |
| web_malasngoding3 | 23/01/2023 |
| web_malasngoding2 | 23/01/2023 |
| web_malasngoding_terbaru | 23/01/2023 |
| web_malasngoding_perubahan_template3 | 23/01/2023 |
| web_malasngoding_perubahan_template2 | 23/01/2023 |
| web_malasngoding_perubahan_template1 | 23/01/2023 |
| web_malasngoding_oke2 | 23/01/2023 |
| web_malasngoding_oke | 23/01/2023 |
| web_malasngoding_forum | 23/01/2023 |
| web_malasngoding_asli | 23/01/2023 |
| web_malasngoding tanpa login | 23/01/2023 |
| web_malasngoding | 23/01/2023 |

Tapi proyek website teman-teman tetap berada dalam 1 folder. tapi sudah di pisah setiap riwayat perubahannya ke dalam versi-versi yang masih bisa diakses kembali kapan saja.

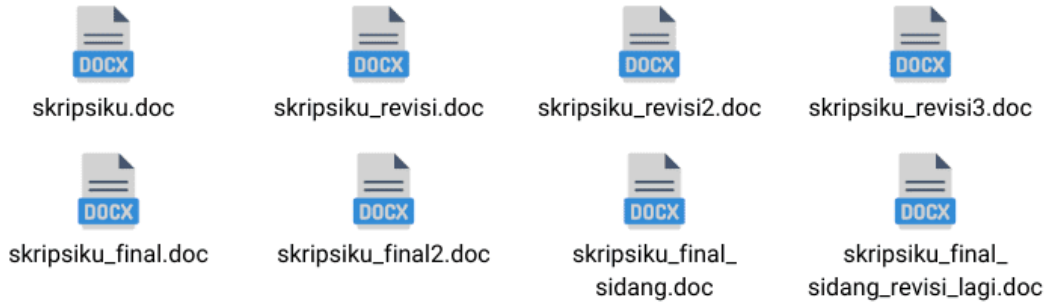
Contoh Penggunaan Git Pada File Skripsi

File skripsi bisa kita jadikan sebagai contoh, karena biasanya banyak revisi dan versi perubahan dari file tersebut.

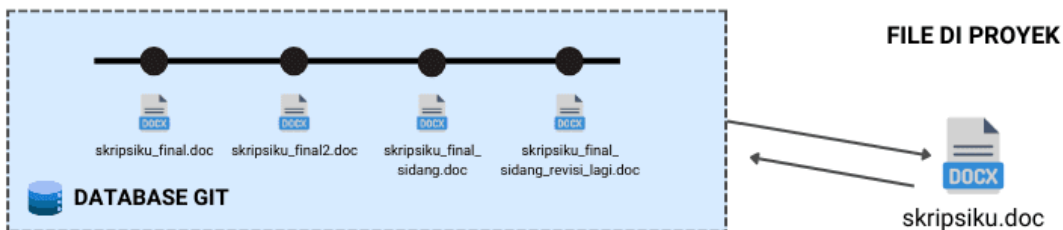
Perhatikan gambar ilustrasi penggunaan Git untuk manajemen file skripsiku.doc.

Belajar Pengertian dan Contoh Penggunaan Git

Biasanya



Menggunakan Git



Versi revisi bisa diakses kapan saja dan yang mana saja



Setelah menggunakan Git, revisi file yang sebelumnya berantakan menjadi lebih rapi. karena masih dalam bentuk 1 file.

Walaupun yang terlihat hanya 1 file saja, tapi kita bisa mengakses file-file revisi yang sebelumnya telah tersimpan dalam database git.

Dari contoh-contoh yang sudah dijelaskan di atas mudah-mudahan bisa memberi gambaran yang jelas untuk teman-teman. hehe ..

Mengenal istilah dalam Git

Ada beberapa istilah dan perintah di Git yang harus teman-teman pahami.

Berikut adalah istilah-istilah dasar Git yang wajib diketahui karena akan kita bahas pada tutorial selanjutnya dalam seri belajar Git di malasngoding.com.

Repository

Repository adalah direktori lokasi penyimpanan sebuah proyek yang menggunakan Git. cara gampangannya, anggap saja maksud dari repository ini sebagai folder proyek website kita.

Commit

Commit bisa kita anggap sebagai perintah untuk menyimpan setiap perubahan yang kita buat pada repository atau proyek website kita.

Saat membuat commit, kita juga bisa memberi pesan sebagai informasi "Perubahan apa saja yang kita buat pada proyek website".

Sehingga akan memudahkan kita saat kita ingin mengakses kembali riwayat revisi proyek website kita.

Branch

Branch atau cabang bisa kita anggap sebagai salinan yang kita buat bercabang. misal ada revisi a, revisi b, dan revisi c. kita bisa membuat salinan baru pada revisi a tanpa mengganggu salinan revisi c.

Merge

Merge adalah proses menggabungkan perubahan dari satu branch ke branch lain. Misalnya, menggabungkan perubahan dari branch tertentu ke branch utama.

Fork

Fork artinya kita salinan sendiri dari repository orang lain. biasanya digunakan untuk berkontribusi pada proyek open source.

Clone

Clone artinya menyalin repository yang ada ke komputer kita. salinan yang dilakukan lengkap yaitu dari repository, branch (cabang) sampai riwayat commit dari repository yang di-clone tersebut.

Remote

Pemberian nama untuk repository yang berada di lokasi lain, baik itu di server github ataupun komputer lokal

Push

Proses mengirim 'perubahan' pada proyek website ke repository proyek website. jadi perubahan yang kita buat pada komputer lokal akan tersimpan ke repository proyek kita.

Pull

Proses penarikan 'perubahan' yang ada di proyek website ke komputer kita.

Maksudnya perubahan yang sudah di commit dan push oleh tim, bisa kita pull ke komputer kita.

Conflict (Konflik)

Conflict pada Git adalah masalah yang terjadi ketika ada 2 atau lebih perubahan pada file atau baris yang sama pada source code proyek website.

Biasanya Git tidak bisa menggabungkan perubahan tersebut secara otomatis. jadi kita perlu menyesuaikannya secara manual.

Dan masih akan ada beberapa istilah lagi yang akan teman-teman temui dalam seri tutorial belajar Git di malasngoding.com selanjutnya.

Kesimpulan

Git adalah perangkat lunak yang memiliki kemampuan sebagai sistem pengontrol versi (Version Control System) dalam sebuah proyek.

Git bisa membantu kita dalam memantau riwayat perubahan yang kita buat dalam proses membangun sebuah aplikasi atau website. dan kita bisa dengan bebas mengakses kembali riwayat revisi tersebut.

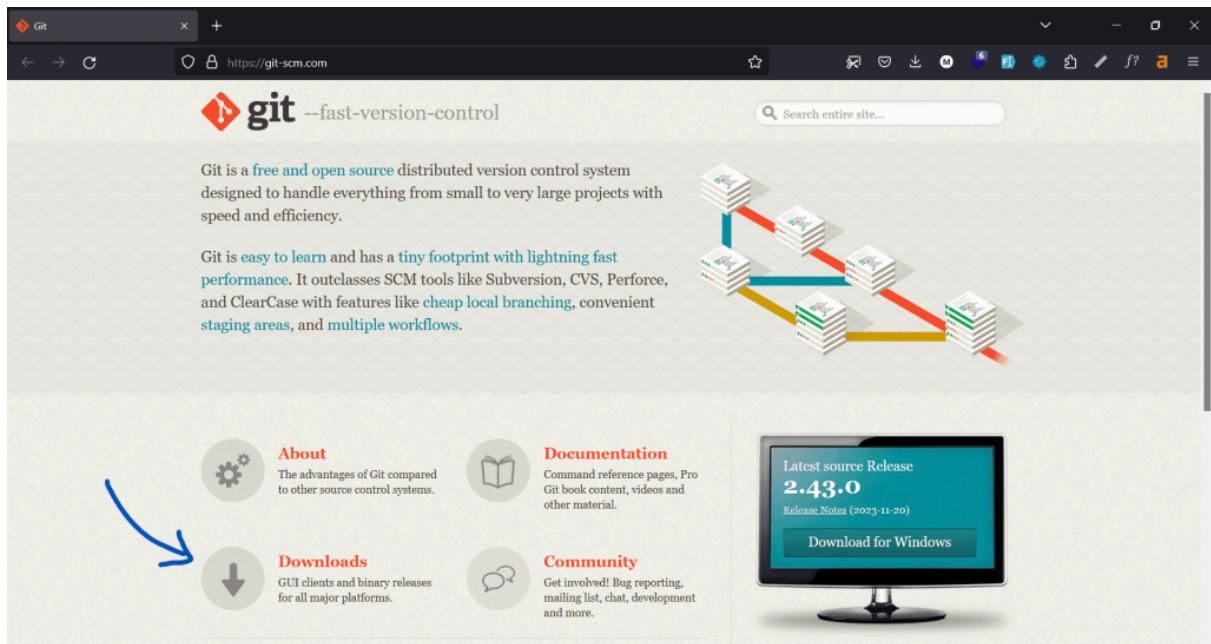
Git juga bisa mempermudah programmer mengerjakan pembuatan aplikasi atau website secara bersama-sama, karena Git memiliki kemampuan untuk menggabungkan setiap perubahan yang dibuat oleh setiap programmer dengan programmer lain.

Install Dan Konfigurasi Git

Seperti cara menginstall aplikasi di Windows pada umumnya, kita harus *download* dulu aplikasinya, kemudian klik *next* sampai selesai.

Silahkan **download Git** pada website resminya. yaitu di git-scm.com.

