



Mata Kuliah : Pemrograman Web Lanjut (PWL)  
Program Studi : D4 – Teknik Informatika / D4 – Sistem Informasi Bisnis  
Semester : 4 (empat) / 6 (enam)  
Pertemuan ke- : 1 (satu)

## JOBSHEET 01

### PENGENALAN WEB FRAMEWORK LARAVEL

#### 1. Framework Laravel

Laravel adalah salah satu kerangka kerja (framework) PHP yang paling populer dan paling kuat untuk pengembangan web. Dirancang dengan tujuan untuk mempercepat proses pengembangan dengan menyediakan alat dan fitur yang kuat serta mudah digunakan.

Berikut adalah beberapa poin penting tentang Laravel:

- ✓ **Ekosistem yang Kuat:** Laravel memiliki ekosistem yang luas dan aktif, yang mencakup dokumentasi yang sangat baik dan komunitas yang bersemangat. Dokumentasi resmi Laravel sangat lengkap dan mudah dipahami, menyediakan panduan langkah demi langkah untuk mempelajari framework ini dari dasar hingga tingkat lanjutan. Komunitas Laravel juga sangat aktif, dengan banyak forum, grup diskusi, dan sumber daya online lainnya yang tersedia untuk mendapatkan bantuan dan berbagi pengetahuan.
- ✓ **Sintaksis yang Ekspresif:** Salah satu keunggulan utama Laravel adalah sintaksisnya yang ekspresif dan mudah dipahami. Dengan menggunakan konvensi yang konsisten dan kode yang bersih, Laravel memungkinkan pengembang untuk menulis kode dengan cepat dan efisien. Misalnya, menggunakan fitur seperti "fluent query builder" membuat penulisan kueri basis data menjadi lebih intuitif dan mudah dipelajari.
- ✓ **ORM Eloquent:** Eloquent adalah ORM yang disertakan dengan Laravel, yang memungkinkan pengembang untuk berinteraksi dengan basis data menggunakan objek PHP. Ini memungkinkan pengembang untuk menulis kode yang lebih bersih dan mudah dipahami, serta mengurangi kebutuhan untuk menulis kueri SQL secara langsung. Eloquent juga menyediakan fitur-fitur seperti relasi objek, pencarian, penyaringan, dan validasi data, yang membuat pengembangan aplikasi menjadi lebih mudah dan efisien.
- ✓ **Routing yang Kuat:** Laravel menyediakan sistem routing yang kuat dan mudah digunakan, yang memungkinkan pengembang untuk menentukan rute URL aplikasi mereka dengan mudah. Ini memungkinkan pengembang untuk mengarahkan



permintaan HTTP ke fungsi-fungsi yang sesuai, dan mengorganisasi kode aplikasi mereka dengan cara yang terstruktur dan mudah dipelihara.

- ✓ **Pengujian yang Terintegrasi:** Laravel menyediakan dukungan yang kuat untuk pengujian aplikasi (unit testing), dengan menyediakan alat dan kerangka kerja yang terintegrasi untuk menulis dan menjalankan tes otomatis. Ini memungkinkan pengembang untuk memastikan bahwa kode mereka berfungsi dengan benar dan terhindar dari regresi ketika mereka membuat perubahan. Laravel juga menyediakan fitur-fitur untuk menguji berbagai aspek aplikasi, termasuk pengujian kontroler, model, dan tampilan.
- ✓ **Templating Blade:** Laravel menggunakan sistem templating yang disebut Blade, yang memungkinkan pengembang untuk membuat tampilan aplikasi dengan mudah dan secara efisien. Blade menyediakan sintaks yang bersih dan intuitif untuk menulis tampilan HTML, serta fitur-fitur seperti warisan tata letak, loop, dan kondisional. Ini memungkinkan pengembang untuk membuat tampilan yang dinamis dan menarik dengan mudah.
- ✓ **Keamanan yang Terintegrasi:** Laravel menyediakan berbagai fitur keamanan yang terintegrasi, yang membantu melindungi aplikasi Anda dari berbagai serangan web yang umum. Misalnya, Laravel menyediakan proteksi terhadap serangan CSRF dengan menyediakan fitur otomatis untuk menghasilkan dan memeriksa token CSRF pada formulir HTML. Laravel juga menyediakan fitur untuk melindungi aplikasi dari serangan XSS, SQL Injection, dan serangan keamanan web lainnya.
- ✓ **Paket Bawaan:** Laravel dilengkapi dengan berbagai paket yang berguna untuk tugas-tugas umum dalam pengembangan web, seperti autentikasi pengguna, manajemen sesi, dan penjadwalan pekerjaan. Ini memungkinkan pengembang untuk membangun aplikasi dengan cepat dan efisien, tanpa perlu menulis ulang kode yang sama berulang-ulang.

Ini hanya sebagian kecil dari apa yang ditawarkan oleh Laravel sebagai framework pengembangan web. Dengan kombinasi fitur-fitur ini, Laravel memungkinkan pengembang untuk membangun aplikasi web yang kuat, efisien, dan mudah dipelihara.



## 2. Perangkat Pendukung

Sebelum kita mulai menggunakan Laravel untuk praktikum, pastikan sistem Anda memenuhi persyaratan berikut:

No	Sistem	Keterangan
1	Laragon - Web Server	Aplikasi untuk webserver, sama seperti XAMPP. Instal Komponen: <ul style="list-style-type: none"><li>- Apache Webserver</li><li>- MySQL</li><li>- PHP 8.2</li></ul>
		Link Download : <a href="#">Download Laragon</a>
2	PhpMyAdmin	Aplikasi manajemen basisdata MySQL/MariaDB berbasis website. Aplikasi ini bersifat <i>open-source</i> dan sudah banyak digunakan oleh pengembang web.
		Link Download: Bisa auto download dan <i>install</i> via Laragon
3	Text Editor (boleh pilih) <ul style="list-style-type: none"><li>1. VS Code</li><li>2. PHPStrom</li></ul>	Text editor yang digunakan untuk menulis kode program PHP. Untuk VSCode bersifat open-source jadi bisa digunakan secara bebas, sedangkan PHPStrom berbayar. Akan tetapi, untuk PHPStrom bisa dapat free license dengan registrasi pakai akun/email mahasiswa.
		Link Download : <ul style="list-style-type: none"><li>1. <a href="#">Download VSCode</a></li><li>2. <a href="#">Download PHPStrom</a></li></ul>
4	Composer	Composer adalah manajer dependensi untuk bahasa pemrograman PHP yang digunakan secara luas dalam pengembangan aplikasi web modern. Aplikasi ini dapat mengelola library PHP yang ingin kita gunakan saat pengembangan web.
		Link Download : <a href="#">Download Composer</a>
5	Notepad++	Aplikasi tambahan untuk second text editor
		Link Download : <a href="#">Download Notepad++</a>
6	GIT	Merupakan version control yang digunakan untuk mengelola versi, dan bertugas mencatat setiap perubahan pada file kode program yang kita tulis.
		Link Download : <a href="#">Download Git</a>
7	Browser (minimal 2)	Web browser untuk preview web yang sedang dikembangkan.
		Link Download :



	1. Firefox Developer Edition 2. Mozilla Firefox 3. Google Chrome	1. <a href="#">Firefox Developer Edition</a> 2. <a href="#">Mozilla Firefox</a> 3. <a href="#">Google Chrome</a>
8	Git Fork (opsional)	Aplikasi tambahan untuk aksi pada GIT (clone, pull, commit, push, dll). Sebenarnya sudah ada fitur serupa di text editor VSCode dan PHPStorm, tapi ini bisa untuk opsi lain selain pakai text editor Link Download : <a href="#">Download Git-Fork</a>

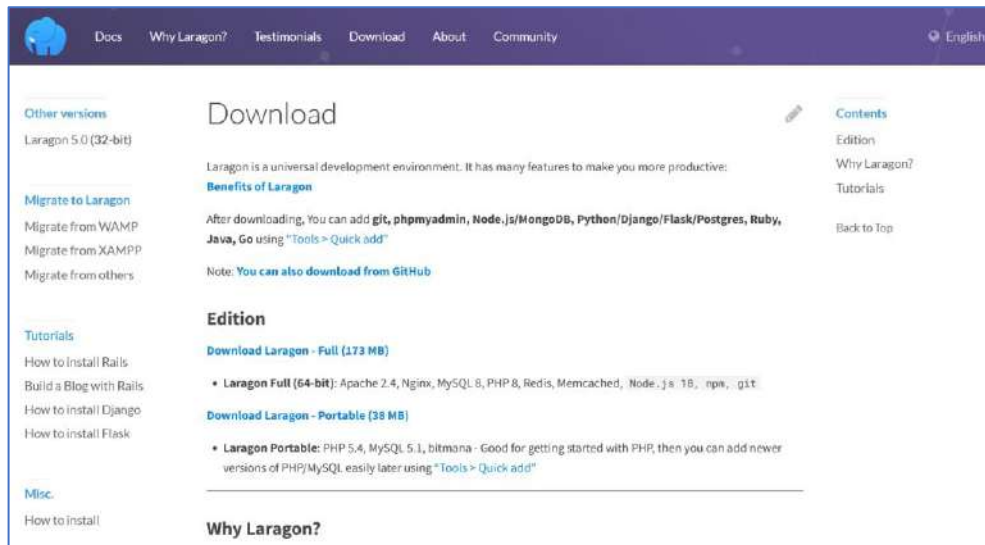
### 3. Video Cara Instalasi Perangkat Pendukung & Laravel

