

MODUL 1. INSTALASI IDE DAN GIT

1.1 Tujuan

1. Mahasiswa mampu melakukan instalasi Java, Text Editor dan Git

1.2 Alat dan Bahan

1. JDK versi 21
2. Visual Studio Code
3. Git

1.3 Dasar Teori

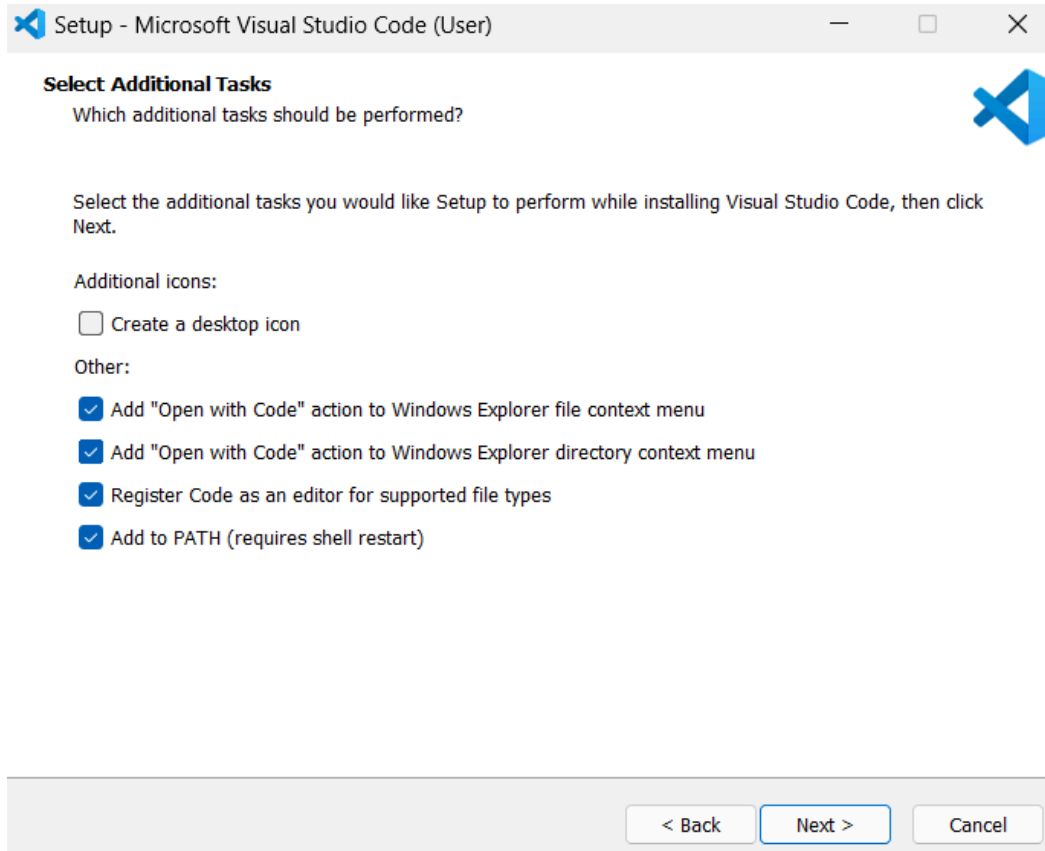
Halo, di sini, kalian bakalan mempelajari Implementasi Algoritma, yang merupakan langkah awal dari perjalanan programming kalian. Bener, lho! Implementasi Algoritma ini adalah mata kuliah yang akan mengenalkan kalian kepada basic programming di prodi D3 RPLA ini. Di sini, kalian akan belajar membuat algoritma dan mengimplementasikannya dalam bahasa Java.

Namun, sebelum kita masuk ke pemrograman, kita membutuhkan beberapa hal penting! Di antaranya, tentu alat atau tools yang dibutuhkan untuk membantu kalian memprogram, compiler, dan juga git. Lalu apa saja tools yang dibutuhkan? Dan apa itu compiler dan git? Yuk, mari kita bahas!

1.3.1 Tools

Tools atau alat-alat yang dibutuhkan untuk membantu pemrograman yang paling utama adalah text editor atau editor teks. Kalau kalian tahu notepad, notepad adalah salah satu editor teks bawaan Windows. Tapi, notepad kelihatan hmm... Maaf, agak jelek ya? Karena itu, kita akan memakai sesuatu yang lebih modern, yaitu Visual Studio Code, atau disingkat VSCode!