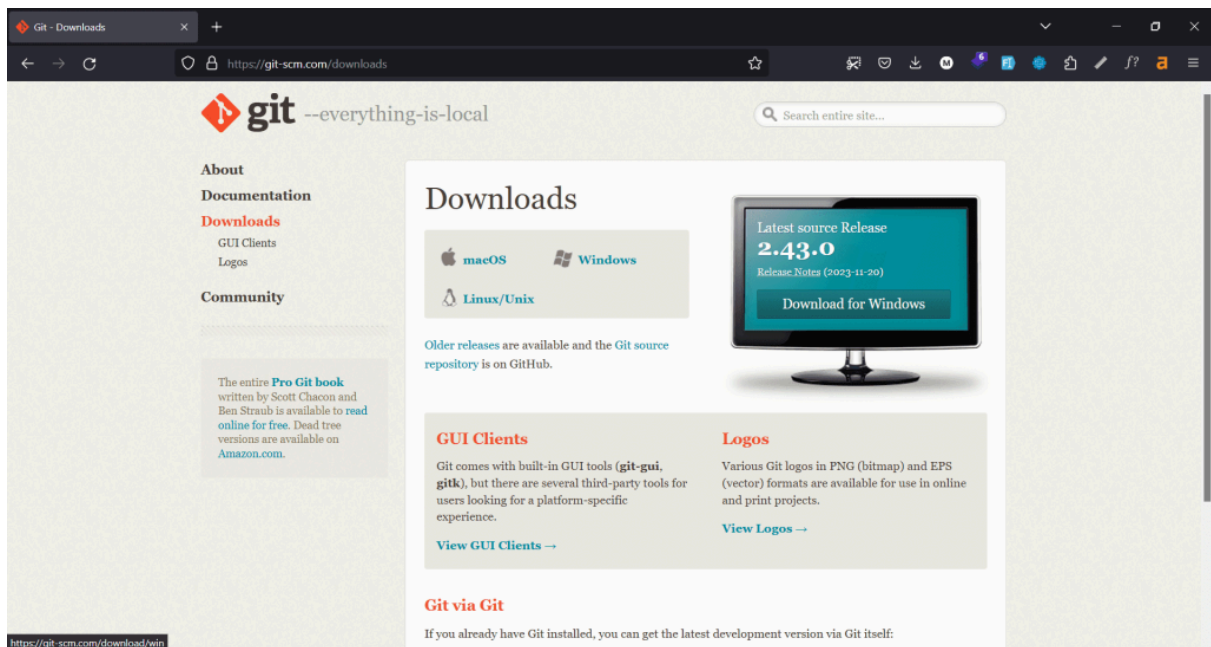
Cara install Git di windows



Gambar diatas adalah tampilan dari website resmi Git.

Klik pada "Download For Windows" atau pilih Menu "Downloads".

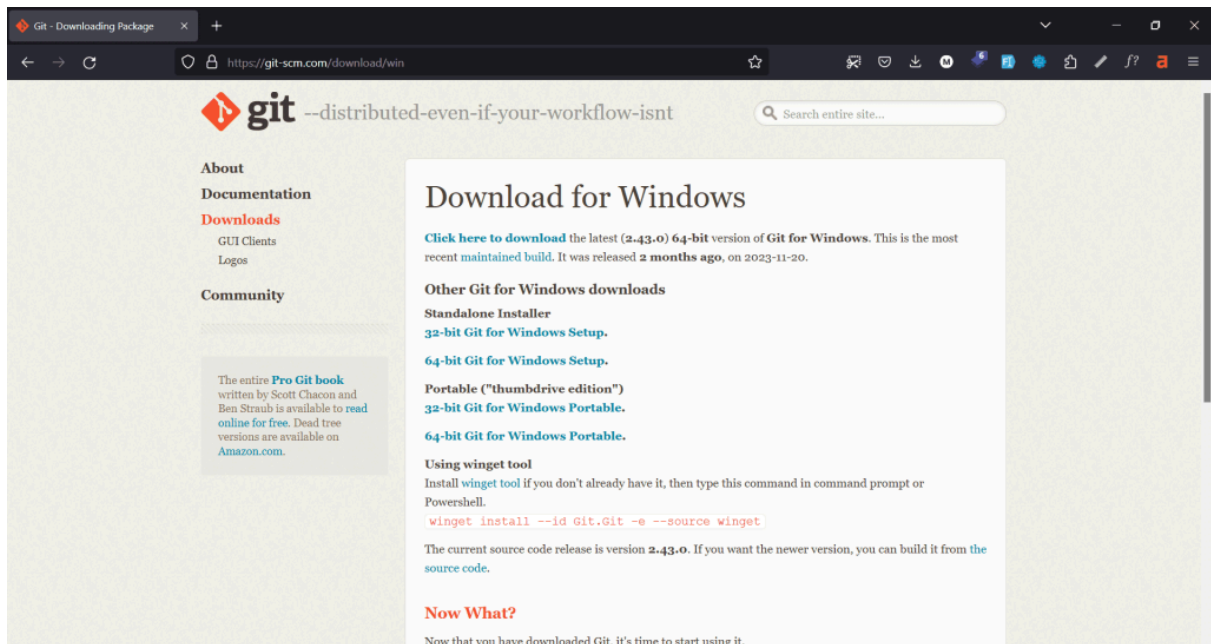
Download Git untuk Windows



Jika teman-teman klik pada menu "Downloads" maka dialihkan ke halaman pilihan os.

Pilihan OS nya, bisa pilih Windows atau klik tombol "Download for Windows".

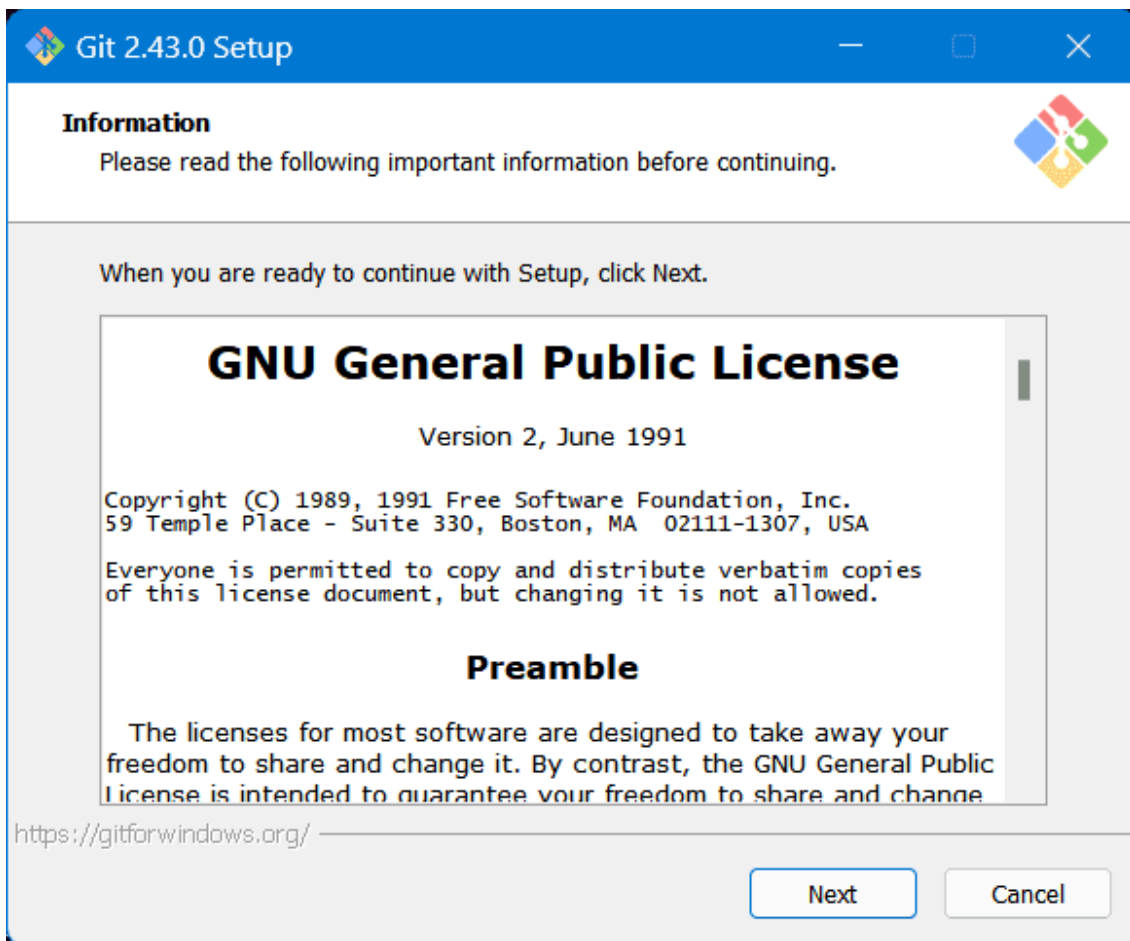
Pilih git sesuai komputer masing-masing



Langkah terakhir, klik pada salah satu pilihan. tergantung spek komputer teman-teman.

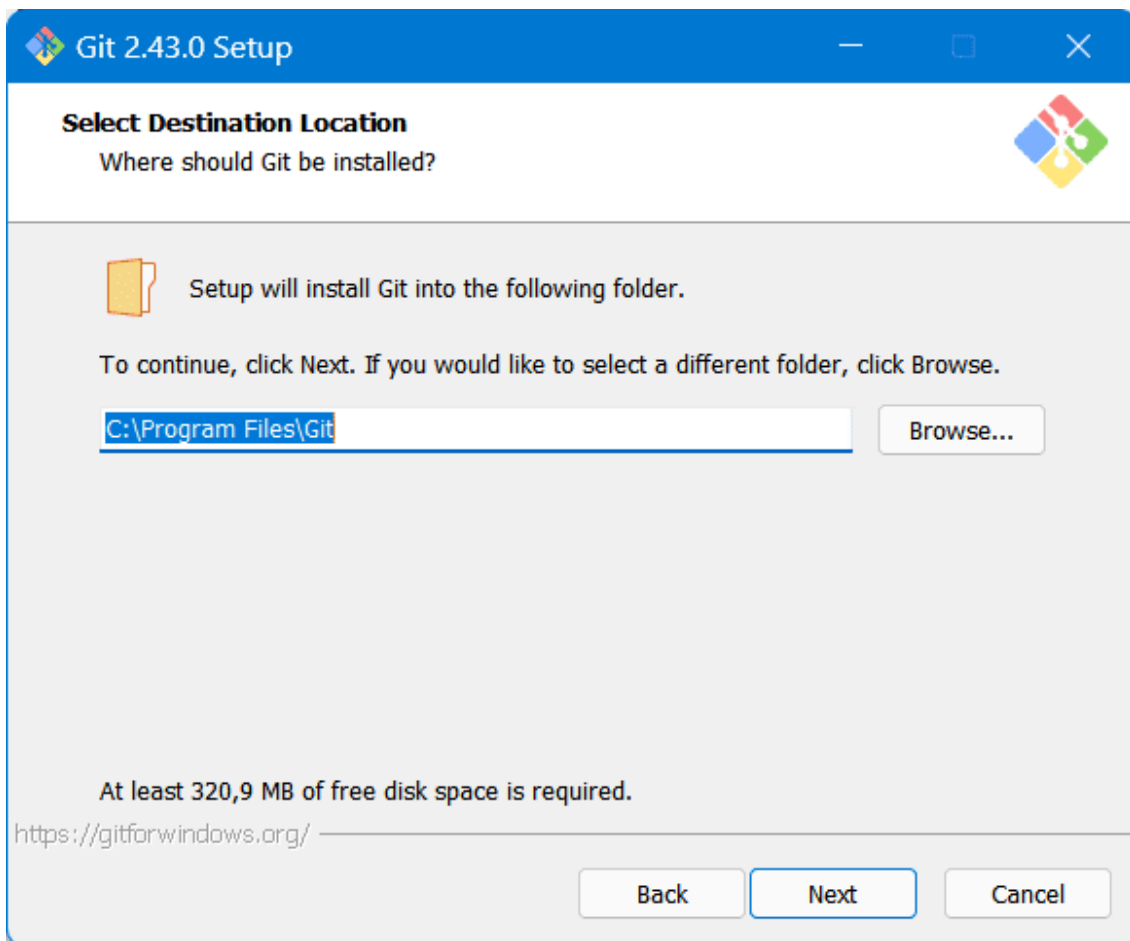
Karena saya menggunakan komputer berarsitektur 64-bit. Maka saya pilih download 64-bit Git for Windows Setup.