- Instalasi Laragon : <https://youtu.be/aYj87sGEtv8>
- Instalasi phpMyAdmin : <https://youtu.be/ESMkGvoER2w>
- Instalasi VS Code : <https://youtu.be/-ap4lWD4b1A>
- Instalasi Composer : [https://youtu.be/juRRp\\_6lDzg](https://youtu.be/juRRp_6lDzg)
- Instalasi Git : <https://youtu.be/Q4ucu9wwyTk>
- Instalasi Laravel : <https://youtu.be/KRw6ZyeRxtM>
- Publish dan Sinkronisasi ke Github : <https://youtu.be/hrfTi7v0JFY>

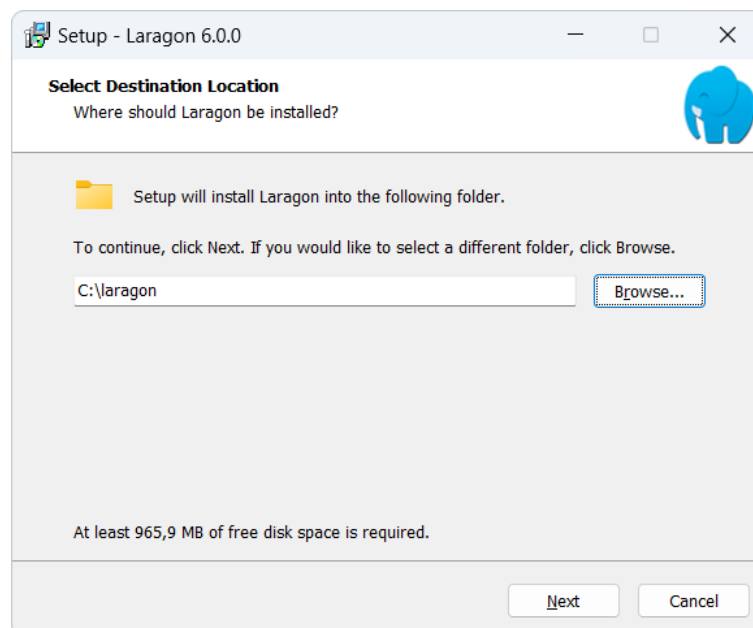
### 4. Instalasi Laragon

Pada perkuliahan praktikum ini, kita menggunakan Laragon sebagai web server utama kita dalam menjalankan program Laravel kita nantinya. Berikut langkah-langkah instalasi Laragon

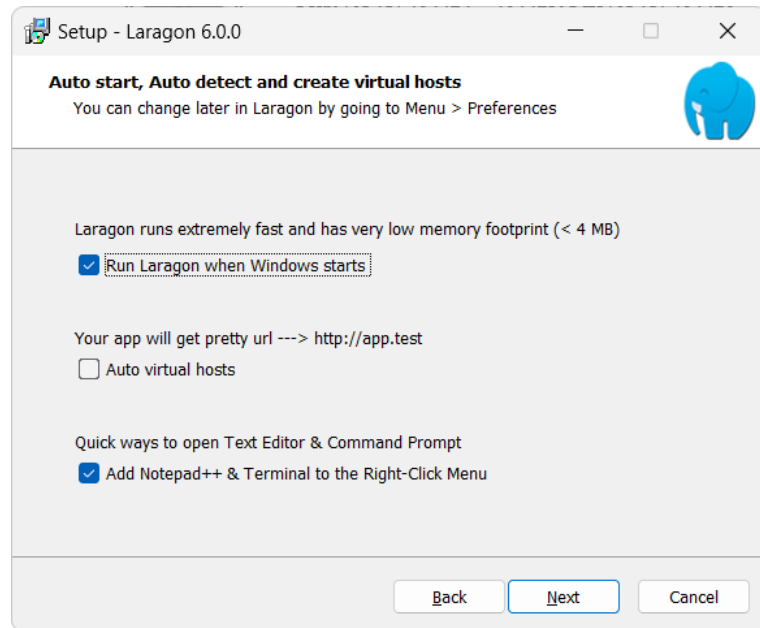
- Klik link download aplikasi laragon. [Download Laragon](#)
- Saat sudah masuk ke dalam website laragon, pilih opsi **Download Laragon – Full** dan download aplikasi tersebut



- c. Setelah download selesai, jalankan file laragon tersebut
- d. Pilih letak folder instalasi laragon (biasanya di drive C/D), lalu klik *next*



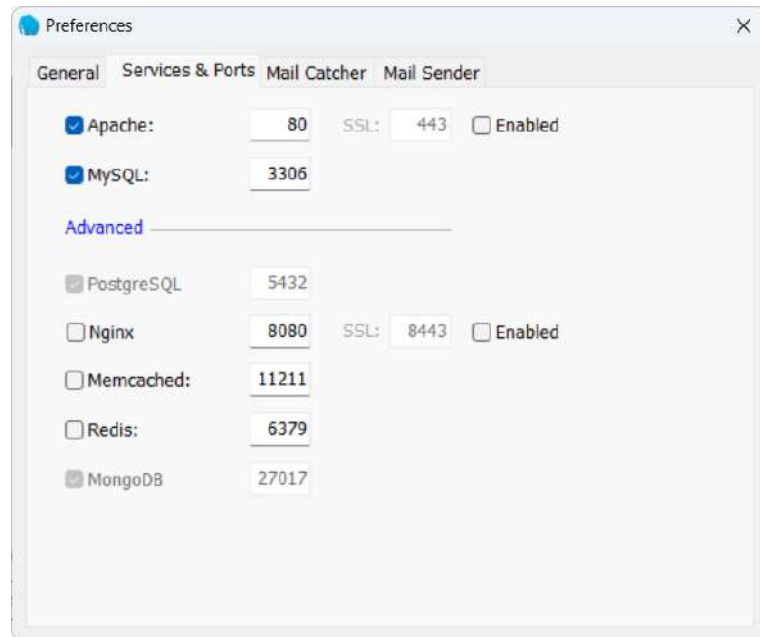
- e. Untuk opsi “Run laragon when windows starts” bisa kita centang, agar laragon bisa dijalankan saat laptop kita menyala.



- f. Kemudian klik *next*, dan klik *install*
- g. Setelah di-*install*, kita buka aplikasi laragon, dan kita klik *setting*



- h. Kita pilih *tab Services & Ports*, dan pastikan **centang** service *Apache* pada *port 80*, dan *MySQL* pada *port 3306*.





- i. Close dialog di atas, dan klik **start all** untuk menjalankan webserver dan database

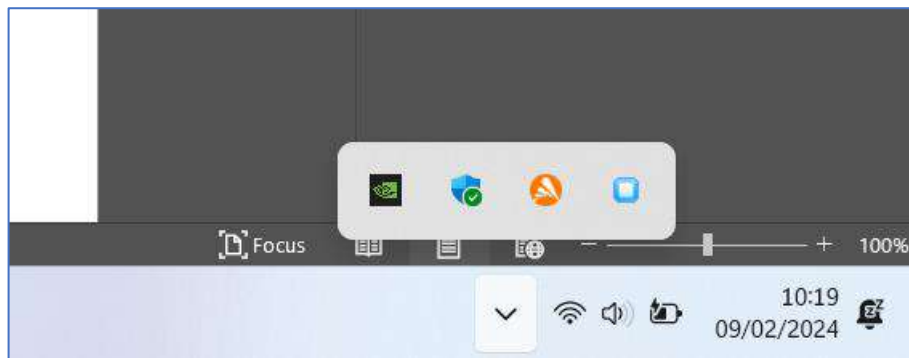




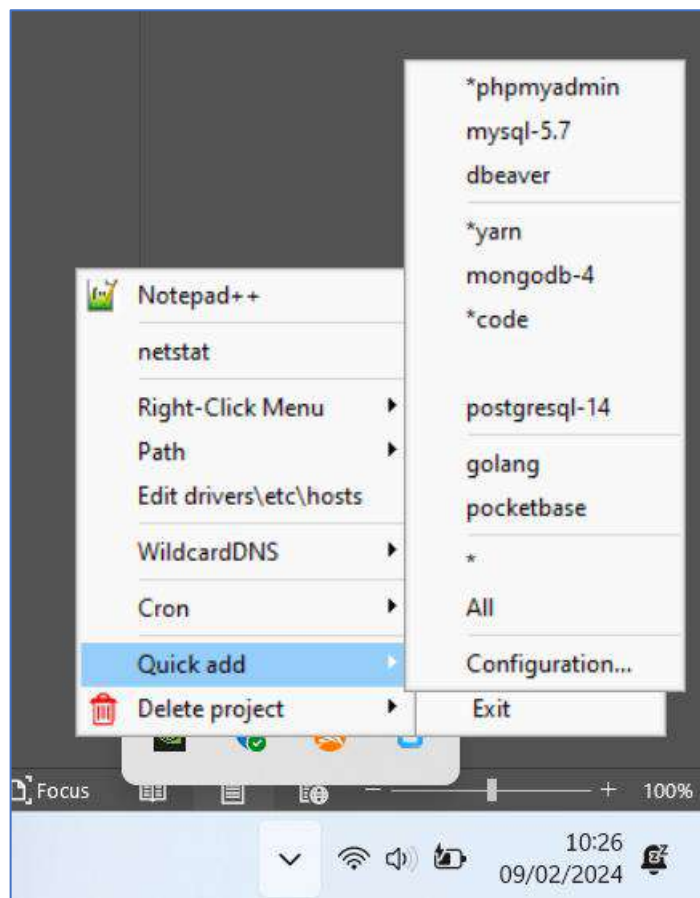
## 5. Instalasi PhpMyAdmin

Phpmyadmin kita gunakan untuk mengelola basisdata MySQL/MariaDB yang akan kita gunakan nantinya. Berikut langkah-langkah instalasi Phpmyadmin via Laragon

- Buka taskbar pada pojok kanan bawah dan perhatikan icon (  laragon saat belum dijalankan /  laragon saat sudah dijalankan) . Icon tersebut adalah icon untuk aplikasi laragon.