Untuk melakukan instalasi VSCode, kalian bisa men-downloadnya terlebih dahulu di <https://code.visualstudio.com/download> dan pilihlah sesuai dengan sistem operasi kalian! Setelah itu, apabila menu berikut muncul:



Maka ceklis saja 4 itu, dan kalian juga boleh ceklis **Create a desktop icon** bila ingin menambahkan VSCode ke desktop komputer kalian. Kemudian, lakukan instalasi secara biasa.

1.3.2 Compiler

Sudah beres melakukan instalasi VSCode? Bagus! Sekarang, kita akan membahas apa itu compiler. Jadi, kalian tentunya sudah mengetahui apa itu bahasa pemrograman, kan? Itu adalah bahasa yang akan kalian gunakan untuk berkomunikasi dengan komputer! Bahasa yang akan digunakan untuk melakukan implementasi dari logika kalian, dan membentuk sebuah aplikasi. Nah, tapi gimana sih sebuah aplikasi itu bisa dijalankan? Kan awalnya hanya dalam bentuk kode aja? Nah! Itu karena ada compiler!

Dengan adanya compiler ini, kode yang kalian tulis akan digabungkan (compile), kemudian akan diproses hingga menjadi bentuk binary atau biner, yang mungkin biasa kalian kenal dengan file “.exe” atau file executable, yang artinya, aplikasi kalian bisa dieksekusi atau dijalankan! Nah, bahasa yang akan kita gunakan dalam mata kuliah Implementasi Algoritma ini adalah Java, sehingga yang harus kita lakukan adalah melakukan instalasi dari compiler Java. Namun, kita tidak hanya akan menginstall compiler itu sendiri, melainkan satu paket lengkap toolkit untuk Java yang juga memiliki compiler Java di dalamnya, yaitu JDK atau Java Development Toolkit. Pertama-tama, masuklah ke laman <https://www.oracle.com/id/java/technologies/downloads/#java21> dan pilih OS yang sesuai dengan OS

yang kamu pakai! Kita memakai JDK 21, atau versi Java 21 karena versi 21 adalah versi LTS atau Long Term Support yang terbaru. LTS di sini artinya dia akan didukung terus sama oracle untuk waktu yang lama.

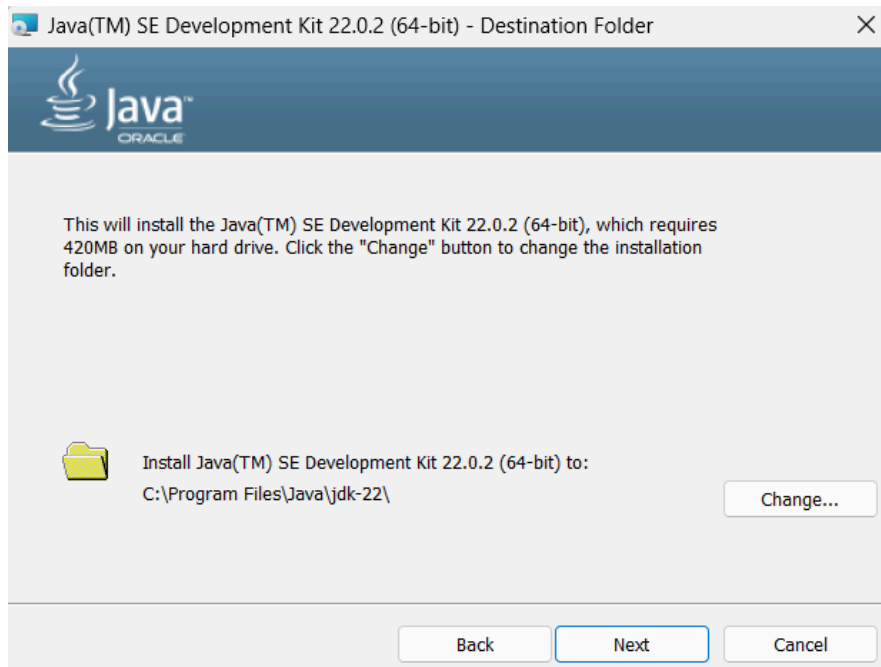
1.3.2.1 Windows

Pilih saja x64 Installer, kemudian ia akan di-download.

