

# Pengenalan Web Framework

## Pendahuluan

Pada praktikum kali ini anda akan mempelajari tentang pengenalan terhadap web framework dan juga untuk mempersiapkan lingkungan kerja pada pemrograman web lanjut.

## Tujuan Pembelajaran

1. Mahasiswa mampu memahami konsep web framework
2. Mahasiswa mampu memasang web framework
3. Mahasiswa mampu memahami struktur web framework
4. Mahasiswa mampu menggunakan git

## Alat dan Bahan

1. PC atau Laptop
2. Text Editor/IDE (rekomendasi PHPStorm atau VSCode)
3. Git Client
4. Web Browser
5. PHP

## Web Framework

Framework adalah kerangka kerja, sehingga web framework dapat diartikan sebagai kerangka kerja yang bertujuan untuk membantu pembangunan web. Dengan adanya framework pembangunan web menjadi lebih mudah dikarenakan fungsi-fungsi yang sudah disediakan. Selain itu pengembangan web juga menjadi lebih terstruktur, sehingga lebih mudah dalam perawatan web.

Untuk membangun web digunakan bahasa pemrograman yang sebenarnya tidak terbatas pada PHP. Beberapa contoh web framework yang umum digunakan antara lain: Ruby on Rails, Django, CodeIgniter, Laravel dan lainnya. Pada perkuliahan ini akan digunakan Laravel sebagai web framework. Konsep web framework lainnya sebenarnya tidak jauh berbeda. Anda bisa mengeksplorasi secara mandiri melalui dokumentasi official yang tersebar di internet.

Laravel adalah web framework berbahasa pemrograman PHP. Sehingga dibutuhkan instalasi PHP dan composer supaya framework dapat digunakan. Composer merupakan package manager yang digunakan untuk manajemen library dalam suatu project PHP. Definisi library-library yang digunakan dituliskan dalam sebuah file berformat json composer.json. Selain itu pada

Laravel dibutuhkan juga NodeJS sebagai kebutuhan opsional. NodeJS akan dibutuhkan pada saat pembangunan bagian tampilan.

## Struktur Web Framework

Umumnya struktur web framework mengusung konsep MVC (Model View Controller). Web framework memisahkan tampilan, logika dan proses bisnis terlibat untuk memudahkan pembangunan web. Model biasanya berisi pemodelan data dan juga terkait akses terhadap data. View menangani bagaimana tampilan akan ditampilkan. Serta Controller bertanggung jawab dalam menerima input dan mengolah proses yang dibutuhkan. Gambaran struktur framework pada Laravel dapat dijabarkan sebagai berikut.

```
├── README.md
├── app
│   ├── Console
│   ├── Exceptions
│   ├── Http
│   │   ├── Controllers
│   │   └── Middleware
│   ├── Providers
│   └── User.php
├── artisan
├── bootstrap
├── composer.json
├── config
├── database
├── package.json
├── phpunit.xml
├── public
├── resources
│   ├── js
│   ├── lang
│   ├── sass
│   └── views
├── routes
├── server.php
├── storage
├── tests
└── webpack.mix.js
```

Walaupun Laravel mengusung konsep MVC, tetapi untuk susunan directory sedikit berbeda. Pada gambar struktur tersebut yang merupakan Model adalah `User.php`. Class ini diturunkan dari class `Eloquent`. Sedangkan untuk View diletakkan pada lokasi

resources/views. Dan Controller diletakkan pada lokasi app/Http/Controllers. Untuk penjelasan mengenai struktur lain akan dijabarkan pada pertemuan selanjutnya.

## Instalasi Perangkat Lunak Pendukung

Sebelum masuk ke bagian praktikum pastikan sistem anda telah memenuhi kebutuhan berikut PHP >= 7.2 Mysql dan Apache atau menggunakan XAMPP biasanya software tersebut sudah di install pada mata kuliah sebelumnya yaitu Desain dan Pemrograman Web. Software pendukung lain yang harus anda install adalah Git (untuk version control), Composer (untuk package manager php), NodeJs (untuk build front end laravel) dan Laravel sendiri ( Framework )

### Git

Git adalah software untuk melakukan version control terhadap kode program yang anda buat. Git diperlukan untuk memanage kode program anda dan mempermudah anda dalam berkolaborasi dan bekerja sama ketika mengerjakan kode program dalam kelompok. Berikut ini langkah langkah untuk melakukan instalasi git di sistem operasi windows.