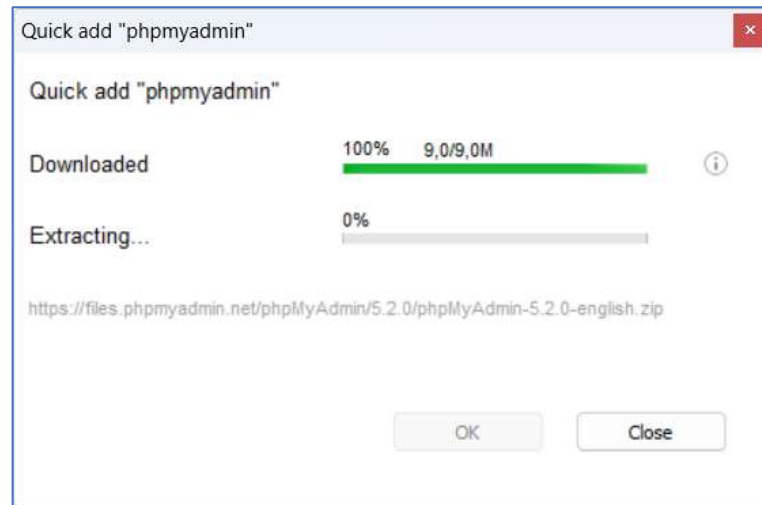


- Klik kanan pada icon laragon, kemudian klik **tools** → **Quick add** → **phpmyadmin**



- Maka akan muncul dialog laragon men-*download* phpMyAdmin

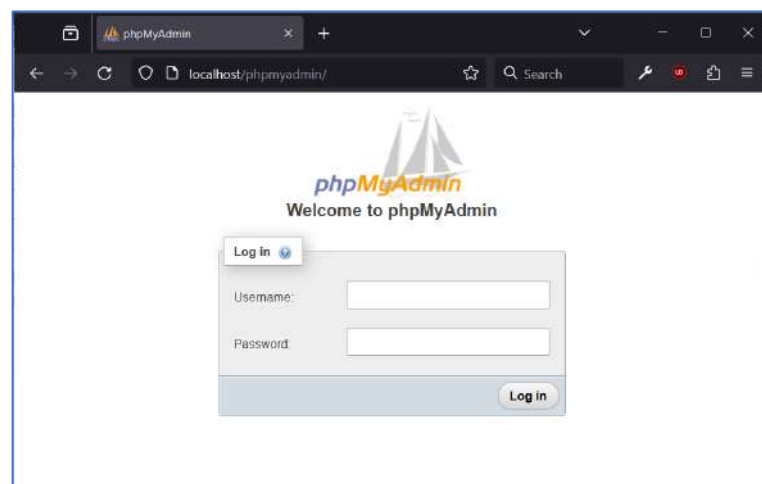




- d. Setelah selesai di download, aplikasi phpMyAdmin akan disimpan pada folder `C:\laragon\etc\apps\phpMyAdmin`
- e. Buka folder `phpMyAdmin` dan copy file `config.sample.inc.php` menjadi `config.inc.php` dan edit file `config.inc.php` tersebut seperti gambar berikut

```
29  /* Server parameters */
30  $cfg['Servers'][$i]['host'] = 'localhost';
31  $cfg['Servers'][$i]['compress'] = false;
32  $cfg['Servers'][$i]['AllowNoPassword'] = true;
```

- f. Setelah selesai diedit, simpan perubahan, dan kita coba buka aplikasi phpMyAdmin di browser dengan mengakses halaman `localhost/phpMyAdmin`



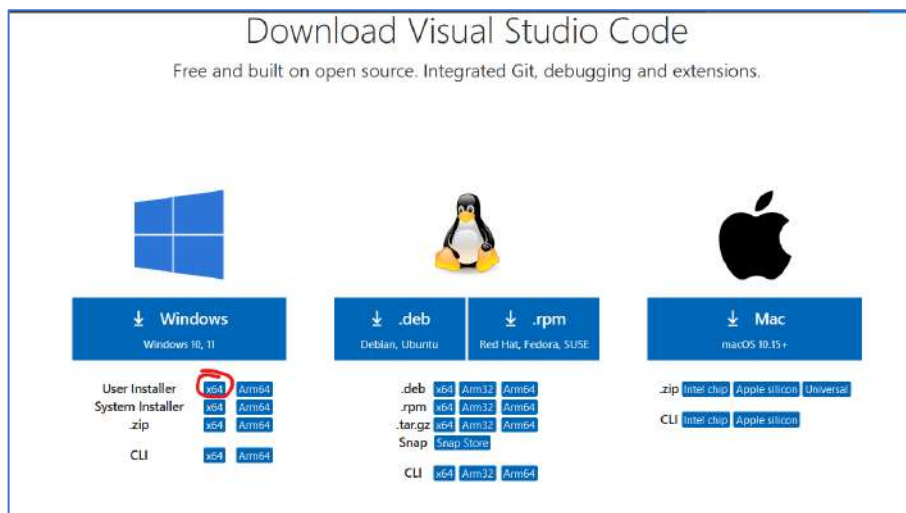
- g. Secara *default*, akun untuk masuk ke database MySQL/MariaDB adalah username `root` dan password dikosongi



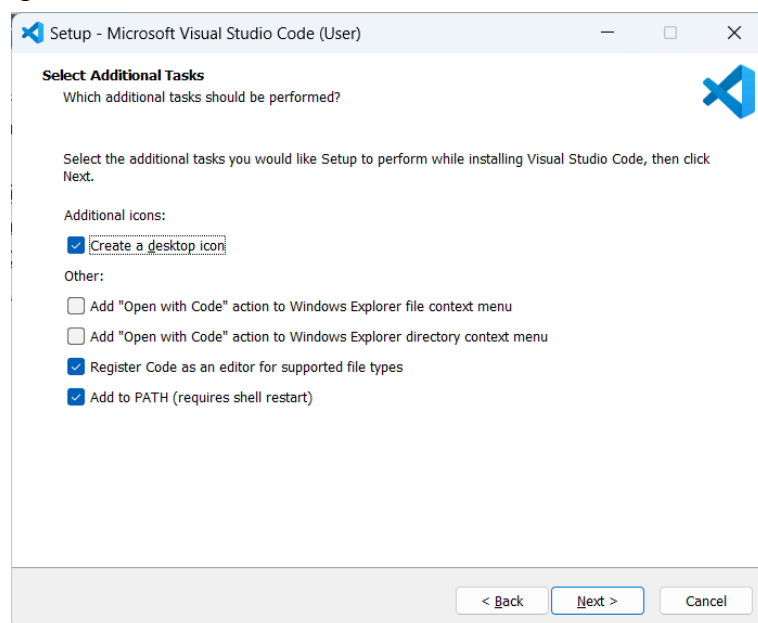
## 6. Instalasi Visual Studio Code

Visual Studio Code atau yang sering kita sebut VSCode adalah editor kode program yang sangat populer yang dikembangkan oleh Microsoft. VScode merupakan *text editor* yang kuat dan ringan yang dirancang untuk pengembangan aplikasi lintas platform. Berikut langkah-langkah instalasi VSCode

- Klik link [Download VSCode](#), dan untuk pengguna sistem operasi windows, pilih yang dilingkari merah



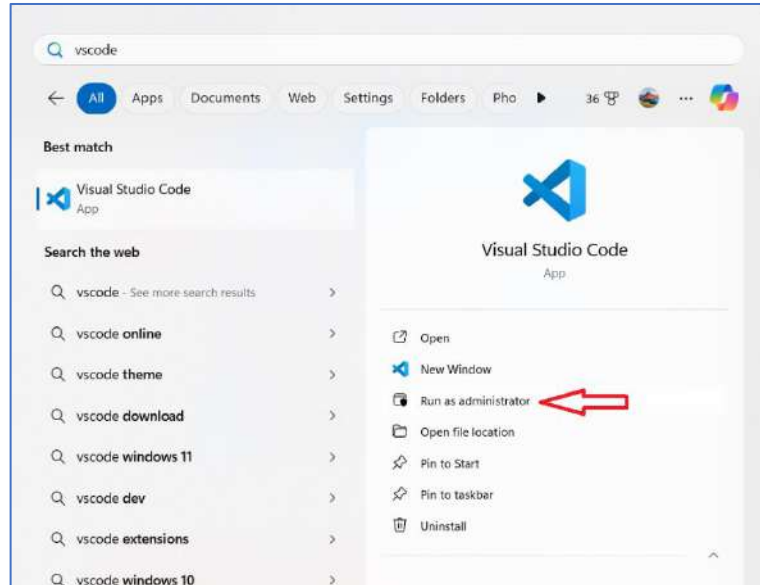
- Jika sudah selesai download, jalankan file tersebut
- Pilih opsi yang sesuai