Setelah proses download Git selesai, buka file tersebut lalu klik *next*.

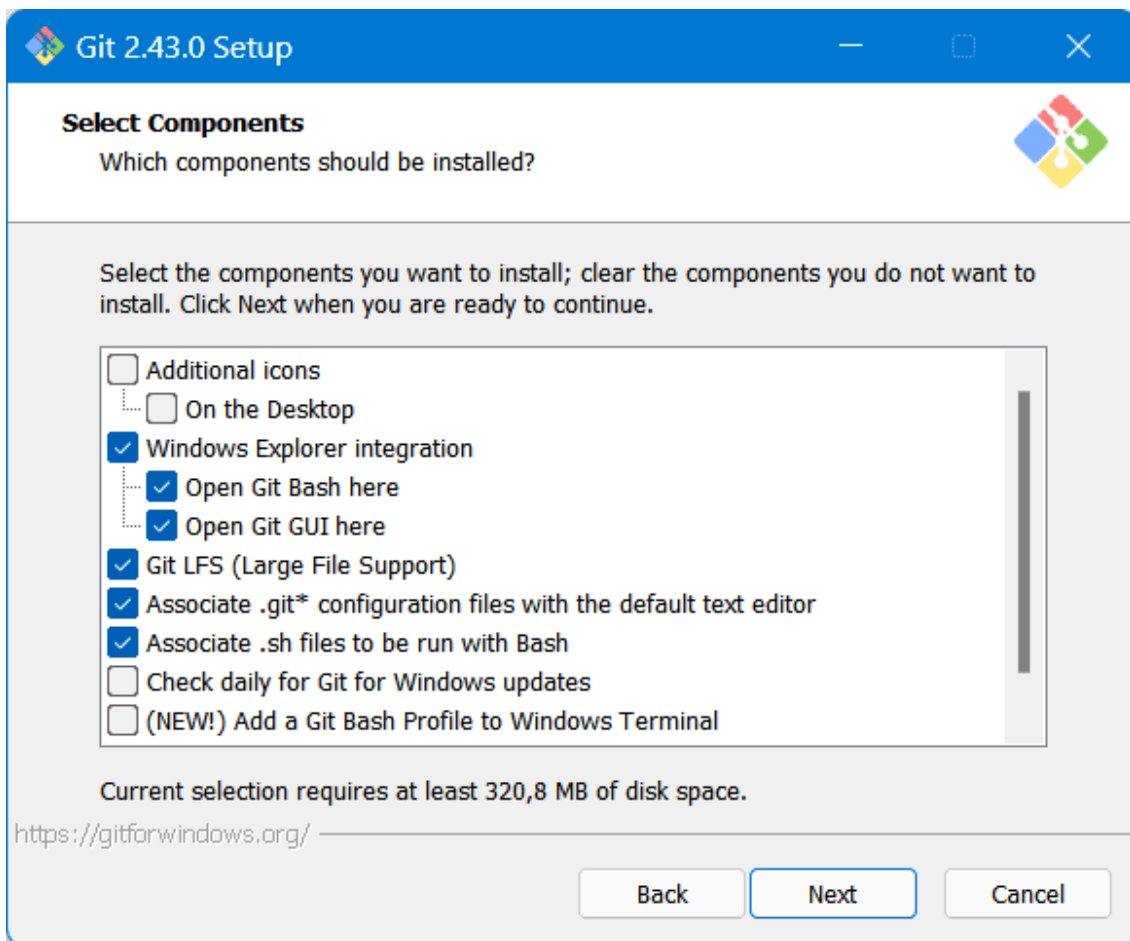


Klik next.

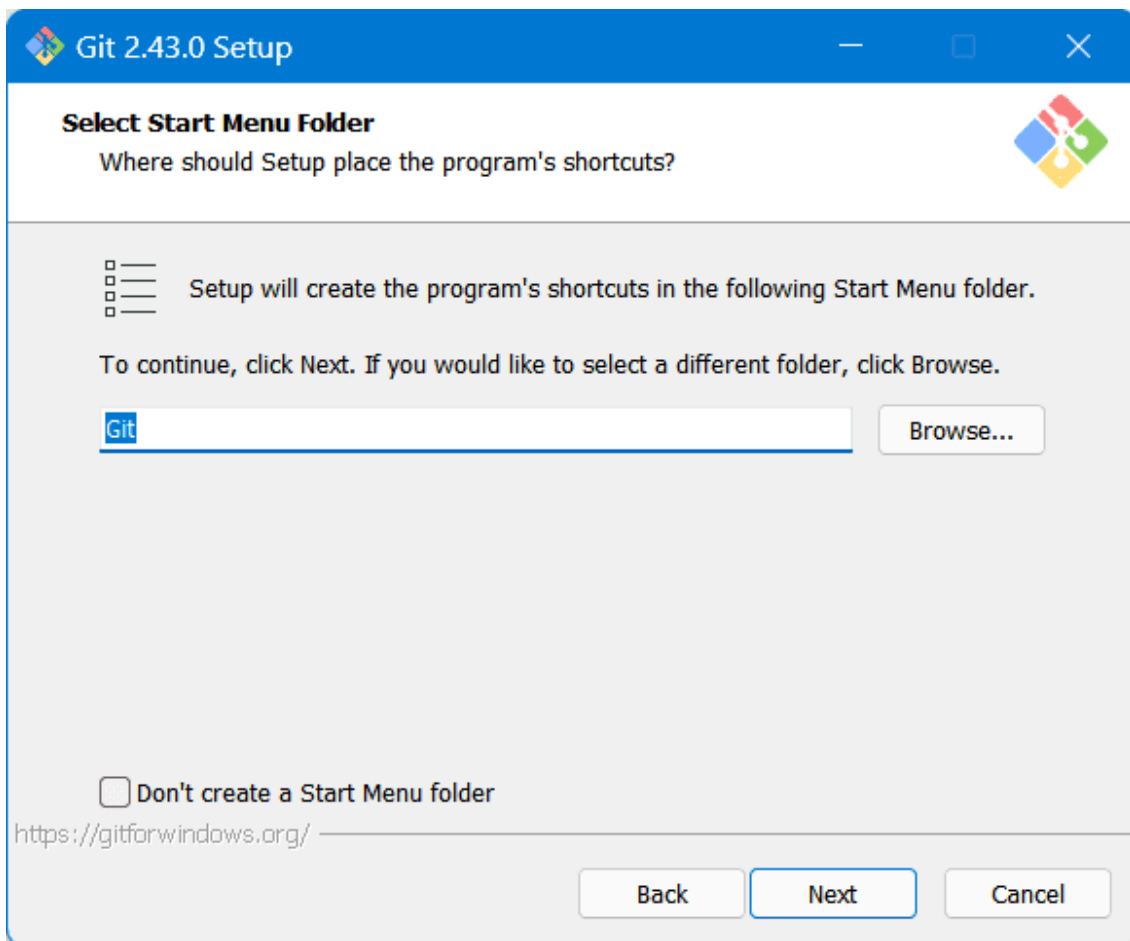
Untuk pengaturan lokasi penyimpanan Git saja secara default. dan klik *next* lagi.



Pilihan komponen yang akan ikut diinstall biarkan saja seperti bawaannya, lalu klik *next*.



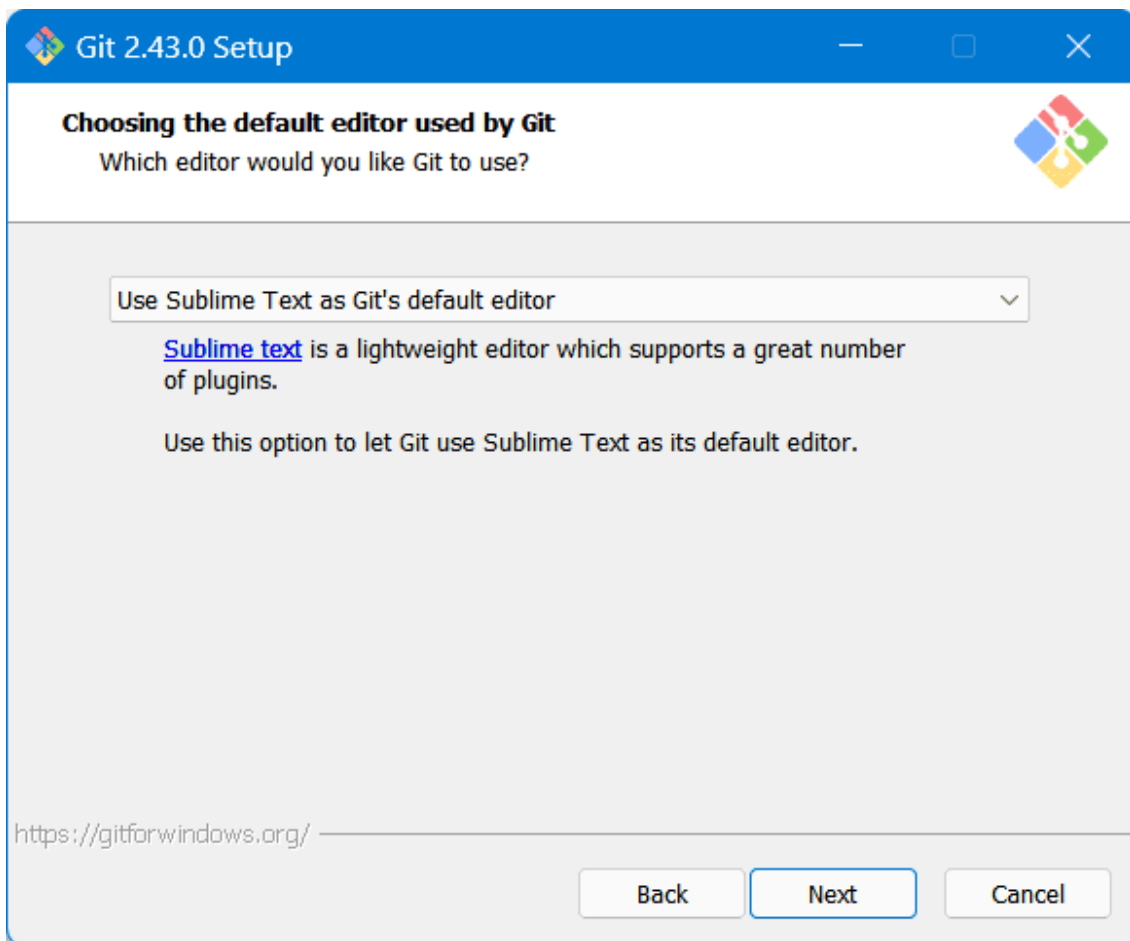
Penamaan shortcut Git pada menu biarkan saja, lalu klik *next*.