Linux	macOS	Windows
Product/file description	File size	Download
x64 Compressed Archive	184.16 MB	https://download.oracle.com/java/22/latest/jdk-22_windows-x64_bin.zip (sha256)
x64 Installer	164.35 MB	https://download.oracle.com/java/22/latest/jdk-22_windows-x64_bin.exe (sha256)
x64 MSI Installer	163.09 MB	https://download.oracle.com/java/22/latest/jdk-22_windows-x64_bin.msi (sha256)

Bila proses download sudah selesai, kalian bisa membuka installernya dan menu installer akan terbuka.





Kalian bisa menekan next untuk semuanya, dan Java akan terinstall di OS kalian.

1.3.2.2 Linux

Untuk pengguna Linux, harusnya udah tahu dong apa yang harus dilakukan. Pada Ubuntu dan turunannya yang memakai apt, kita bisa langsung saja mengetik perintah berikut setelah melakukan update repo.

```
>> "Linux Mint" - rahman @ furude-PC in ~
-> sudo apt install openjdk-21-jdk openjdk-21-jre
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-5.15.0-117 linux-headers-5.15.0-117-generic
  linux-image-5.15.0-117-generic linux-modules-5.15.0-117-generic
  linux-modules-extra-5.15.0-117-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  openjdk-21-jdk-headless openjdk-21-jre-headless
Suggested packages:
  openjdk-21-demo openjdk-21-source visualvm fonts-ipafont-gothic
  fonts-ipafont-mincho fonts-wqy-microhei | fonts-wqy-zenhei
The following NEW packages will be installed:
  openjdk-21-jdk openjdk-21-jdk-headless openjdk-21-jre
  openjdk-21-jre-headless
0 upgraded, 4 newly installed, 0 to remove and 82 not upgraded.
Need to get 131 MB of archives.
After this operation, 301 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Lalu masukkan Y.

Rizza Mandasari, Rahman Hakim

Ilustrasi oleh Nazwa Mursyidan Baldan

1.3.2.3 Mac

Pengguna mac dapat menginstall java melalui link di atas, atau menginstallnya langsung dengan package manager *homebrew* seperti berikut.

```
$ brew install openjdk@21
```

1.3.2.4 Melakukan cek instalasi Java

Untuk melakukan cek apakah Java kalian telah terinstall dengan baik, ketiklah perintah “java –version” pada command line kalian.