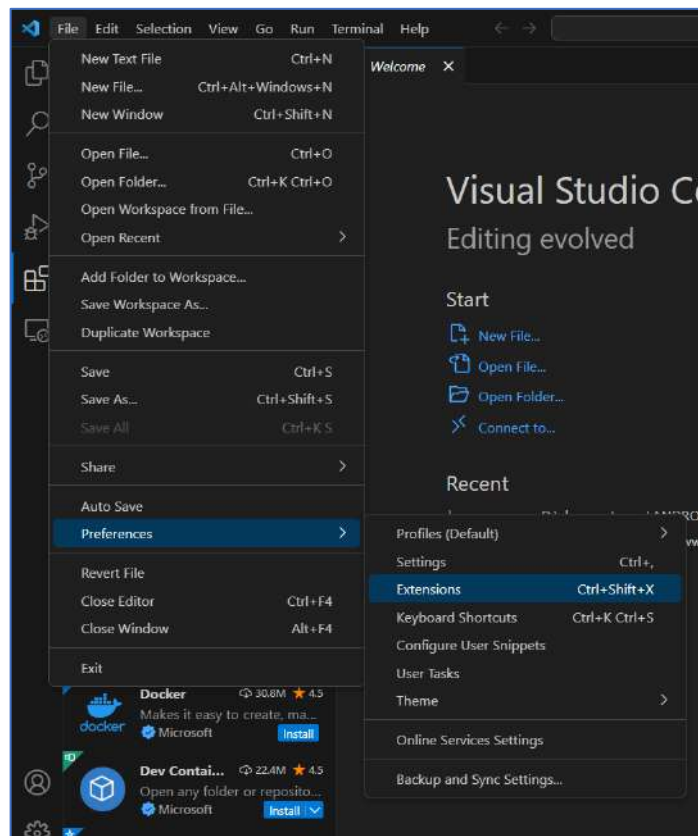
- Kemudian lanjutkan proses *install* VSCode



- e. Setelah selesai proses intall VSCode, jalankan program VSCode secara administrator (*run as administrator*)

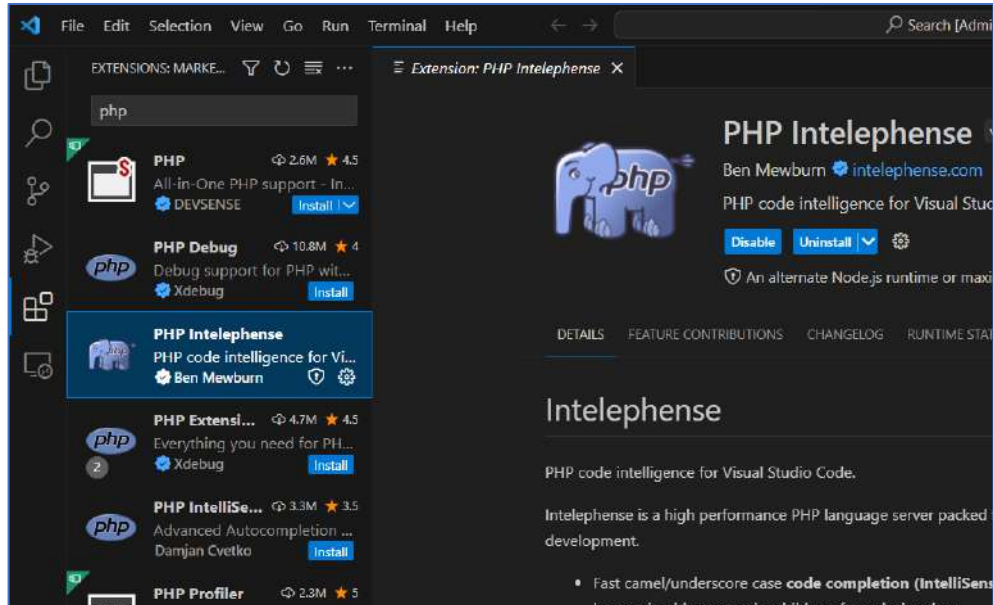


- f. Disini kita akan menyiapkan plugin/extension agar lebih mudah dalam menulis kode program untuk Laravel
- g. Klik **File** → **Preferences** → **Extensions**

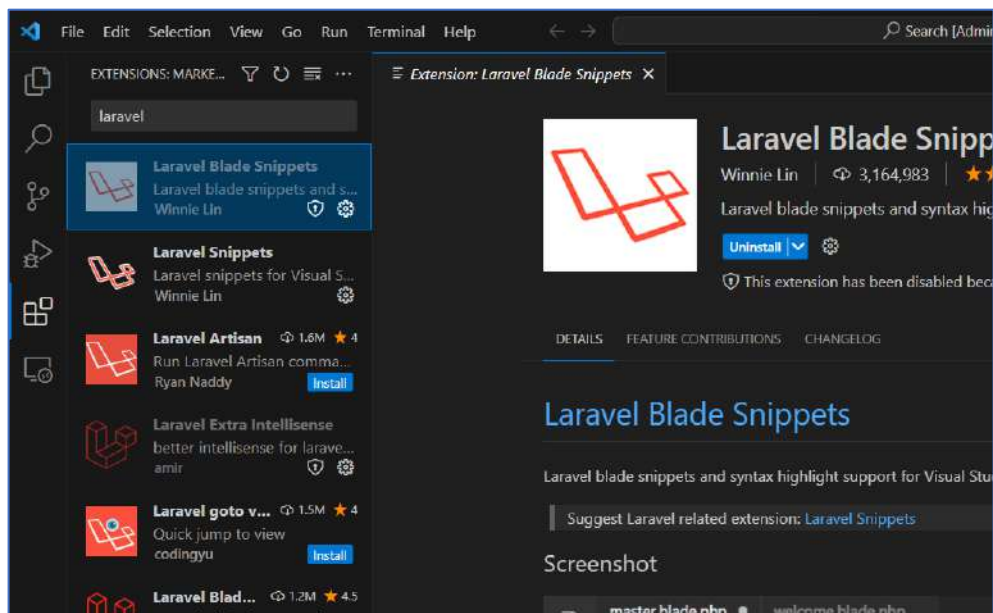




- h. Kemudian kita ketikkan pencarian ekstensi yang mempermudah kita dalam menulis kode program PHP. Kita tulis keyword “php”



- i. Kita pilih dan *install* ekstensi **PHP Intelephense** yang berguna untuk mempermudah kita koding PHP.
- j. Kita juga bisa mencari ekstensi agar memudahkan dalam menulis kode program PHP, khususnya untuk Laravel. Kita tulis keyword “Laravel”



- k. Kita bisa pilih dan *install* ekstensi **Laravel Blade Snippets** dan **Laravel Extra Intellisense** untuk mempermudah koding kita. Kalian juga bisa explore ekstensi lain sesuai selera kalian.

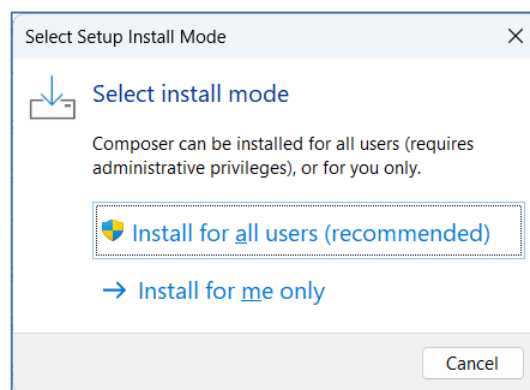


1. Untuk instalasi Text Editor PHPStrom tidak jauh beda dengan VSCode, jadi bisa dicoba sendiri

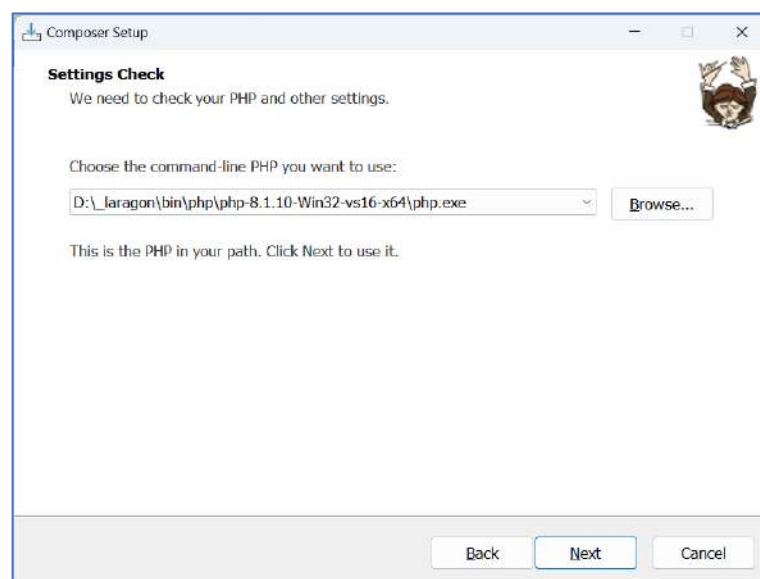
## 7. Instalasi Composer

Composer adalah manajer dependensi untuk bahasa pemrograman PHP yang memungkinkan pengembang untuk mengelola dan mengatur dependensi proyek secara efisien. Ini adalah alat yang sangat berguna dalam pengembangan perangkat lunak PHP modern, memungkinkan pengguna untuk menentukan dan menginstal pustaka, kerangka kerja (framework), dan paket-paket PHP lainnya yang diperlukan. Berikut adalah Langkah-langkah instalasi Composer

- a. Klik link [Download Composer](#)
- b. Jalankan file composer yang sudah di download, jika terdapat dialog [Select Setup Install Mode](#), pilih [Install for all users](#)

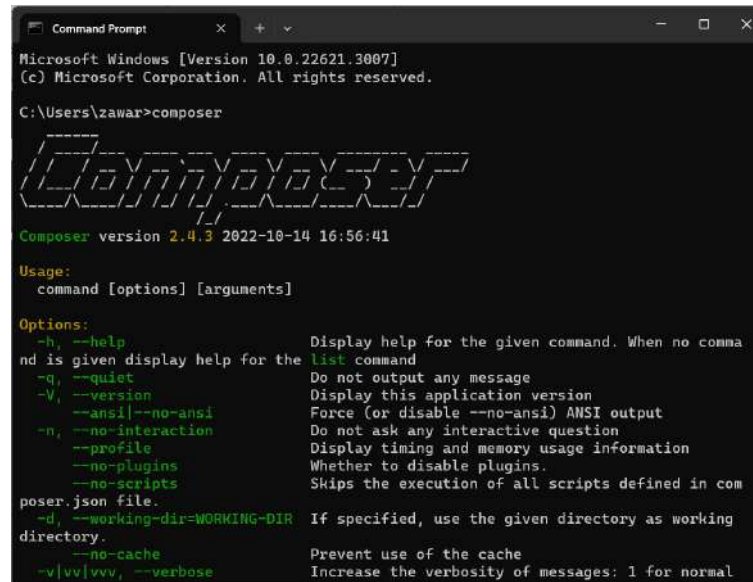


- c. Klik *next*, dan akan muncul dialog untuk memilih lokasi file [php.exe](#) berada. Pastikan file [php.exe](#) yang dipilih berada di dalam folder instalasi laragon.