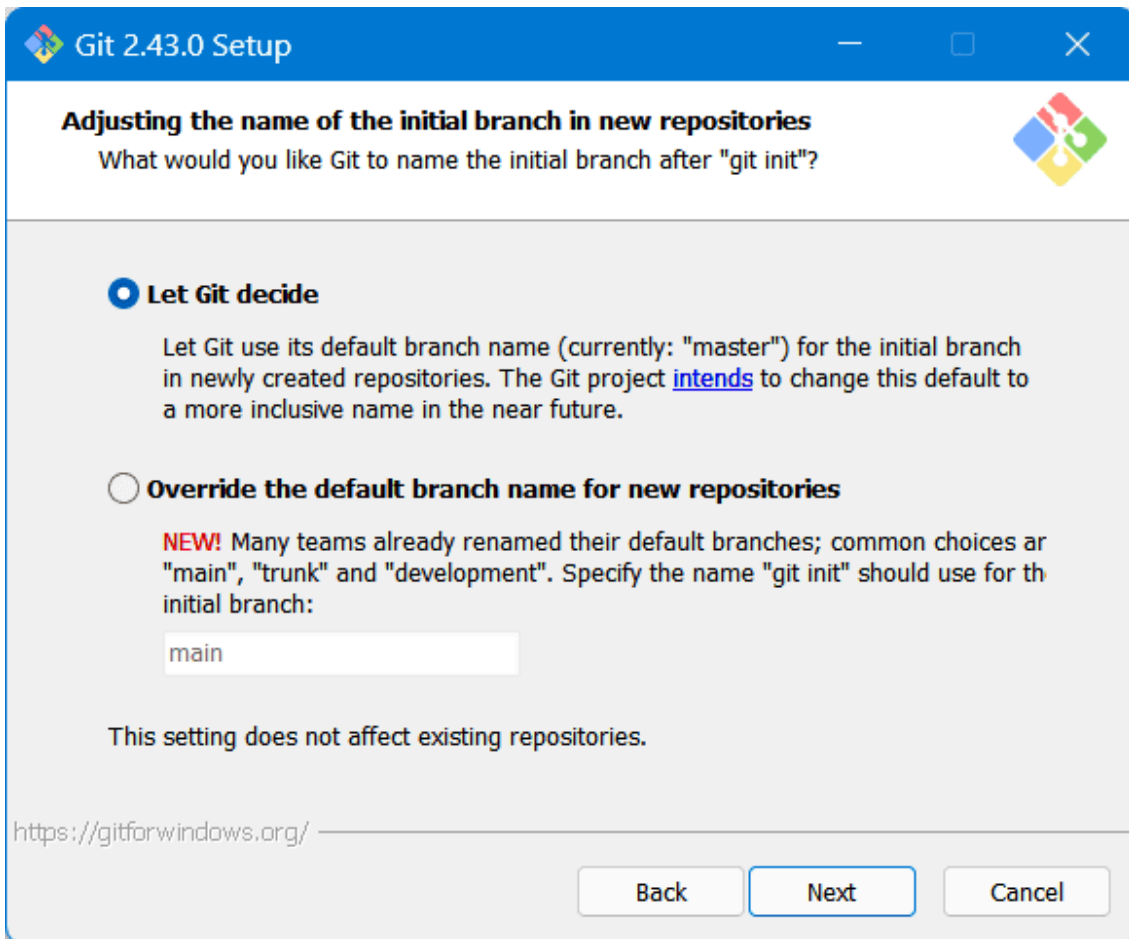
Kemudian ada pilihan *text editor* yang akan digunakan Git secara default. Di sini saya memilih sublime text editor.

Silahkan sesuaikan dengan pilihan *text editor* yang teman-teman gunakan.

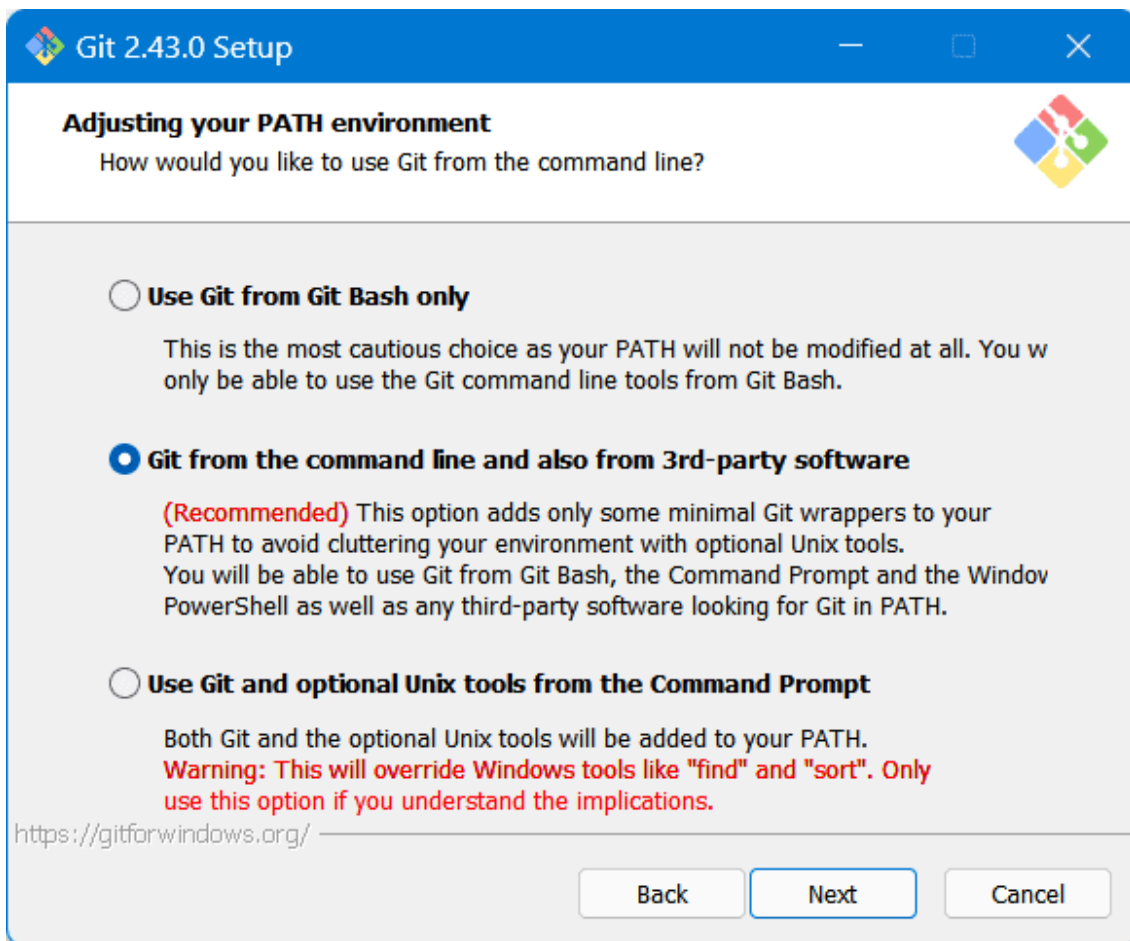
memilih text editor default untuk git



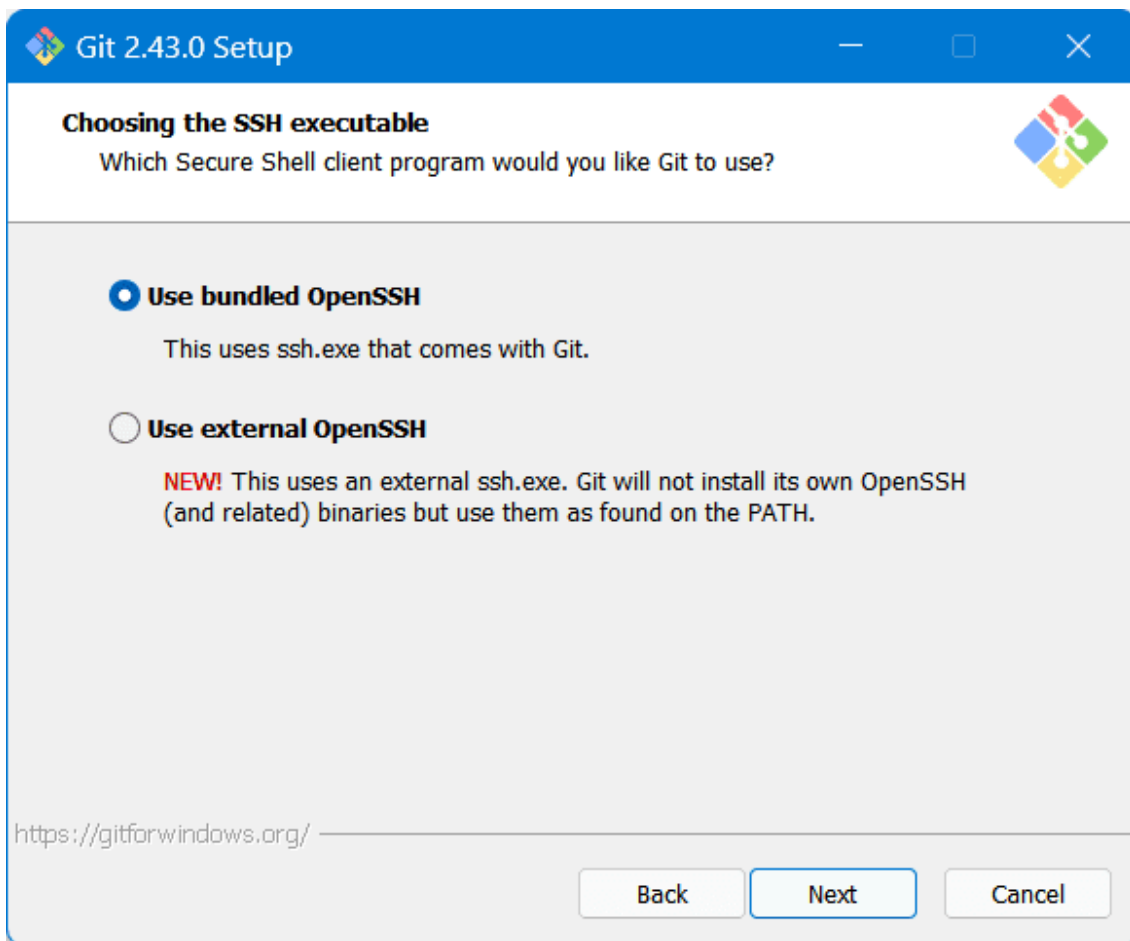
Pilihan `branch` default nya, biarkan saja dulu. klik *next*.