```
C:\Users\msi GF66>java --version
java 22.0.2 2024-07-16
Java(TM) SE Runtime Environment (build 22.0.2+9-70)
Java HotSpot(TM) 64-Bit Server VM (build 22.0.2+9-70, mixed mode, sharing)
```

Bila sudah terinstall, maka versi Java kalian akan bisa dilihat.

1.3.3 Git dan GitHub

Oke, kalau begitu, sekarang kita akan masuk ke Git, dan juga GitHub. Jadi apasih Git ini? Lalu apa bedanya Git dan juga GitHub? Oke, pertama-tama apakah kalian pernah bikin beberapa file berbeda, yang isinya revisian, yang mungkin sampai banyak banget? Hayo ngaku berapa banyak yang bikin versi Tugas Flx, Tugas Fix Banget, Tugas Fix Fix Dikumpulin. Hahahaha. Nah, kan filenya jadi banyak tuh, dan jadi gak praktis, benar? Lalu, coba bayangin kalian ada sebuah proyek aplikasi, dan yang kerja banyak orang. Masa seluruh file di dalam folder proyeknya dijadiin zip, terus dikirim via flash drive atau email ke satu sama lain? Ga selesai-selesai dong proyeknya?

Nah, Git ini ada untuk mengatasi hal itu. Git ini merupakan sebuah alat yang version control system. Dari namanya, mungkin kalian bisa menebak. Git adalah alat untuk mengatur atau mengontrol versi. Dengan begini, kalau ada revisi dalam proyek kalian, kalian nggak perlu capek-capek copy filenya terus paste lagi dan ngubah revisinya di file yang baru, tapi kalian cukup melakukan yang namanya commit sebagai semacam checkpoint atau tanda kalau revisi yang kalian kerjakan sudah selesai sebelum ngerjain revisi baru. Jadi kalau ternyata revisi setelahnya nggak diperlukan, kalian bisa ngelakuin rollback atau kembali ke revisi sebelumnya.

Kalian juga bisa melakukan merging atau penggabungan dari banyak perubahan yang dikerjakan orang berbeda-beda, jadi nggak perlu nunggu satu sama lain, tapi bisa kerja barengan dan nanti perubahannya tinggal digabungin deh sama Git! Praktis banget, kan? Mantep nih Git!

Rizza Mandasari, Rahman Hakim

Ilustrasi oleh Nazwa Mursyidan Baldan