- d. Sekarang kita coba cek pada `command prompt` dengan mengetikkan perintah `“composer”`. Jika muncul respon seperti di bawah ini, berarti composer sudah berhasil di *install*



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

C:\Users\zawar>composer

Composer version 2.4.3 2022-10-14 16:56:41

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display help for the given command. When no comma
nd 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) 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 com
poser.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
```

## 8. Instalasi Notepad++

Notepad++ adalah teks editor ringan yang bisa digunakan untuk menulis/membaca kode program. Aplikasi ini begitu ringan, dan bisa menjadi opsi untuk digunakan sebagai second text editor. Tidak ada setingan khusus pada aplikasi ini sehingga mudah untuk di install. Berikut link download aplikasi [notepad++](#).

## 9. Instalasi GIT

Git adalah software untuk melakukan version control terhadap kode program yang kita buat. Git diperlukan untuk mengelola kode program dan mempermudah kita dalam berkolaborasi dan bekerja sama ketika mengerjakan kode program dalam kelompok. Berikut Langkah-langkah untuk melakukan instalasi git pada sistem operasi windows.

- a. Pertama, kita periksa dulu apakah sistem operasi kita sudah support GIT atau belum dengan mengetikkan perintah `“git”` pada command prompt





```
Command Prompt

C:\Users\zawar>git
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
        [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
        [-p | --paginate] [-P | --no-pager] [--no-replace-objects] [--bare]
        [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
        [--super-prefix=<path>] [--config-env=<name>=<envvar>]
        <command> [<args>]

These are common Git commands used in various situations:


start a working area (see also: git help tutorial)
clone      Clone a repository into a new directory
init       Create an empty Git repository or reinitialize an existing one


work on the current change (see also: git help everyday)
add        Add file contents to the index
mv         Move or rename a file, a directory, or a symlink
restore    Restore working tree files
rm         Remove files from the working tree and from the index


examine the history and state (see also: git help revisions)
bisect     Use binary search to find the commit that introduced a bug
diff       Show changes between commits, commit and working tree, etc
grep       Print lines matching a pattern
log        Show commit logs
show       Show various types of objects
status     Show the working tree status


grow, mark and tweak your common history
```

- b. Jika keluar output seperti gambar di atas maka **git** sudah ter-*install* pada sistem operasi kita dan kita bisa lanjut ke Tahap 8 (*skip* tahapan instalasi Git ini). Jika keluar notif **git is not recognized..** seperti gambar di bawah ini, maka kita lanjut ke langkah berikutnya

```
Administrator: Command Prompt

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

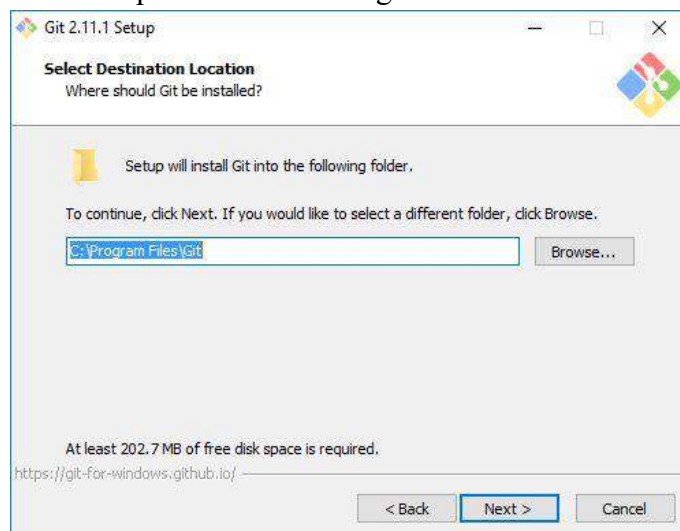
C:\Users\WDAGUtilityAccount>git
'git' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\WDAGUtilityAccount>https://notepad-plus-plus.org/downloads/
```

- c. Klik link [Download Git](https://notepad-plus-plus.org/downloads/)
- d. Jalankan file git yang telah di download, dan akan muncul dialog seperti berikut.



- e. Selanjutnya klik *next* dan pilih lokasi *install* git



- f. Selanjutnya pilih komponen yang dibutuhkan dan pilihan pilihan selanjutnya jika anda tidak paham gunakan pilihan *default* dan klik *next*. Diperbolehkan mencoba opsi instal yang lain namun jika terjadi hal yang tidak diinginkan silahkan *uninstall* dan *install* lagi dengan pilihan *default*. Berikut ini gambar pilihan pilihan *default* installer *git* yang dapat kita ikuti.





The image displays the Git 2.11.1 Setup Wizard, a series of six windows guiding the user through the installation process on Windows.

- Step 1: Adjusting your PATH environment**  
How would you like to use Git from the command line?  
  - ☐ Use Git from Git Bash only  
This is the safest choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.
  - ☒ Use Git from the Windows Command Prompt  
This option is considered safe as it only adds some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from both Git Bash and the Windows Command Prompt.
  - ☐ Use Git and optional Unix tools from the Windows Command Prompt  
Both Git and the optional Unix tools will be added to your PATH.  
**Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.**
- Step 2: Configuring the line ending conversions**  
How should Git treat line endings in text files?  
  - ☒ Checkout Windows-style, commit Unix-style line endings  
Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").
  - ☐ Checkout as-is, commit Unix-style line endings  
Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").
  - ☐ Checkout as-is, commit as-is  
Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").
- Step 3: Configuring the terminal emulator to use with Git Bash**  
Which terminal emulator do you want to use with your Git Bash?  
  - ☐ Use MinTTY (the default terminal of MSYS2)  
Git Bash will use MinTTY as terminal emulator, which sports a resizable window, non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via "winpty" to work in MinTTY.
  - ☒ Use Windows' default console window  
Git will use the default console window of Windows ("cmd.exe"), which works well with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.
- Step 4: Configuring extra options**  
Which features would you like to enable?  
  - ☒ Enable file system caching  
File system data will be read in bulk and cached in memory for certain operations ("core.fsmonitor" is set to "true"). This provides a significant performance boost.
  - ☒ Enable Git Credential Manager  
The [Git Credential Manager](#) for Windows provides secure Git credential storage for Windows, most notably multi-factor authentication support for Visual Studio Team Services and GitHub. (requires .NET framework v4.5.1 or later).
  - ☐ Enable symbolic links  
Enable [symbolic links](#) (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.
- Step 5: Configuring experimental options**  
Which bleeding-edge features would you like to enable?  
  - ☐ Enable experimental, builtin difftool  
Use the experimental builtin diff tool (fast, but only lightly tested).
- Step 6: Installing**  
Please wait while Setup installs Git on your computer.  
Extracting files...  
C:\Program Files\Git\mingw64\lib\locale\America\Antigua
- Step 7: Completing the Git Setup Wizard**  
Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.  
Click Finish to exit Setup.  
  - ☐ Launch Git Bash
  - ☒ View Release Notes



- g. Selamat kita sudah berhasil melakukan *install* git pada sistem operasi windows. Langkah selanjutnya adalah melakukan konfigurasi terhadap git agar sesuai dengan akun github yang kita miliki.
- h. Bukalah terminal kemudian ketikkan perintah `git config --list`, maka output terminal akan seperti berikut

```
Command Prompt
C:\Users\zawar>git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
core.editor="C:\\Program Files\\Notepad++\\notepad++.exe" -multiInst -notabbar -no
session -noPlugin
pull.rebase=false
credential.helper=manager-core
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.email=zawaruddin@polinema.ac.id
user.name=zawaruddin