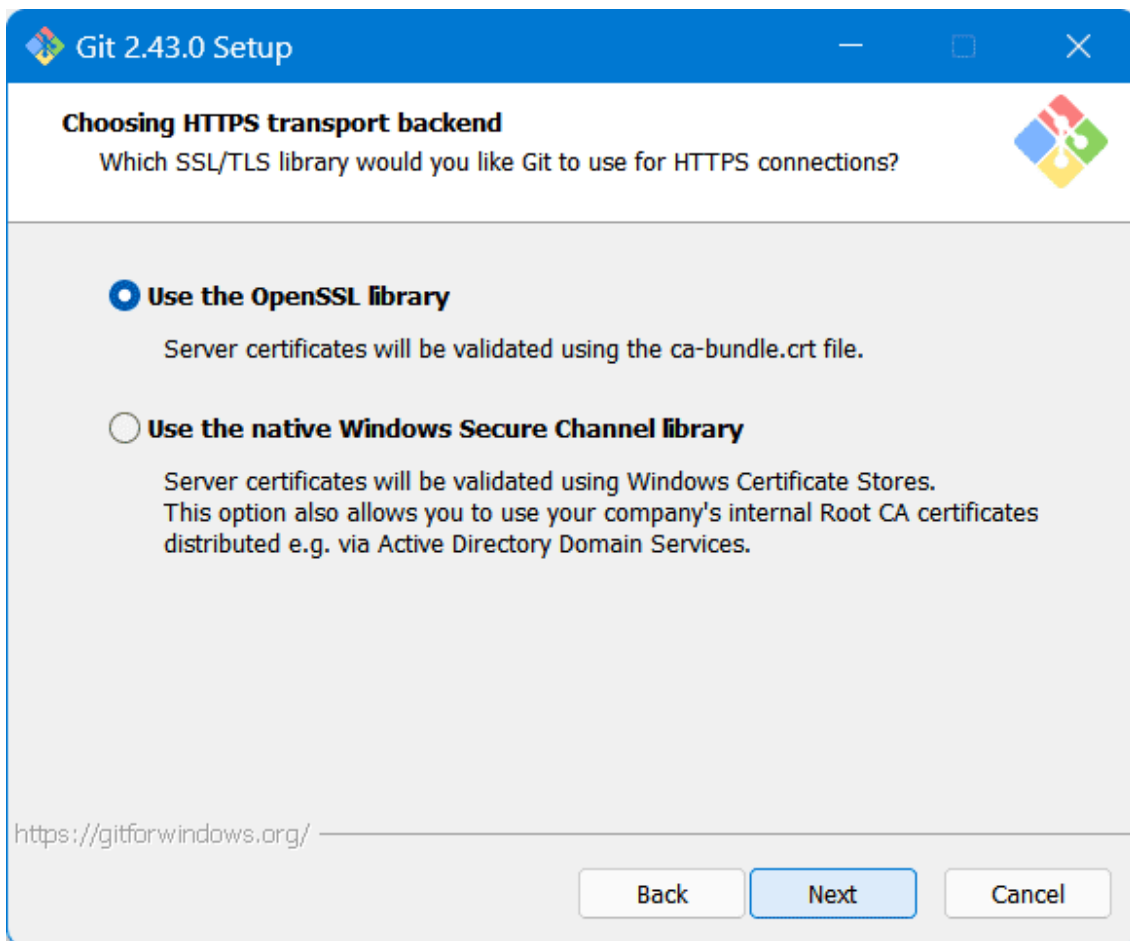
Sampai di step ini, pilih pilihan kedua, agar Git bisa mengenali CMD pada windows.



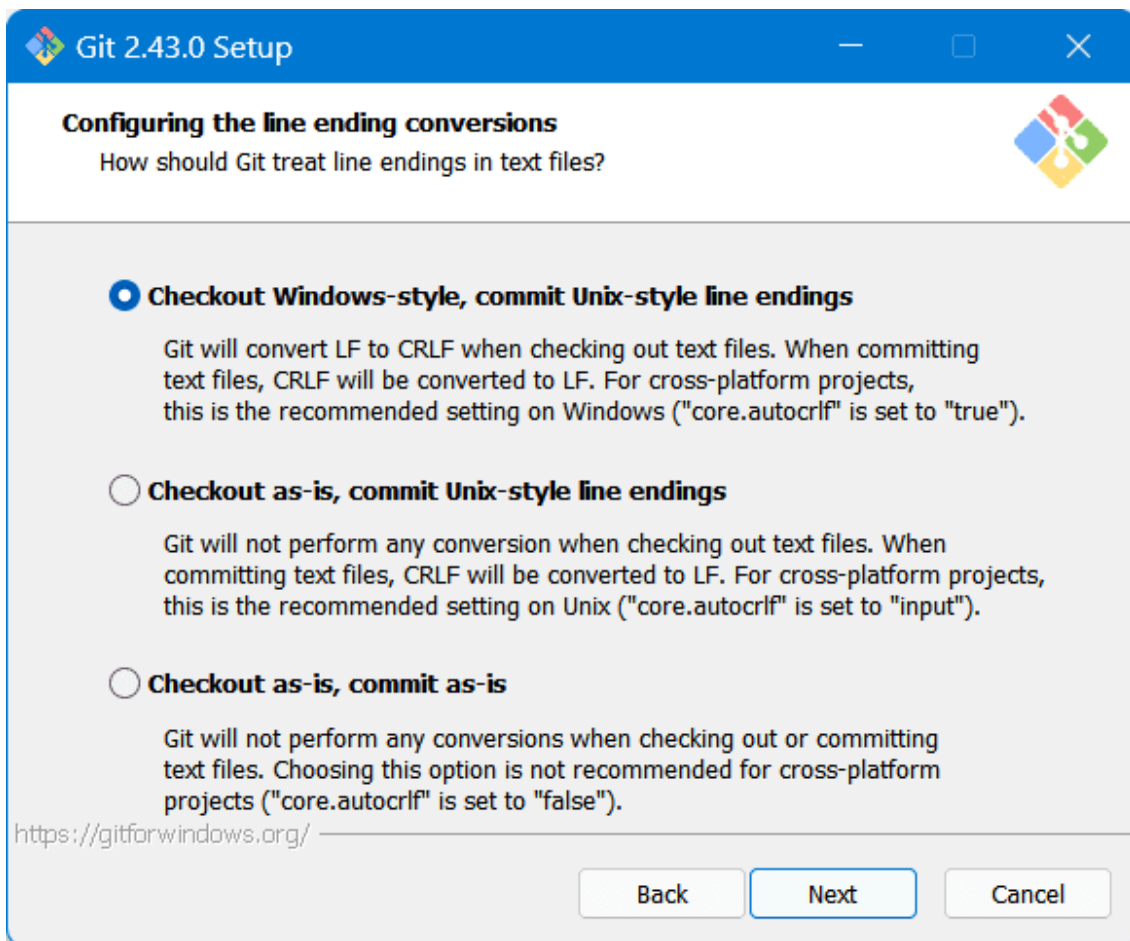
Klik *next* lagi ya teman-teman, tangan pegel kebanyakan klik *next*. hehe



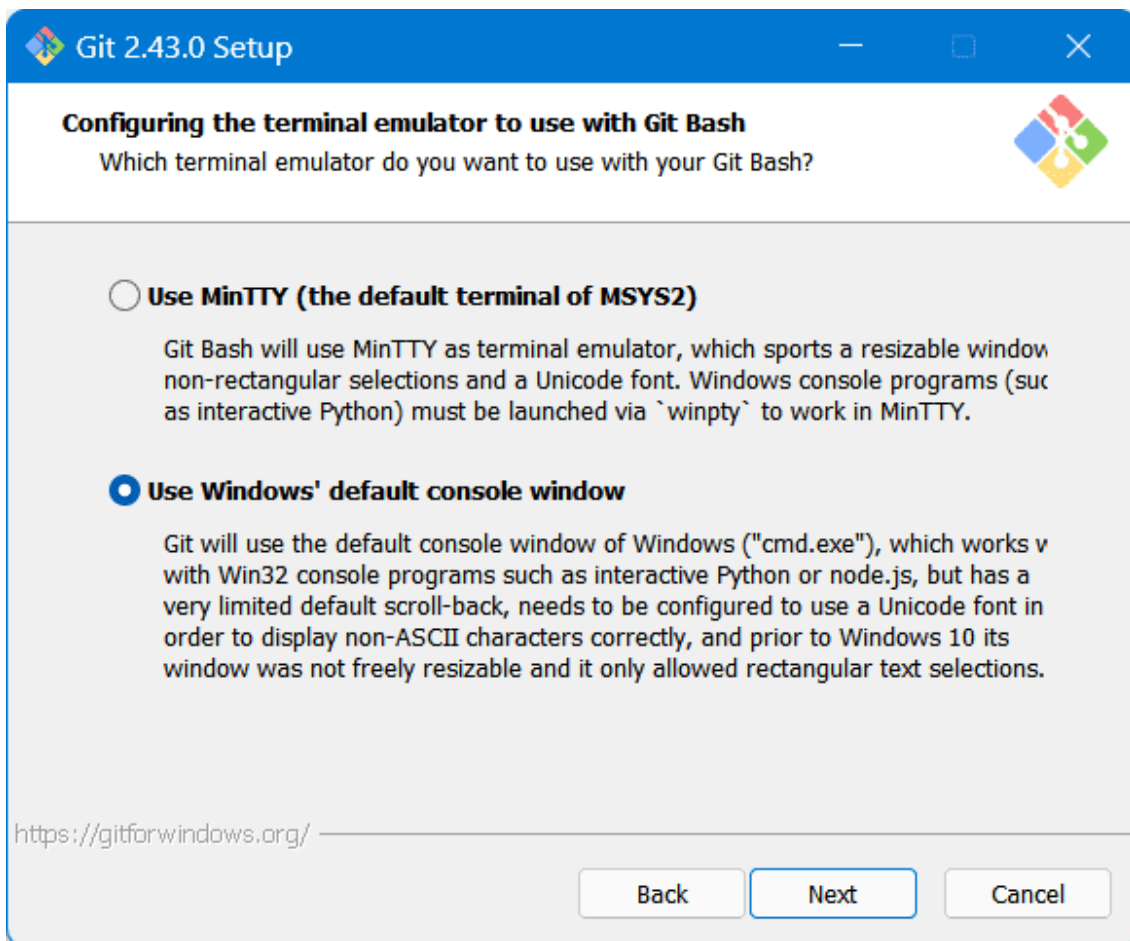
Next lagi. yuhuuu.