Lalu, apa itu GitHub? Hmm, mari kita berpikir kayak gini. Kalian kan punya file manager ya di laptop kalian? Isinya ada folder-folder dan file-file kalian. Tapi, semuanya tersimpan secara lokal, hanya di laptop kalian saja, kan? Lalu, kalian tentunya pernah menggunakan Google Drive kan? Untuk melakukan backup file, lalu agar filenya bisa diakses bersama-sama. Google Drive ini adalah contoh cloud atau penyimpanan di “awan”, yang berarti kita menyimpan bukan secara lokal, namun pada server Google Drive.

Nah, Git dan GitHub ini kurang lebih seperti itu! Git merupakan software yang akan melakukan track kepada perubahan proyek kalian atau versioning, dan GitHub ini tempat menyimpan proyek kalian secara cloud atau di server GitHub ini sendiri. Nantinya, proyek kalian bisa diakses bersama-sama meski kalian berjauhan, dan juga kalian bisa melakukan backup proyek kalian di sini.

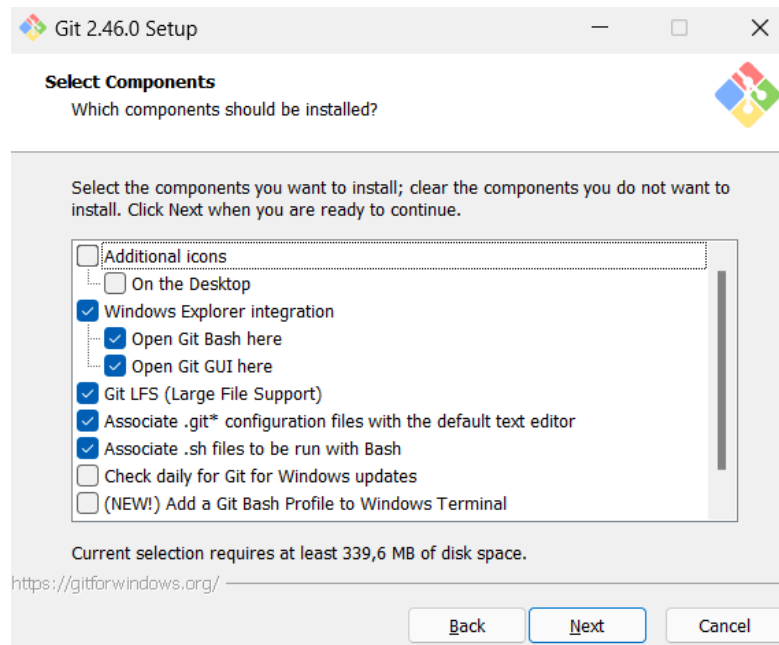
Kalau begitu, mari kita melakukan instalasi Git terlebih dahulu.

1.3.3.1 Windows

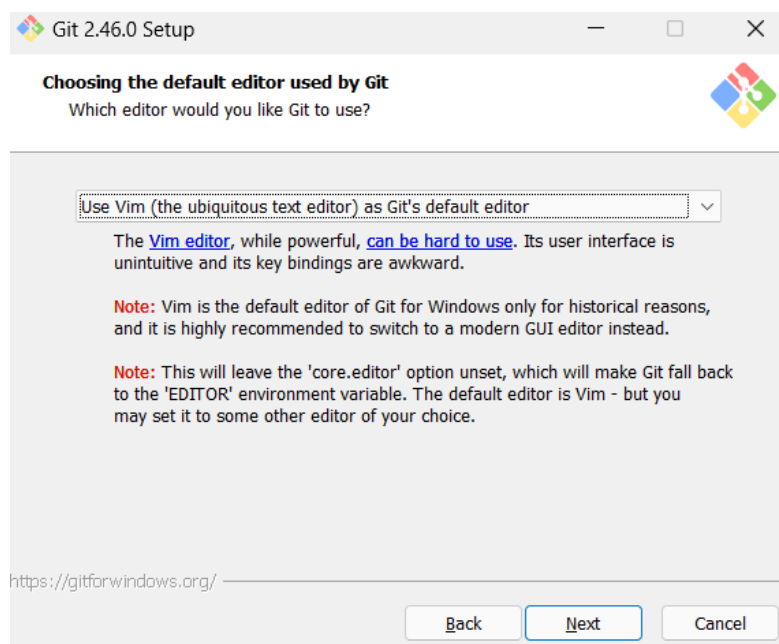
Untuk menginstall Git di Windows, pergilah ke laman <https://git-scm.com/download/win> kemudian pilihlah **64-bit Git for Windows Setup**. Setelah download selesai, bukalah file installer tersebut dan nantinya akan muncul halaman seperti berikut.



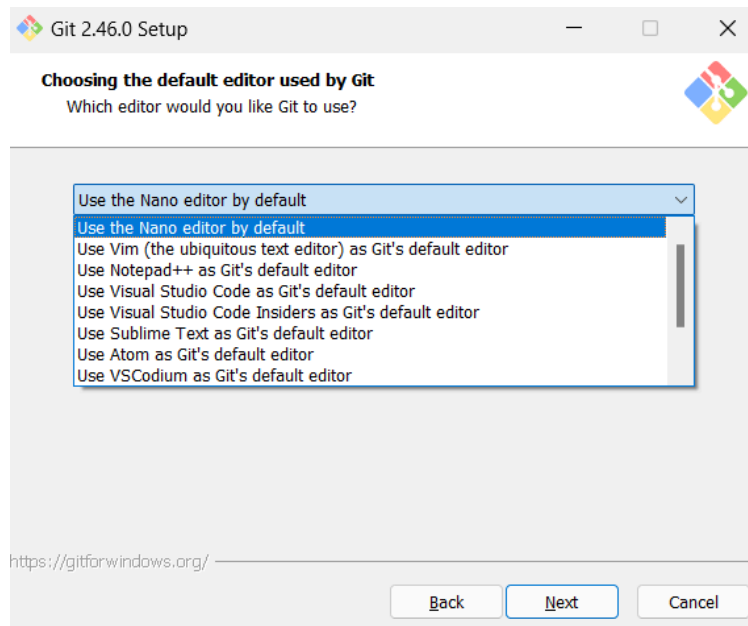
Setelah kalian klik “next”, nantinya kalian bisa memilih di folder mana Git akan diinstall. Biasanya, secara default akan diinstall pada “C:\Program Files\Git”. Namun, kalian bisa menggantinya sesuka kalian. Setelahnya, akan muncul menu berikut.



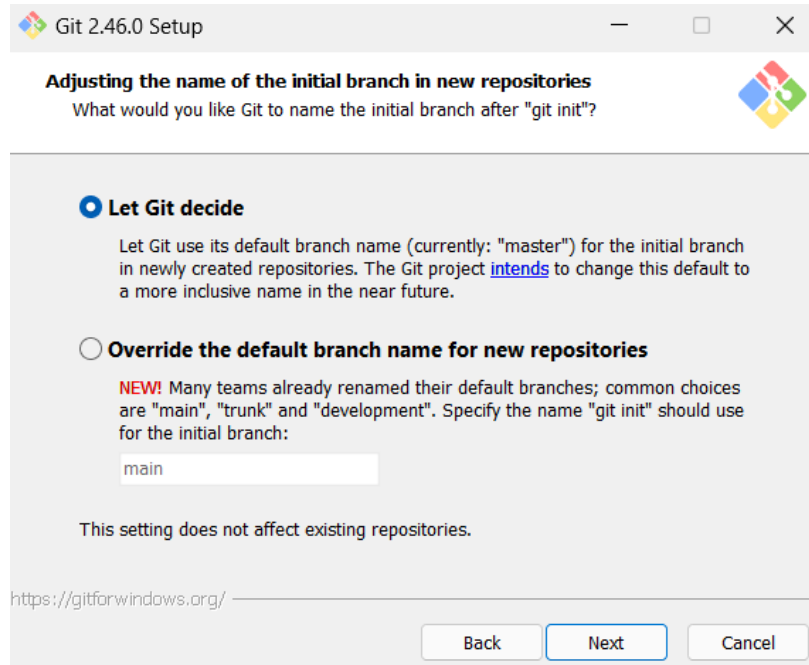
Tidak perlu mengubah apa-apa, kecuali “Additional icons”, kemudian “On the desktop” bila kalian ingin Git ada di halaman depan desktop komputer kalian. Bila tidak, maka tidak perlu diceklis. Setelahnya, akan muncul menu untuk memilih editor teks yang ingin kalian pakai untuk Git.



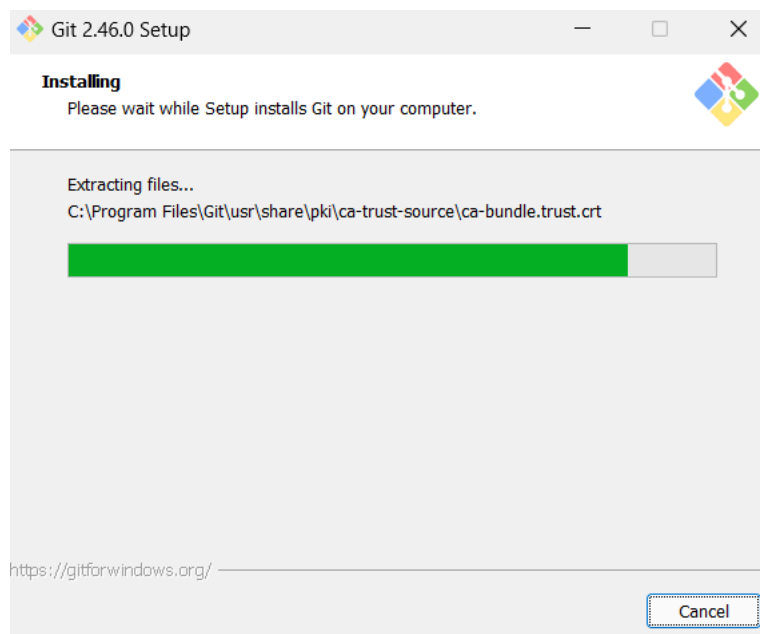
Di sini, kalian bebas memilih apa saja yang kalian inginkan. Namun, rekomendasi dariku dan Git adalah menggunakan editor teks “Nano” yang ringan dan mudah digunakan di terminal. Bila ingin mengganti, klik ikon drop down yang ada di pilihan, lalu pilih editor teks yang kalian ingin gunakan.