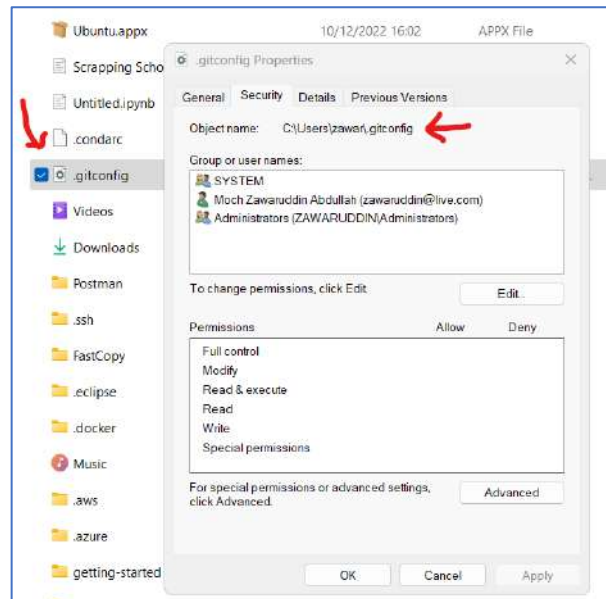
C:\Users\zawar>
```

- i. Perhatikan dua baris terakhir dalam konfigurasi tersebut. Perhatikan pada opsi `user.name` dan `user.email` *field* ini adalah *field* yang menentukan nama pengguna dan *email git* yang akan digunakan pada saat menyimpan perubahan atau *publish* ke github. **Pastikan opsi tersebut sudah sesuai dengan nama dan email kita yang kita daftarkan di github.**
- j. Jika belum di *set* nilai dari `user.name` dan `user.email` dapat di set dengan menggunakan perintah berikut ini :

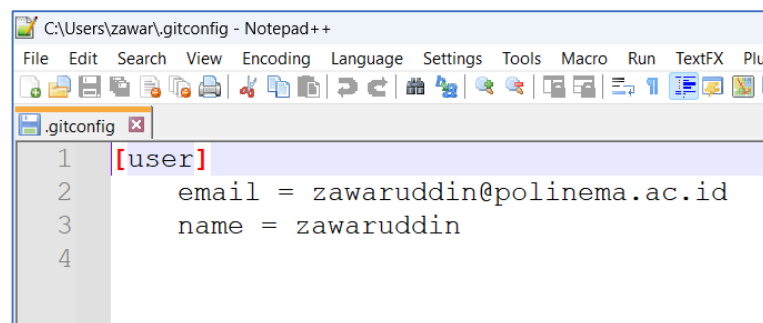
```
git config --global user.name = usernameAnda
git config --global user.email = emailAnda
```

Atau alternatif edit file secara langsung di Windows

```
C:/users/<nama_akun_komputer>/ .gitconfig
```



- k. Klik kanan pada file .gitconfig menggunakan teks editor notepad++, dan isi sesuai data akun github kalian



## 10. Instalasi Web Browser

Web browser digunakan untuk membuka dan melihat aplikasi website yang kita buat. Setidaknya kita membutuhkan minimal 2 (dua) browser untuk menguji tampilan aplikasi website kita, dan memastikan bahwa tampilannya sudah sesuai dengan standar dari masing-masing browser tersebut. Tidak ada setingan khusus untuk instalasi web browser, sehingga bisa kalian coba *install* aplikasi web browser sendiri.

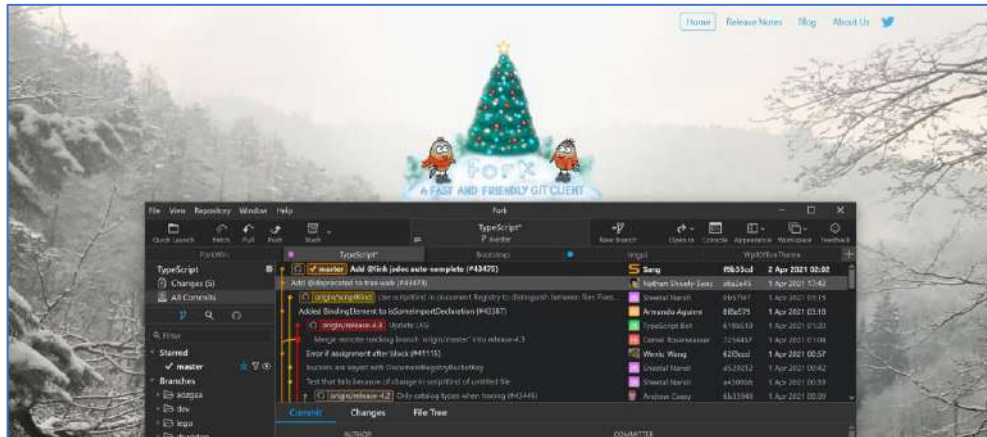
## 11. Instalasi Git-Fork

Git-Fork adalah aplikasi desktop yang digunakan untuk mengelola repositori Git dengan antarmuka pengguna grafis (GUI). Aplikasi Git-Fork memungkinkan pengguna untuk melakukan berbagai operasi Git, seperti menyalin (*cloning*) repositori, membuat cabang (*branch*), melakukan komit (*commit*), menyelesaikan konflik pada kode program, melakukan penggabungan kode program (*merge*), dan banyak lagi, semua dengan cara yang lebih intuitif

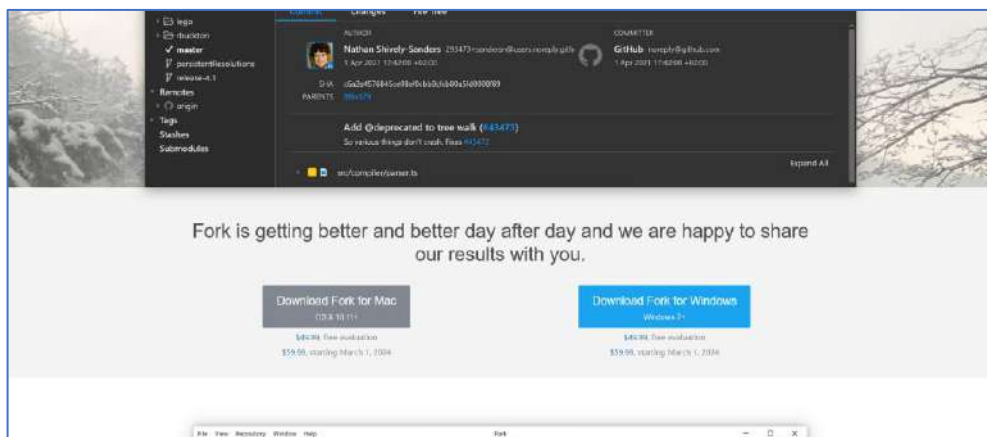


(berbasis GUI) dan mudah dipahami daripada menggunakan perintah Git melalui *terminal*. Berikut langkah-langkah menginstal git-fork

- a. Buka link [download Git-Fork](#)



- b. Download file git-fork sesuai dengan sistem operasi kalian

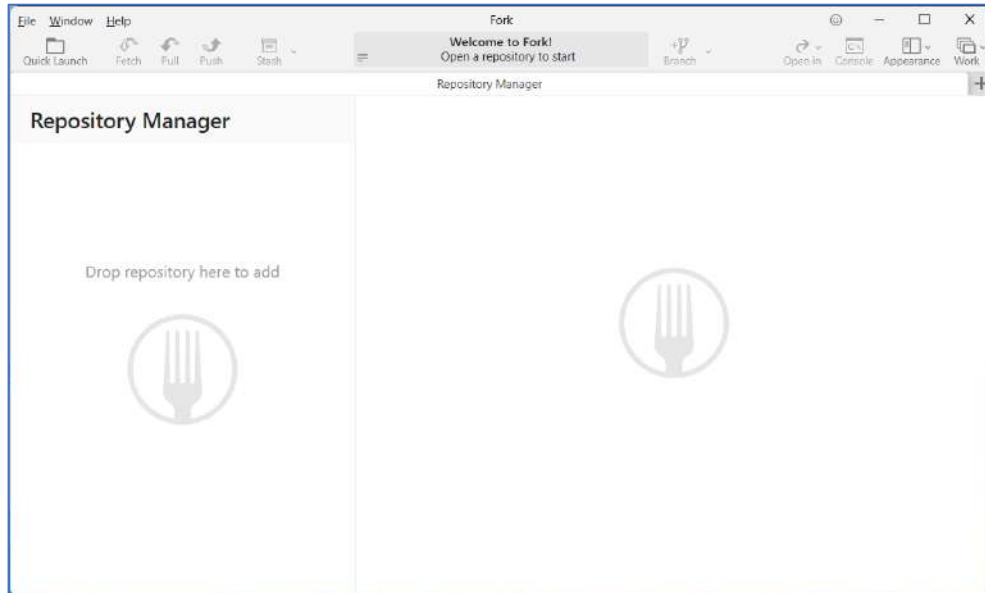


- c. Selanjutnya jalankan file yang sudah berhasil di download, dan muncul dialog seperti berikut

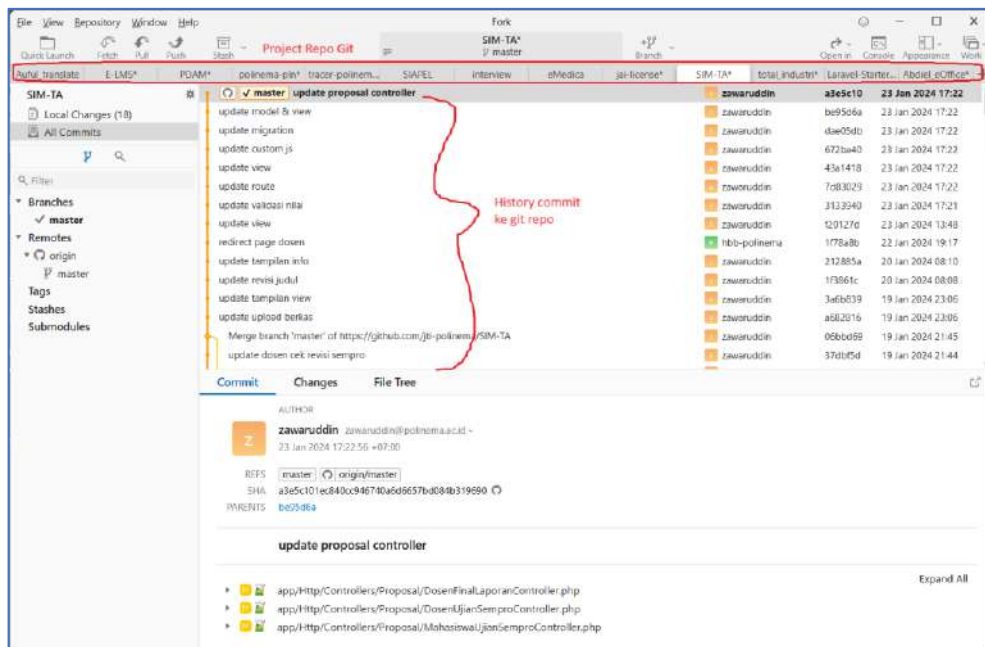




- d. Silahkan isi user name (nama lengkap kalian) dan email kalian yang terdaftar pada github.
- e. Kemudian klik tombol finish.
- f. Berikut adalah tampilan aplikasi git-fork setelah berhasil di *install*



- g. Dan berikut adalah contoh aplikasi git-fork untuk membuka banyak project git



## 12. Instalasi Laravel

Laravel dapat di-*install* dengan beberapa cara. Pada praktikum ini akan dijelaskan bagaimana cara melakukan instalasi laravel dengan menggunakan *command proamp/terminal*. Berikut langkah langkah instalasi laravel yang akan dilakukan





- a. Untuk keperluan *development*, Web server yang kita gunakan adalah **laragon**, jadi jalankan dan aktifkan service yang kita butuhkan di laragon (apache dan mysql)



- b. Buka *terminal*/CMD, dan arahkan ke direktori `C:\laragon\www`  
c. Ketikkan command berikut untuk menginstal Laravel spesifik versi 10

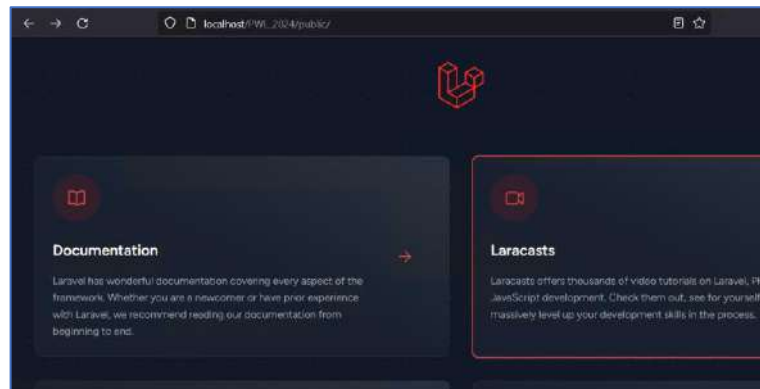
```
composer create-project laravel/laravel="10.3.*" PWL_2024
```

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

C:\Users\WDAGUtilityAccount>cd C:\laragon\www

C:\laragon\www>composer create-project laravel/laravel="10.3.*" PWL_2024
Creating a "laravel/laravel=10.3.*" project at "./PWL_2024"
Installing laravel/laravel (v10.3.2)
- Installing laravel/laravel (v10.3.2): Extracting archive
Created project in C:\laragon\www\PWL_2024
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 111 installs, 0 updates, 0 removals
- Locking brick/math (0.11.0)
- Locking carbonphp/carbon-doctrine-types (2.1.0)
- Locking dflydev/dot-access-data (v3.0.2)
- Locking doctrine/inflector (2.0.9)
- Locking doctrine/lexer (3.0.1)
- Locking dragonmantank/cron-expression (v3.3.3)
- Locking egulias/email-validator (4.0.2)
- Locking fakerphp/faker (v1.23.1)
- Locking filp/whoops (2.15.4)
- Locking fruitcake/php-cors (v1.3.0)
- Locking graham-campbell/result-type (v1.1.2)
- Locking guzzlehttp/guzzle (7.8.1)
- Locking guzzlehttp/promises (2.0.2)
- Locking guzzlehttp/psr7 (2.6.2)
- Locking guzzlehttp/uri-template (v1.0.3)
- Locking hamcrest/hamcrest-php (v2.0.1)
```