1. Periksa apakah git sudah terinstall di sistem operasi anda. Jalankan perintah berikut ini pada terminal / command prompt.

```
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

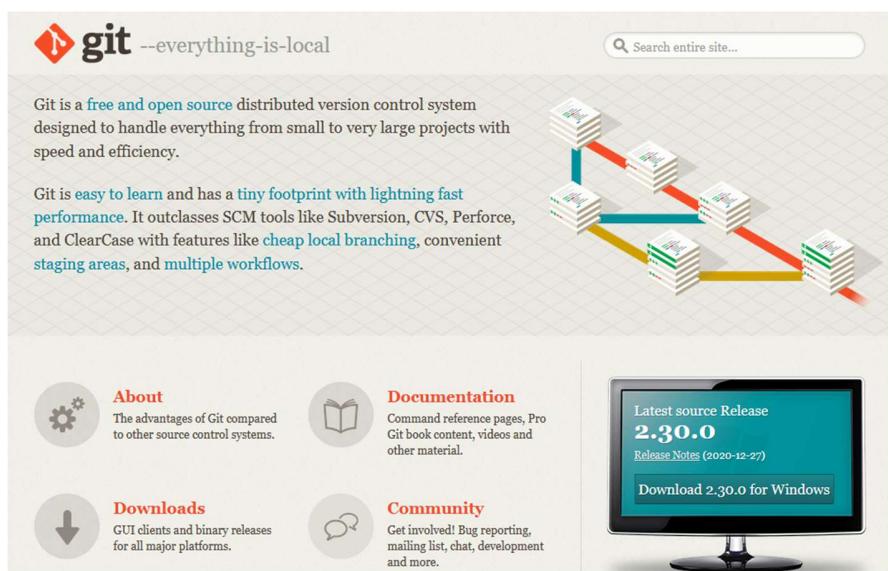
~> git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <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
  reset     Reset current HEAD to the specified state
  rm        Remove files from the working tree and from the index
```

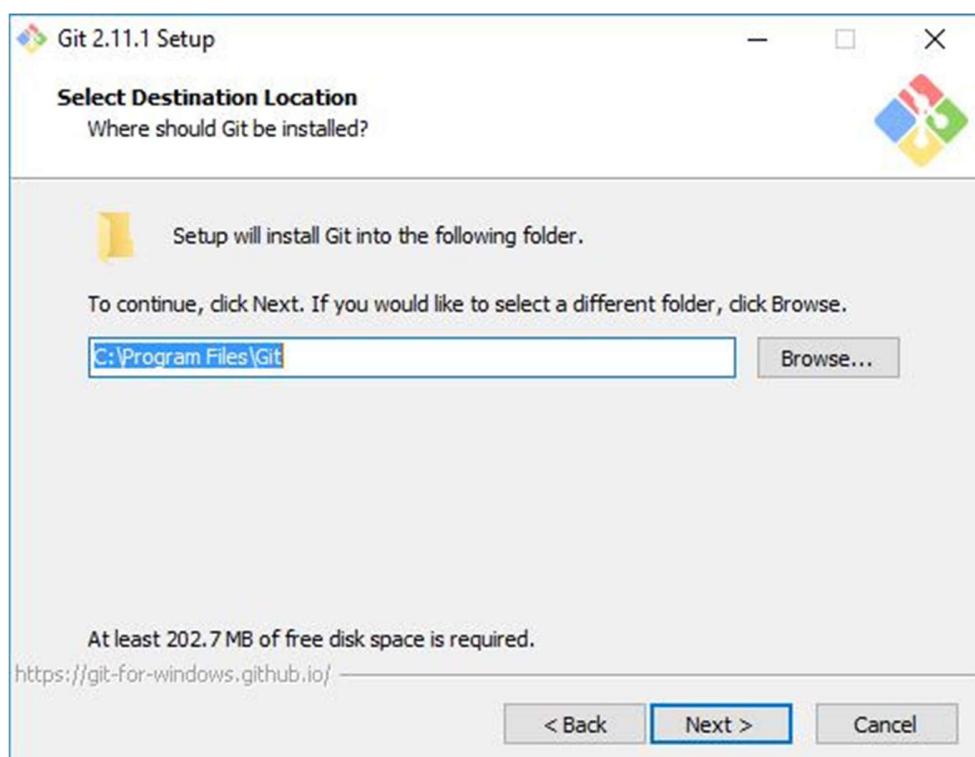
2. Jika keluar output seperti gambar diatas maka anda sudah berhasil melakukan install git, silahkan melanjutkan ke installasi composer. Jika belum lanjutkan ke langkah 3.
3. Download master file git di website berikut ini <https://git-scm.com/>