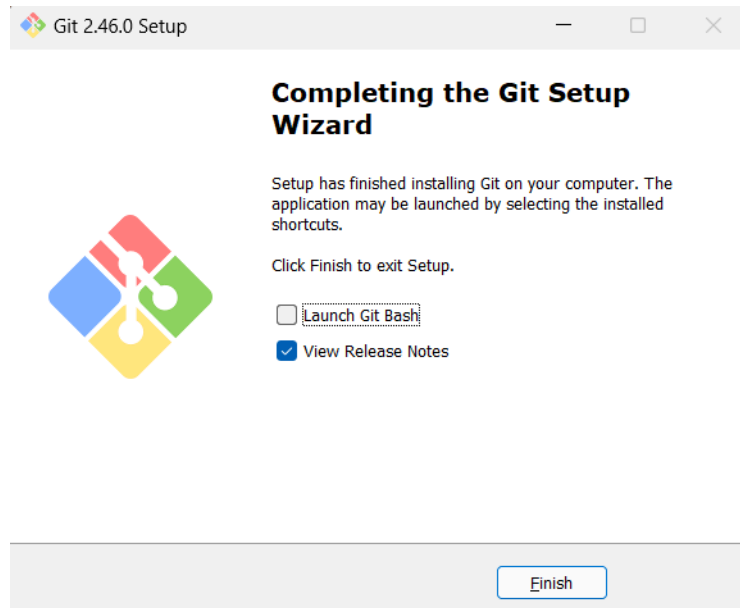


Lalu klik next. Setelahnya, akan ada pilihan untuk memilih nama “branch” utama default untuk Git. Branch adalah tempat dimana kalian nanti akan mengerjakan proyek kalian. Secara default, branch utama Git bernama “master”. Namun kalian bisa menggantinya menjadi yang lain, dan alternatif populer adalah “main”. Namun, untuk sekarang, tidak masalah kita membiarkan Git menggunakan “master”.

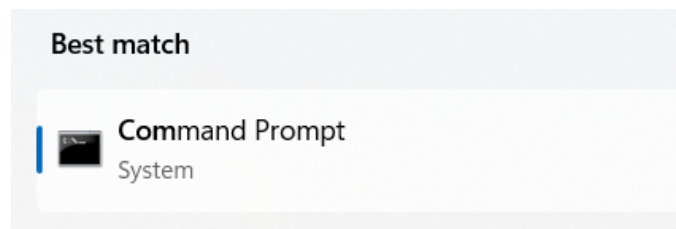


Kalian bisa langsung klik next. Sisanya, akan muncul halaman-halaman lain seperti **Adjusting Path Environment**, **Choosing the SSH executable**, **Choosing HTTPS transport backend**, dan sebagainya. Kalian bisa menggunakan default yang dipilihkan oleh Git di sini dan klik next hingga akhir. Lalu, setelah tombol install tersedia, langsung klik install saja kemudian finish bila sudah selesai.

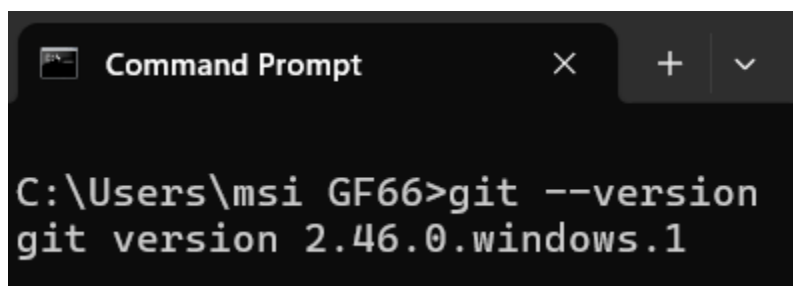




Dan Git pun sudah terinstall! Kalian bisa membuka command prompt dengan mengetik “cmd” kemudian klik “Command Prompt” di menu Windows search kalian, lalu buka command prompt tersebut.



Setelah command prompt terbuka, kalian bisa mengetik perintah “git --version” untuk mengecek apakah Git sudah terinstall dengan baik. Bila sudah, maka versi Git kalian akan ditampilkan.



Bila belum, maka cobalah untuk melakukan reinstall!

1.3.4 Konfigurasi Git

Sekarang adalah waktunya kalian melakukan konfigurasi Git. Dengan melakukan konfigurasi Git, kalian akan memberitahu Git siapakah diri kalian. Wow maknanya dalam sekali! Hahaha! Oke, tapi serius, karena nantinya ketika mengerjakan proyek, saat kalian melakukan perintah “commit”, nantinya nama dan email kalian akan tercatat sebagai orang yang telah melakukan perubahan-perubahan di proyek kalian tersebut. Karena itulah, kalian harus melakukan konfigurasi Git.

Bila kalian menggunakan laptop pribadi, untuk melakukan konfigurasi Git, kalian bisa menuliskan perintah berikut.