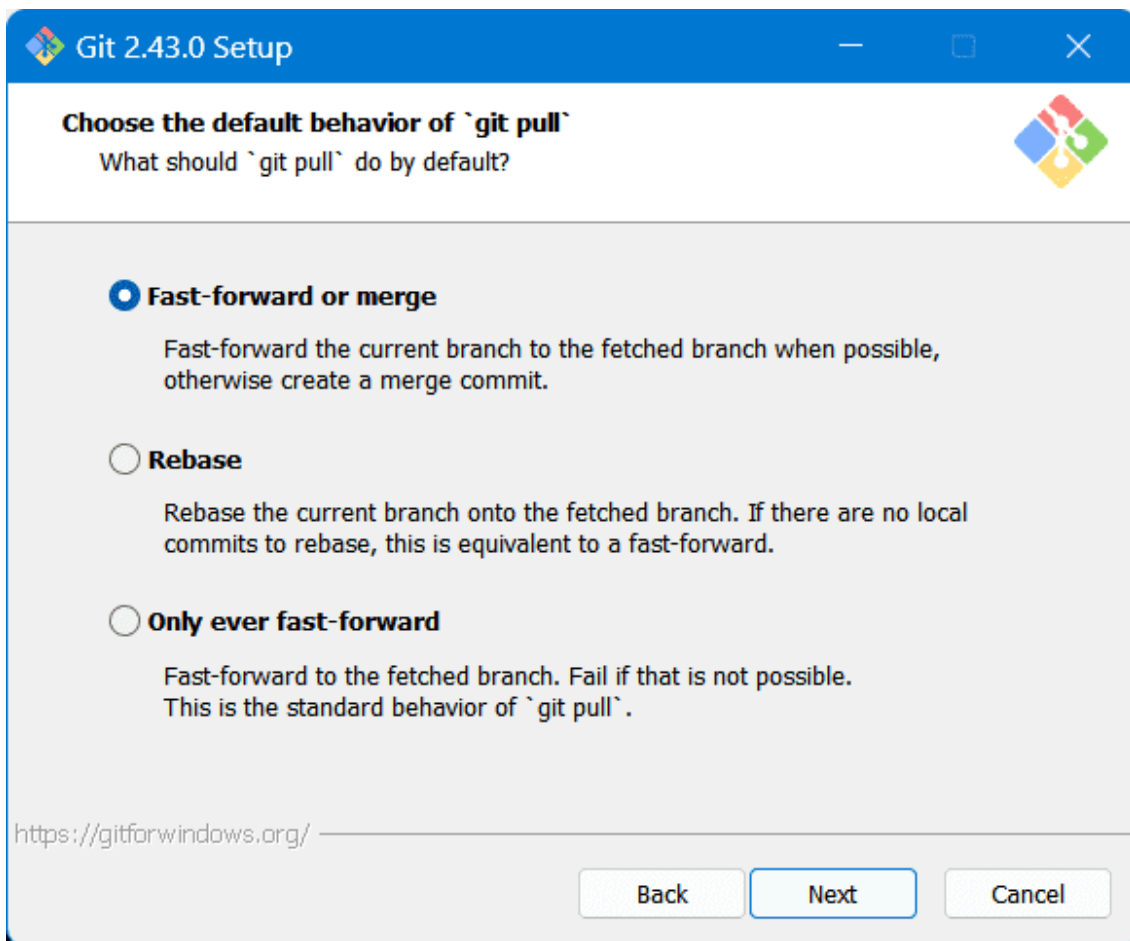
Lagi lagi *next* lagi.



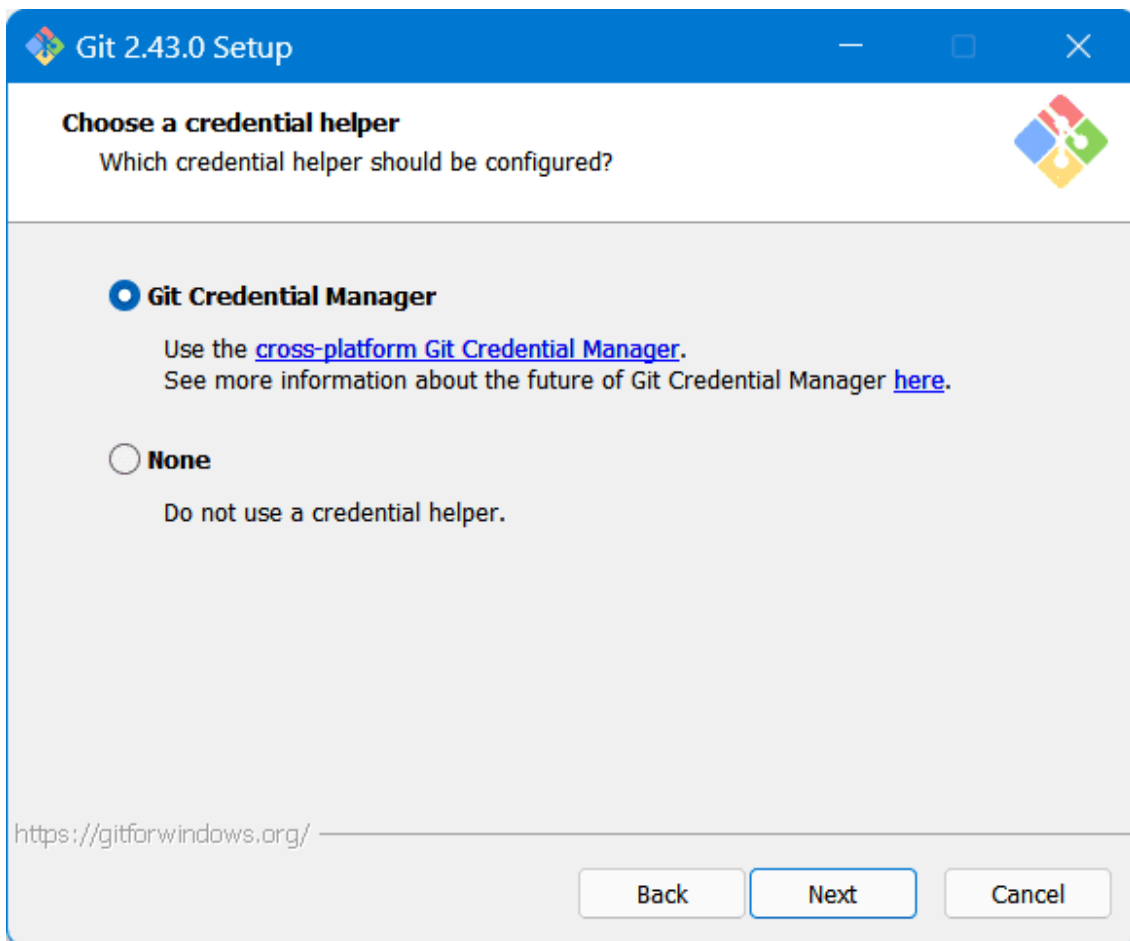
Next lagi.



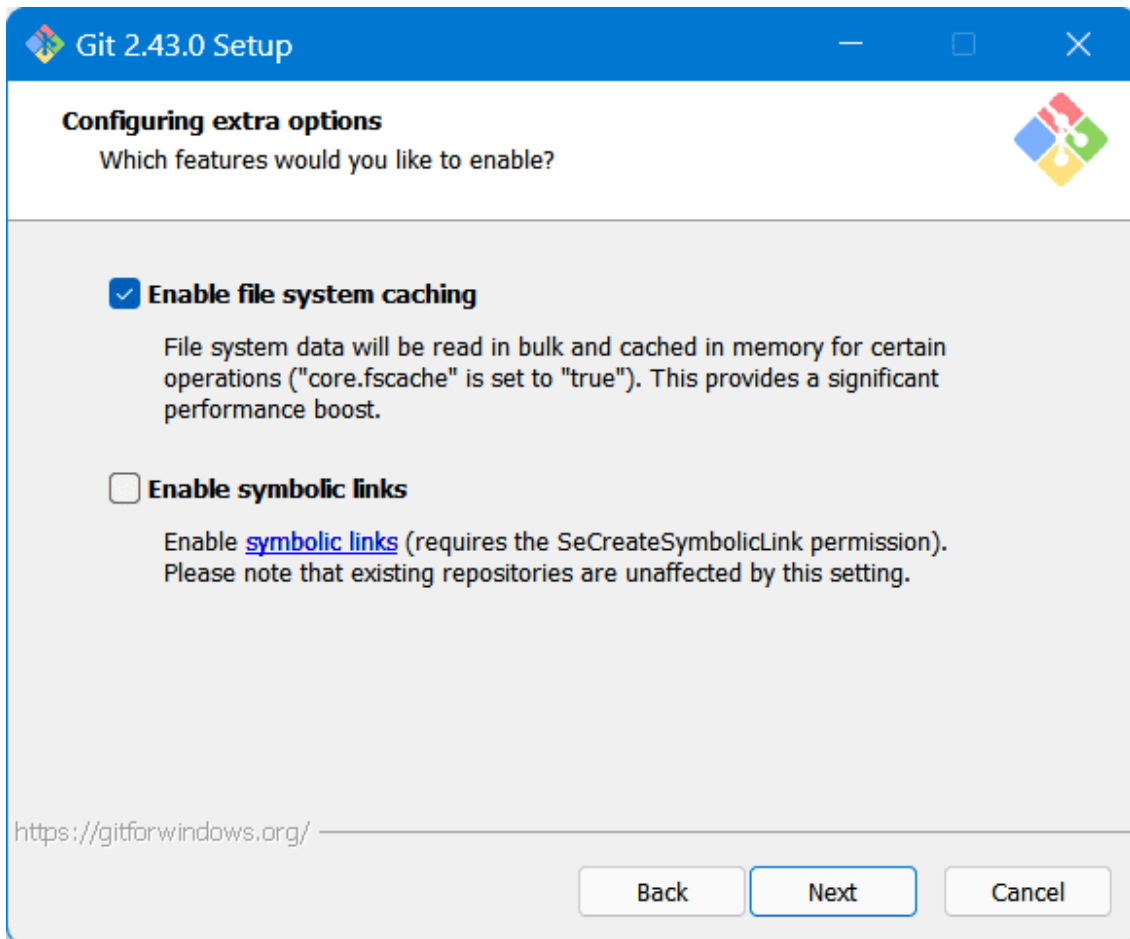
Next.