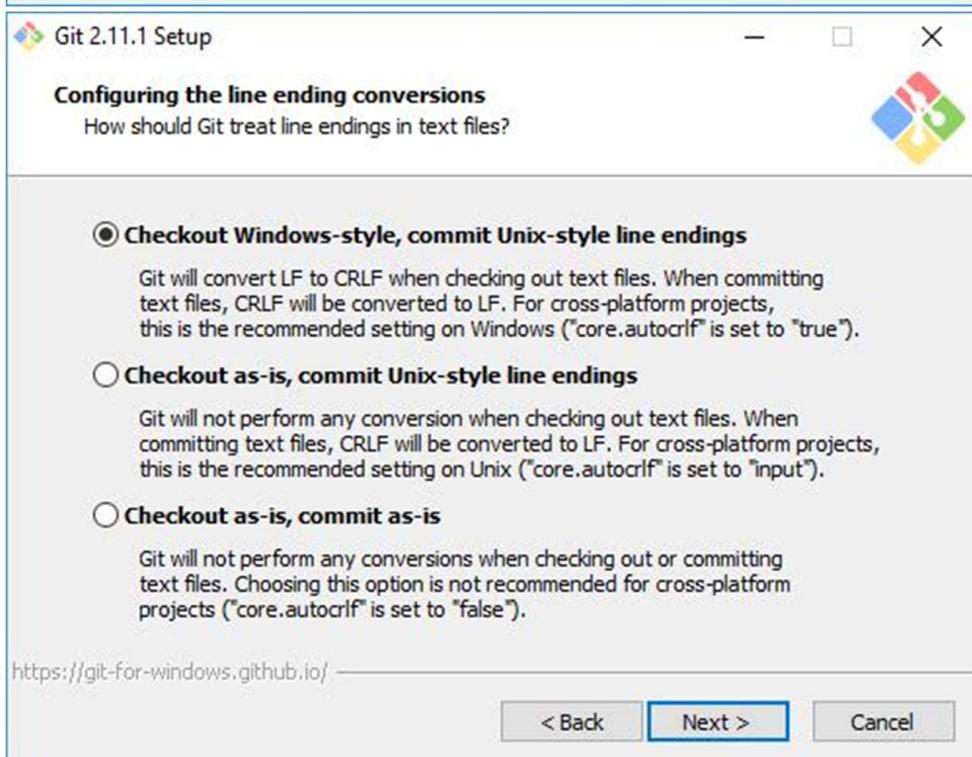
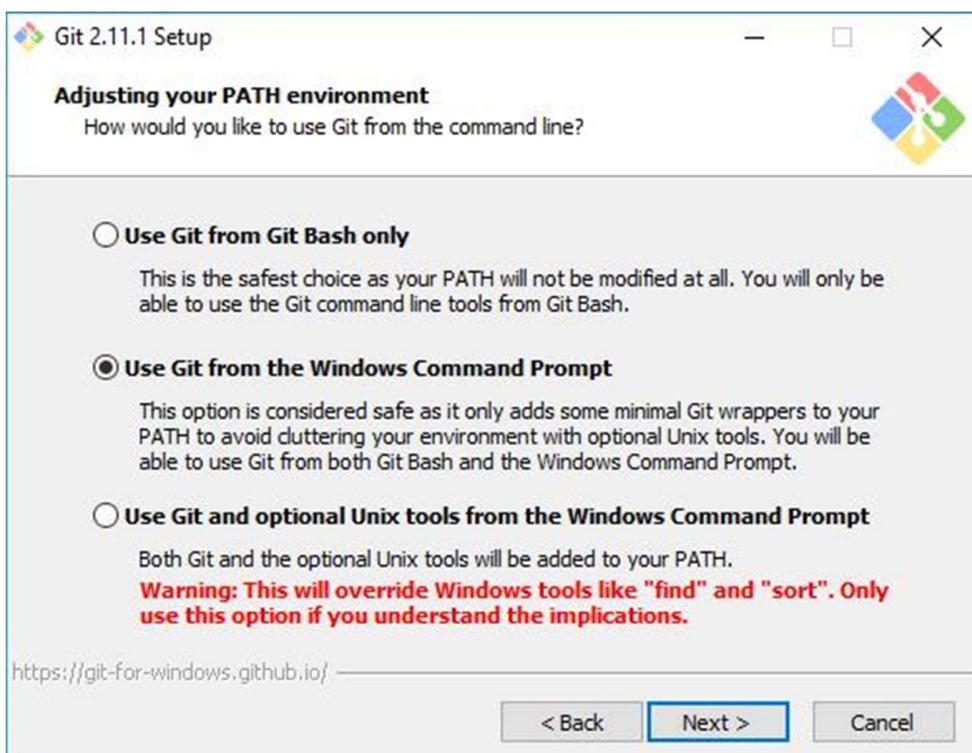
4. Bukalah master installer yang sudah di download dan double klik agar keluar wizard installer berikut ini.