```
C:\Users\msi GF66>git config --global user.name "Nama Kalian"  
C:\Users\msi GF66>git config --global user.email "emailkalian@gmail.com"
```

Di sini, ada “flag” atau tanda “--global” yang berarti nama dan email kalian akan dikonfigurasi seperti ini untuk seluruh proyek dengan Git di komputer kalian. Namun, bila kalian menggunakan komputer orang lain atau komputer lab, maka hal seperti ini harus dihindari!

Penting!

Hindari penggunaan “--global” ketika menggunakan perangkat yang bukan milik kalian! Karena nantinya, orang lain yang menggunakan perangkat tersebut bisa-bisa malah tidak sengaja menggunakan nama kalian ketika mereka menggunakan Git untuk proyek mereka.

Kalian tidak perlu melakukan konfigurasi apa-apa bila belum ada proyek yang dikerjakan. Setelah kalian membuat folder proyek dan menginisialisasi Git, maka kalian bisa menggunakan perintah config sebelumnya tanpa flag “--global”. Dengan begini, konfigurasi hanya akan disimpan untuk proyek kalian sendiri. Dan orang lain tidak akan memakai nama kalian!



Dengan begini, konfigurasi Git telah selesai. Bila kalian memakai komputer lab, maka tidak perlu konfigurasi untuk sekarang. Konfigurasi Git akan dibahas lebih lanjut ketika kalian sudah membuat sebuah folder proyek baru, dan kemudian menggunakan perintah “git init”!