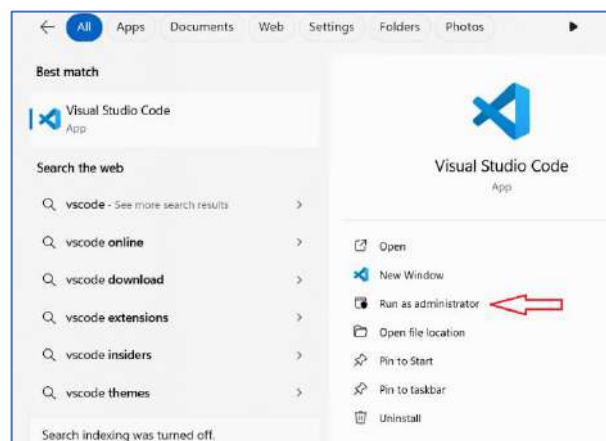
- d. Jika sudah selesai proses instalasi Laravel via composer, kita coba buka di browser dengan mengetikkan `localhost/PWL_2024/public`  
e. Berikut adalah tampilan awal aplikasi laravel yang telah kita install



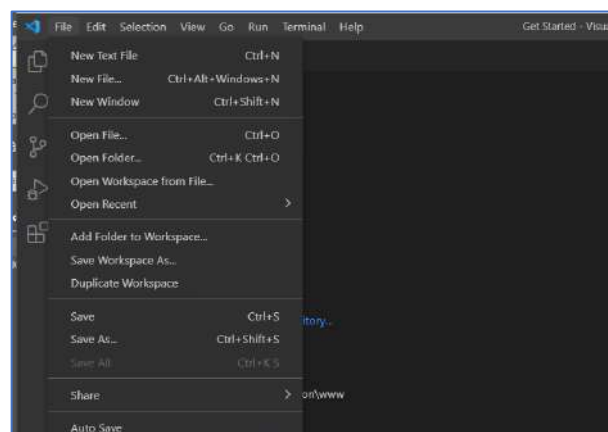
### 13. Publish Project Laravel ke Github

Aplikasi/project Laravel yang sudah kita buat bisa kita *publish/commit* ke github menggunakan Visual Studio Code (VSCode). Berikut langkah-langkah publish project Laravel ke github.

- a. Buka aplikasi VSCode dan *running as administrator*

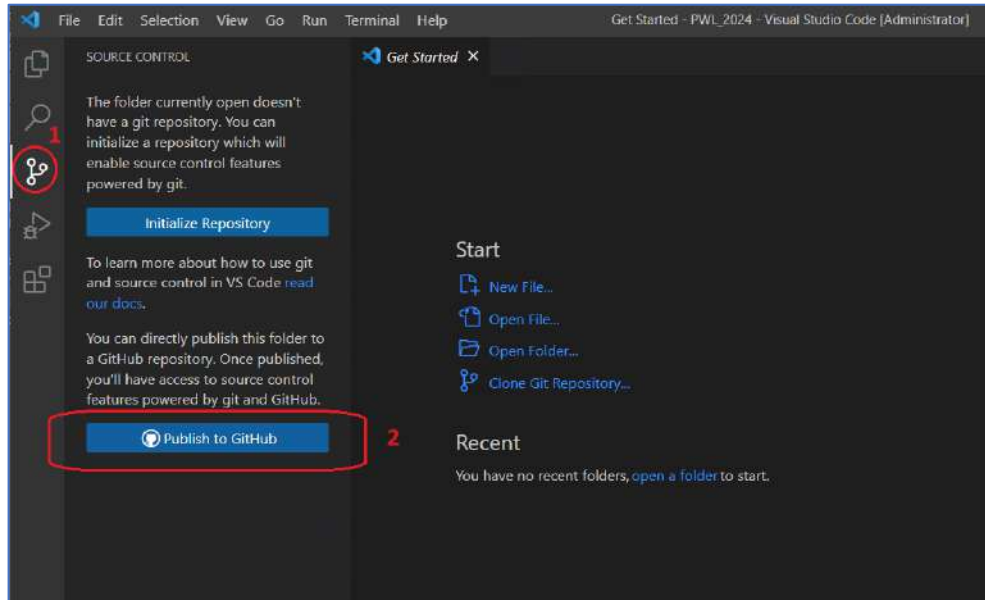


- b. Kemudian pada aplikasi VSCode, klik menu **File** → **Open Folder** → pilih folder **PWL\_2024**

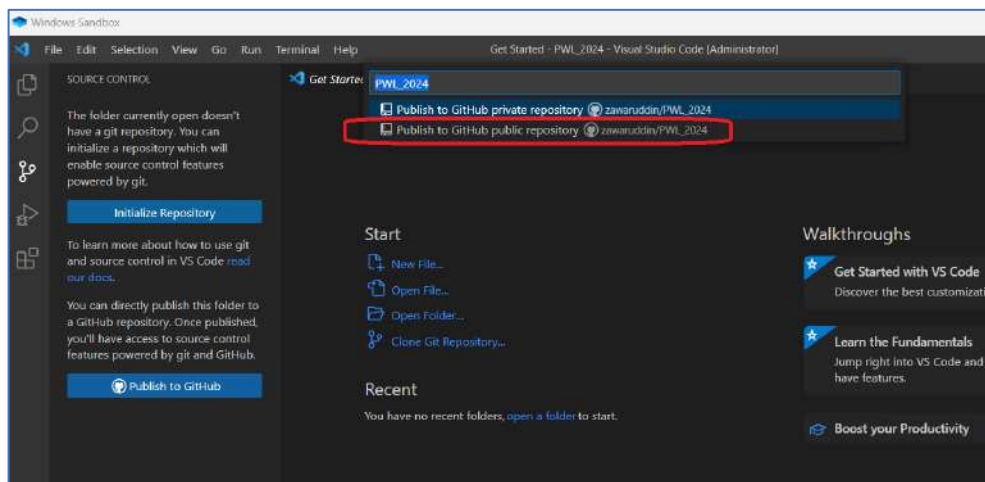




- c. Setelah project PWL\_2024 dibuka di VSCode, selanjutnya perhatikan menu disebelah kiri dari VSCode
- d. Klik menu icon *“Source Control”* dan klik tombol *“Publish to Github”* untuk mem-publish project Laravel kita ke akun github kita masing-masing.