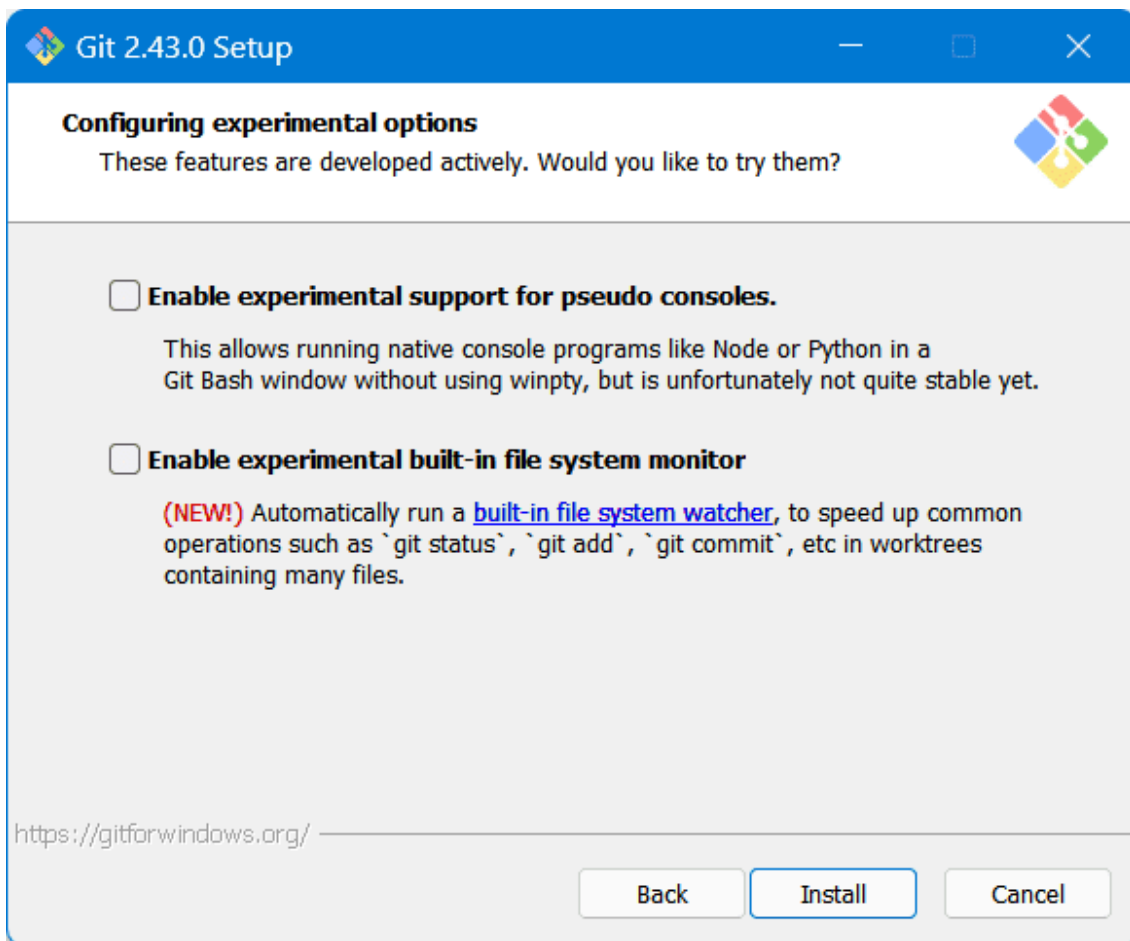
Lagi.



Teruskan sampai selesai.

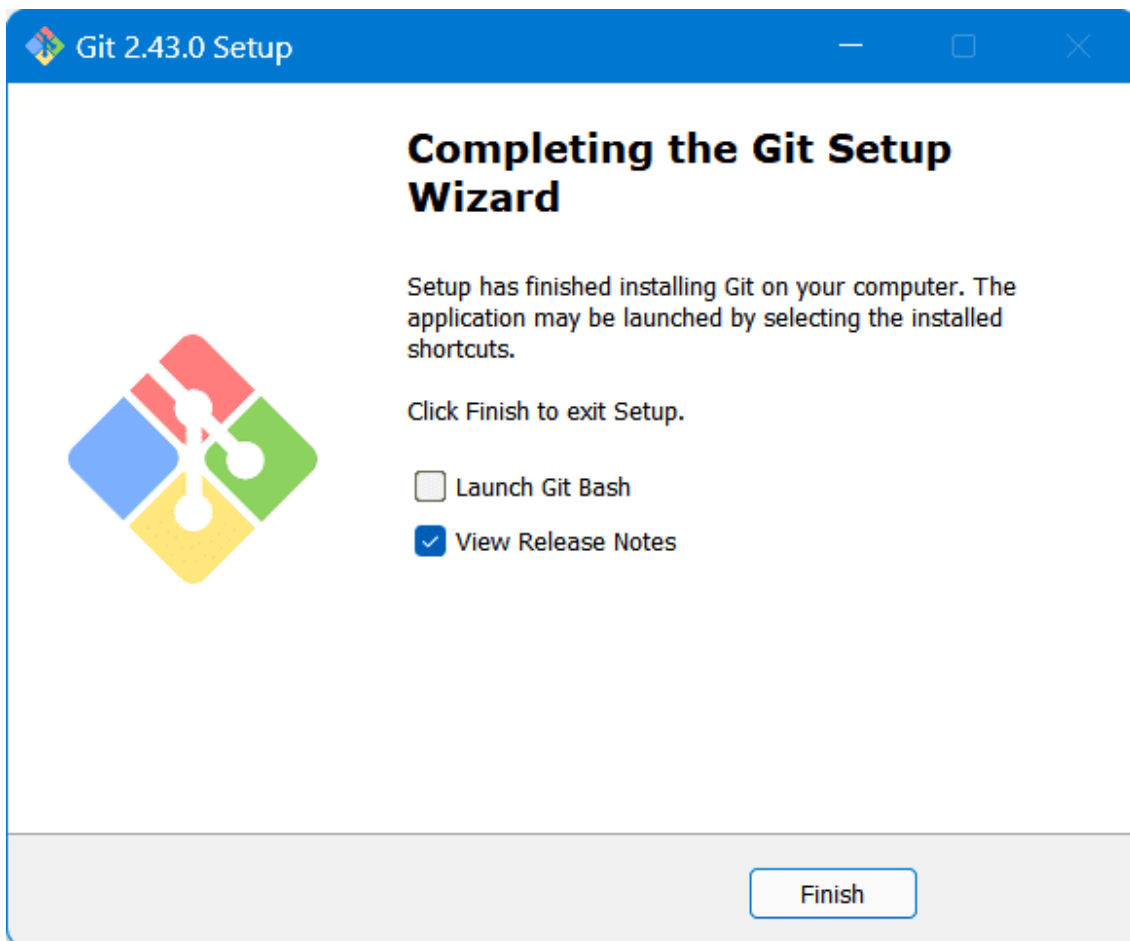


Klik *next* lagi.



Dan **proses instalasi Git di Windows** pun selesai.

Instalasi Git di windows selesai



Selesai.

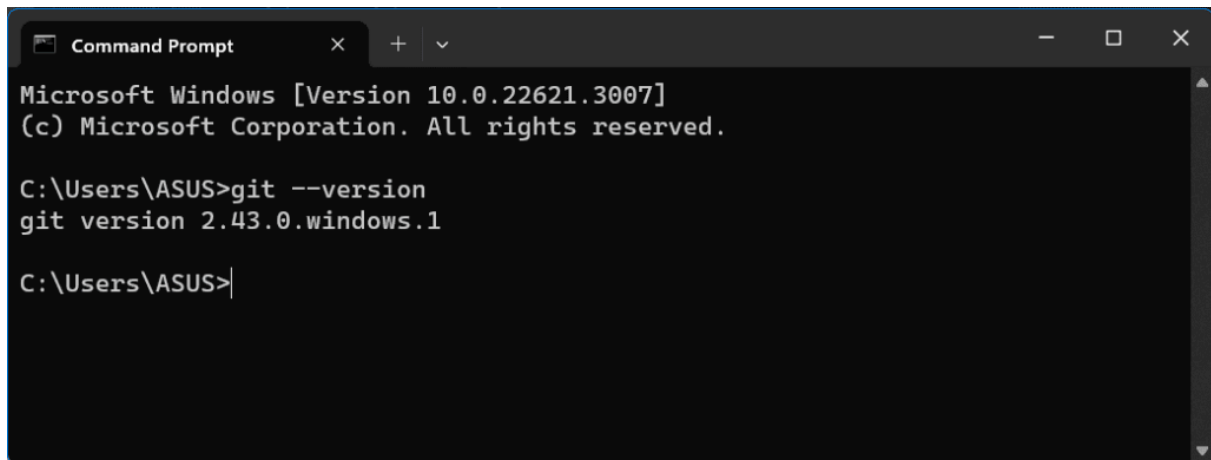
Untuk memeriksa apakah Git sudah benar-benar terinstall pada Windows teman-teman, silahkan buka CMD.

Lalu ketikkan perintah berikut.

```
git --version.
```

Jika Git sudah berhasil terinstall di Windows, maka akan muncul versi dari Git tersebut.

Cara install dan konfigurasi git



```
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>git --version
git version 2.43.0.windows.1

C:\Users\ASUS>
```

Sampai di sini, proses instalasi Git di Windows telah selesai.

Cara Konfigurasi Awal Git

Setelah proses instalasi Git selesai, yang harus kita lakukan selanjutnya adalah melakukan konfigurasi Git pada komputer kita.

Konfigurasi Nama dan Email

Konfigurasi yang akan kita buat adalah menentukan nama dan email kita pada Git.

Ketik perintah berikut pada terminal teman-teman.

```
git config --global user.name "Malas Ngoding"
```

Kemudian atur konfigurasi email kita dengan ketik perintah berikut.

```
git config --global user.email malasngoding@gmail.com
```

Perhatikan cara penulisannya ya, jangan sampai keliru.