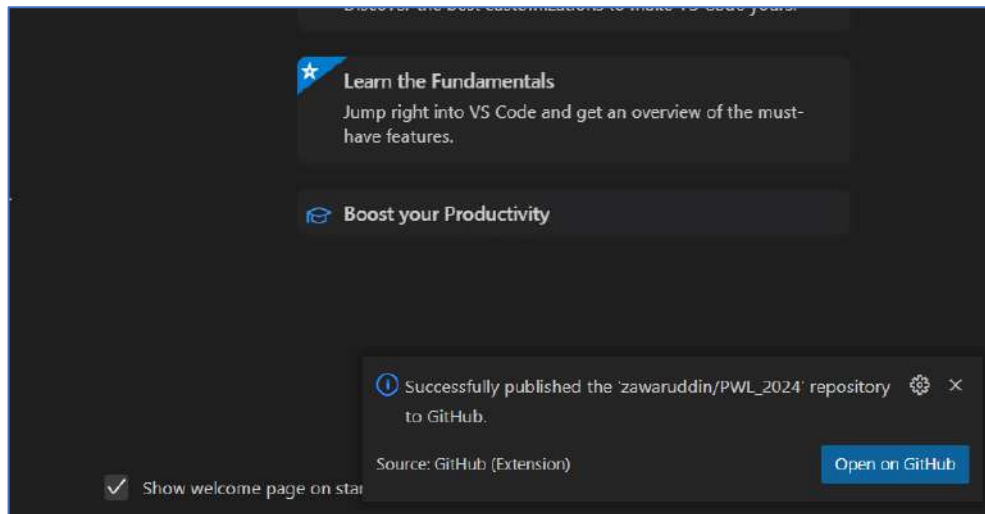


- e. Kemudian pilih opsi yang bawah, yaitu *“Publish to Github public repository”*

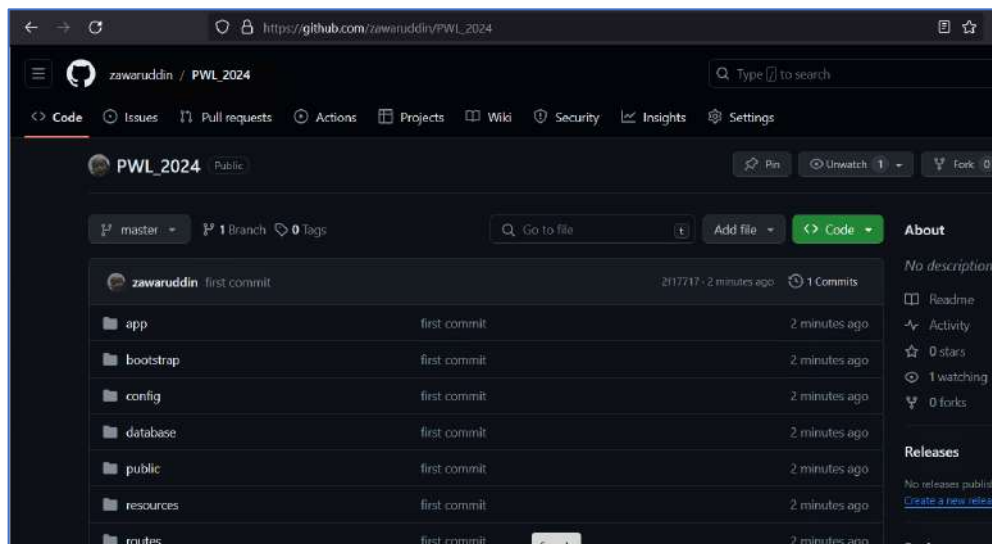


- f. Setelah beberapa saat akan muncul notifikasi untuk login ke akun github. Silahkan isi username dan password dari akun github kalian masing-masing.
- g. Setelah notifikasi login berhasil, maka akan muncul notifikasi bahwa project Laravel kita sudah di publish ke repository github





h. Berikut tampilan project Laravel kita yang bisa kita lihat di repository akun github kita



## 14. Update Kode Program dan Sync ke Github

Ada kalanya kita mengerjakan atau memodifikasi kode program yang sudah kita buat. Dan hal tersebut perlu kita catat perubahan apa saja yang telah kita buat. Nah disini kita bisa memanfaatkan git untuk mencatat perubahan kode program, dan melakukan sinkronisasi dengan repository yang ada di github. Langkah-langkah kita dalam mencatat perubahan kode program bisa seperti berikut

- Buka file `resources/views/welcome.blade.php` pada VSCode
- Modifikasi file tersebut seperti gambar berikut



```
1 <!DOCTYPE html>
2 <html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
3
4     <head>
5         <meta charset="utf-8">
6         <meta name="viewport" content="width=device-width, initial-scale=1">
7         <title>PWL 2024</title>
8     </head>
9     <body>
10         <h2>Pemrograman Web Lanjut 2024</h2>
11         <h3>Semangat Belajar Laravel 10</h3>
12     </body>
13 </html>
```

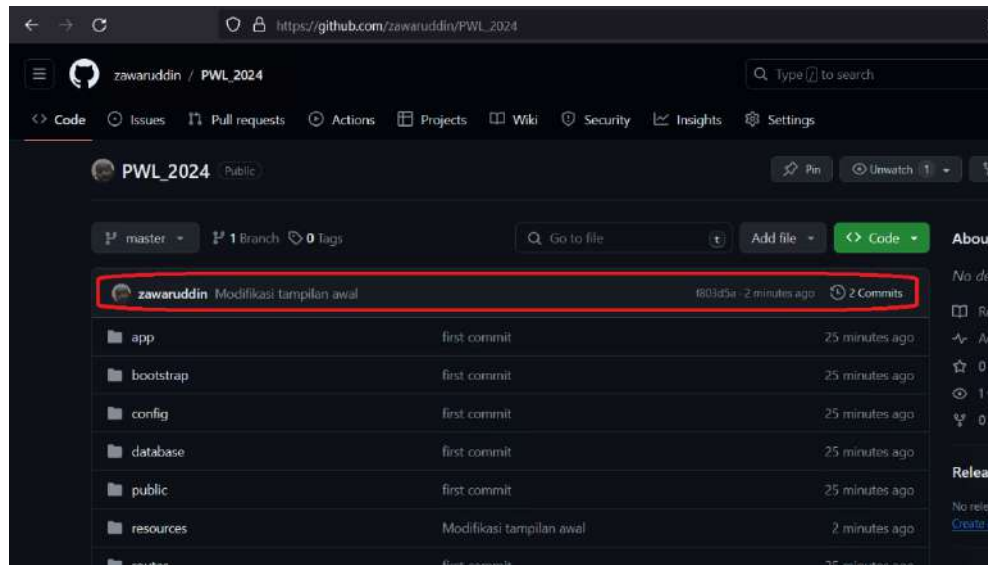
- c. Kemudian simpan, dan coba cek tampilan web kalian pada browser
- d. Sekarang coba perhatikan pada menu icon “Source Control”, terdapat 1 notifikasi (artinya terdapat 1 file yang telah dimodifikasi)

```
1 <!DOCTYPE html>
2 <html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
3
4     <head>
5         <meta charset="utf-8">
6         <meta name="viewport" content="width=device-width, initial-scale=1">
7         <title>PWL 2024</title>
8     </head>
9     <body>
10         <h2>Pemrograman Web Lanjut 2024</h2>
11         <h3>Semangat Belajar Laravel 10</h3>
12     </body>
13 </html>
```

- e. Klik icon menu “*source control*”, tambahkan file yang mau disinkronkan, isi komentar yang ingin di sampaikan (contoh “modifikasi tampilan awal”), kemudian klik tombol *Commit*

```
1 <!DOCTYPE html>
2 <html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
3
4     <head>
5         <meta charset="utf-8">
6         <meta name="viewport" content="width=device-width, initial-scale=1">
7         <title>PWL 2024</title>
8     </head>
9     <body>
10         <h2>Pemrograman Web Lanjut 2024</h2>
11         <h3>Semangat Belajar Laravel 10</h3>
12     </body>
13 </html>
```

- f. Selanjutnya klik tombol “*Sync Changes*” dan klik “*OK*”
- g. Perubahan kode program telah berhasil kita lakukan, dan bisa kita lihat history perubahan tersebut pada repository PWL\_2024 di akun github kita masing-masing



- h. Pelajari mengenai Tahapan 13 ini dengan melakukan Langkah *a-g* secara berulang-ulang agar kalian bisa memahami alurnya.

*\*\*\* Sekian, dan selamat belajar \*\*\**