Karena nantinya kita akan menggunakan email ini juga untuk mendaftar akun di layanan github, gitlab atau yang lain-lain.

Konfigurasi Branch (Cabang) Utama

Secara default, branch atau cabang yang digunakan oleh git adalah branch `master`.

Sebagai informasi, GitHub adalah layanan hosting untuk repository Git yang populer saat ini telah meresmikan perubahan branch utama dari master ke main.

Agar tidak ribet ketika nanti kita menggunakan Github, ada baiknya kita ubah saja branch utama pada git lokal kita.

untuk mengubah branch utama ke branch main, kita bisa mengetik perintah berikut pada terminal.

```
git config --global init.defaultBranch main
```

Git dan GitHub Panduan Dasar untuk Pemula



Dalam dunia pengembangan perangkat lunak, penggunaan alat manajemen versi dan platform kolaborasi sangat penting untuk menjaga kualitas dan integritas kode. Salah satu alat yang sangat populer dalam hal ini adalah Git dan GitHub. Artikel ini akan menjelaskan secara mendalam tentang Git dan GitHub, termasuk pengertian, fungsi, dan cara menggunakannya.

Apa Itu GitHub?

GitHub adalah platform berbasis web yang digunakan untuk hosting dan pengelolaan kode sumber menggunakan sistem kontrol versi Git. GitHub memungkinkan pengembang untuk menyimpan, melacak, dan mengelola proyek perangkat lunak mereka secara efisien. Platform ini menyediakan berbagai fitur tambahan yang mendukung kolaborasi tim dan pengelolaan proyek yang lebih baik.

Pengertian GitHub

GitHub adalah layanan hosting berbasis cloud yang memungkinkan pengembang untuk menyimpan proyek mereka menggunakan sistem kontrol versi Git. Dengan GitHub, Anda bisa mengelola repositori kode, melakukan review kode, dan berkolaborasi dengan tim atau komunitas secara efektif. GitHub menyediakan antarmuka pengguna yang intuitif dan berbagai fitur tambahan seperti pull request, issues, dan GitHub Actions yang mempermudah proses pengembangan perangkat lunak.

Fungsi GitHub

GitHub menawarkan berbagai fungsi yang sangat berguna dalam pengembangan perangkat lunak, terutama dalam hal kolaborasi dan pengelolaan kode. Berikut adalah beberapa fungsi utama dari GitHub:

1. Memudahkan Kolaborasi Pengerjaan Project

Salah satu fungsi utama GitHub adalah memfasilitasi kolaborasi antara anggota tim dalam pengerjaan proyek perangkat lunak. Dengan GitHub, tim dapat bekerja pada proyek yang sama secara bersamaan tanpa mengganggu pekerjaan satu sama lain. Fitur seperti pull request dan branch memungkinkan pengembang untuk mengusulkan perubahan, melakukan review, dan mengintegrasikan perubahan ke dalam kode utama dengan cara yang terstruktur.

2. Mencegah Perubahan Kode yang Bisa Merusak Kode Asli

GitHub membantu mencegah masalah yang mungkin timbul akibat perubahan kode yang tidak diinginkan. Dengan menggunakan sistem kontrol versi Git, setiap perubahan yang dilakukan pada kode disimpan dalam bentuk commit. Jika terjadi kesalahan atau bug, Anda dapat dengan mudah kembali ke versi sebelumnya dari kode tanpa kehilangan perubahan lain yang telah dilakukan. Ini memastikan bahwa kode utama tetap stabil dan aman dari perubahan yang merusak.

3. Sebagai Portofolio Bagi Developer

GitHub juga berfungsi sebagai portofolio bagi para pengembang perangkat lunak. Dengan memiliki akun GitHub, pengembang dapat memamerkan proyek-proyek yang telah mereka kerjakan, berkontribusi pada proyek open source, dan menunjukkan kemampuan mereka kepada calon employer atau klien. Profil GitHub yang aktif dan terisi dengan baik dapat meningkatkan visibilitas dan kredibilitas seorang pengembang di dunia industri TI.

Cara Menggunakan GitHub

Menggunakan GitHub mungkin terlihat rumit pada awalnya, tetapi dengan panduan ini, Anda akan dapat memahami dasar-dasar cara menggunakan GitHub dengan mudah. Berikut adalah langkah-langkah untuk memulai dengan GitHub:

1. Buat Akun GitHub

Langkah pertama dalam cara menggunakan GitHub adalah membuat akun GitHub. Kunjungi situs web GitHub di github.com dan klik tombol "Sign up" untuk membuat akun baru. Anda akan diminta untuk memasukkan alamat email,

nama pengguna, dan kata sandi. Setelah mendaftar, Anda akan menerima email konfirmasi untuk mengaktifkan akun Anda.

2. Memulai Project Baru

Setelah memiliki akun GitHub, langkah berikutnya adalah memulai proyek baru. Setelah masuk ke akun GitHub Anda, klik tombol "New repository" pada halaman utama atau profil Anda. Isi nama repositori, deskripsi, dan pilih opsi visibilitas (publik atau privat). Anda juga dapat memilih untuk menambahkan README file dan lisensi jika diinginkan. Klik tombol "Create repository" untuk membuat repositori baru.

3. Buat File Kode

Dengan repositori yang telah dibuat, Anda bisa mulai menambahkan file kode ke dalamnya. Anda dapat meng-upload file langsung melalui antarmuka web GitHub dengan mengklik tombol "Add file" dan memilih "Upload files". Pilih file kode yang ingin di-upload dari komputer Anda dan klik "Commit changes" untuk menyimpan perubahan.

4. Mengedit File Kode

GitHub juga memungkinkan Anda untuk mengedit file kode langsung dari antarmuka web. Klik pada file yang ingin Anda edit, lalu klik tombol "Edit" (ikon pensil) di sudut kanan atas. Lakukan perubahan yang diperlukan, tambahkan deskripsi perubahan, dan klik "Commit changes" untuk menyimpan perubahan ke repositori.

5. Kolaborasi dengan Developer atau Project Lain

Fitur kolaborasi di GitHub sangat powerful. Anda dapat mengundang kolaborator untuk bergabung dengan repositori Anda dengan mengklik tab "Settings" pada repositori dan memilih "Collaborators". Masukkan nama pengguna GitHub mereka dan klik "Add collaborator". Kolaborator akan menerima undangan untuk bergabung dan dapat mulai bekerja pada proyek bersama Anda.

Kesimpulan

Git dan GitHub adalah alat yang sangat penting dalam dunia pengembangan perangkat lunak saat ini. Git sebagai sistem kontrol versi memungkinkan pengembang untuk melacak perubahan kode, mengelola versi, dan memelihara integritas kode. GitHub, di sisi lain, menyediakan platform berbasis web yang mempermudah kolaborasi, manajemen proyek, dan pengelolaan kode menggunakan Git.

Dengan memahami pengertian GitHub, fungsi GitHub, dan cara menggunakan GitHub, Anda akan dapat memanfaatkan alat ini untuk meningkatkan efisiensi kerja, berkolaborasi dengan tim, dan memperkuat portofolio Anda sebagai pengembang. Memanfaatkan Git dan GitHub dengan baik adalah langkah penting untuk menjadi seorang pengembang perangkat lunak yang sukses dan produktif.



Langkah Menjadi Programmer Hebat

Untuk benar-benar meraih predikat *programmer* ahli, Anda tidak boleh asal belajar. Terapkan prinsip berikut:

- **Tetapkan Fokus Bahasa:** Sebelum mulai, tentukan target aplikasi apa yang ingin dibangun, dan pelajari hanya satu bahasa dasar terlebih dahulu (misalnya Python atau JavaScript) agar fokus.
- **Gunakan Algoritma Efisien:** Algoritma adalah urutan pemecahan masalah. *Programmer* hebat tidak hanya sekadar membuat aplikasi yang "berjalan", melainkan menyempurnakan algoritmanya secara berkala agar aplikasi menggunakan *resource* memori yang sangat minim dan bekerja lebih cepat.
- **Jangan Berhenti saat Berjalan:** Kebanyakan *programmer* pemula berpuas diri ketika program tidak memunculkan *error* lagi. *Programmer* sejati secara aktif meninjau celah kesalahan (*debugging*) dan menulis kode agar kualitasnya lebih rapi, terstruktur, serta menyusun **Automated Testing** agar pengujian program dapat dilakukan secara otomatis untuk meminimalisir masalah sebelum dirilis.
- **Sadari Pentingnya Security:** Perhatikan keamanan aplikasi dengan menguasai praktik *secure coding* untuk mencegah serangan berbahaya pencurian data, semacam *SQL Injection* atau *Buffer overflow*.
- **Teruslah Membaca Kode:** Baca ulang kode yang Anda buat, dan rajinlah membaca dokumentasi atau kode milik *developer* lain. Hal ini membantu Anda mempelajari metode baru dalam memecahkan masalah (*problem solving*) dengan lebih efisien.

- **Gunakan Internet:** Masalah dan *bug* di awal pembelajaran hampir selalu ada. Jangan ragu menggunakan mesin pencari (seperti Google) dan aktif di forum diskusi/komunitas *open source* untuk menemukan solusi *error*.
- **Nikmati Prosesnya:** Tanamkan pemahaman bahwa "belajar pemrograman itu menyenangkan" agar sugesti positif ini membantu Anda menikmati setiap baris kode yang Anda ketik tanpa merasa terbebani sebagai kewajiban belaka.



Pengantar Algoritma dan Flowchart

Definisi Algoritma

- Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis.
- Kata logis (logika) merupakan kata kunci dalam
- Langkah-langkah dalam algoritma harus dapat ditentukan bernilai benar atau

Konsep Dasar Algoritma

- Algoritma adalah kumpulan intruksi/perintah yang dibuat secara jelas dan sistematis berdasarkan urutan yang logis (logika) untuk penyelesaian suatu
- French, s. (1984) menyatakan sejumlah konsep yang mempunyai relevansi dengan masalah rancangan program yaitu kemampuan computer, kesulitan dan ketepatan.
- Penerapan dari konsep tersebut biasanya digunakan dalam rancangan algoritma.

Konsep Dasar Algoritma

Dalam merancang sebuah algoritma, Fletcher (1991) memberikan beberapa cara atau metode yaitu kumpulan perintah, ekspresi, tabel intruksi, program computer, kode semu, dan flow chart, sedangkan Knuth (1973) menyarankan

algoritma fundamental untuk keperluan matematika dan program computer, metode yang sering digunakan yaitu:

- Diagram Alir (Flow chart)
- Kode Semu (Pseudo Code)
- Algoritma Fundamental