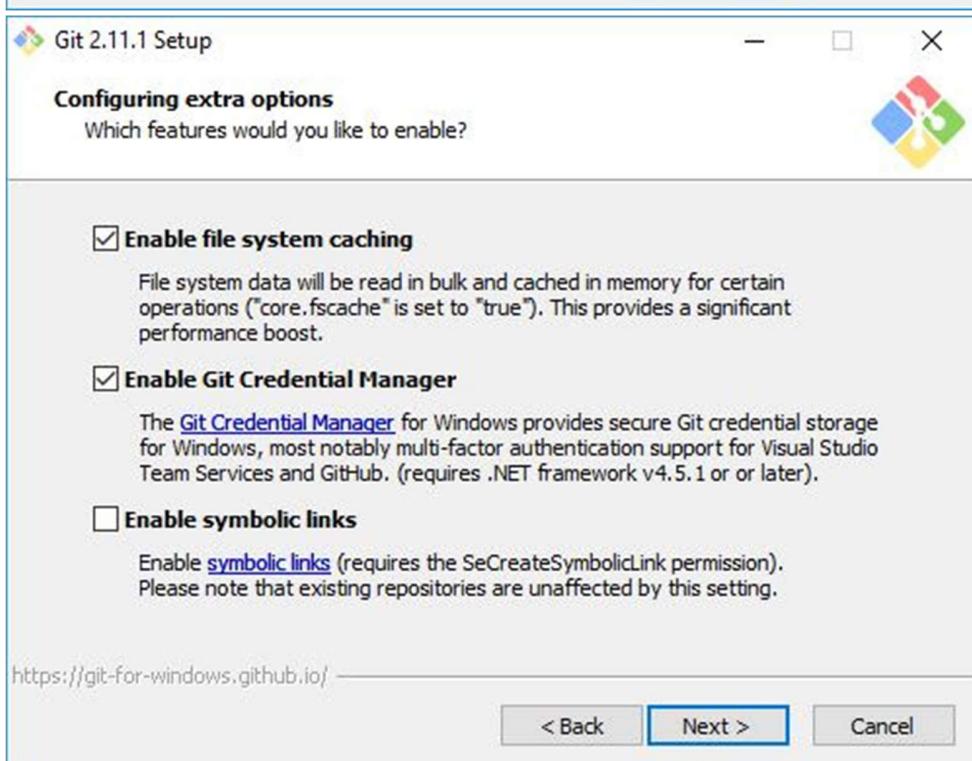
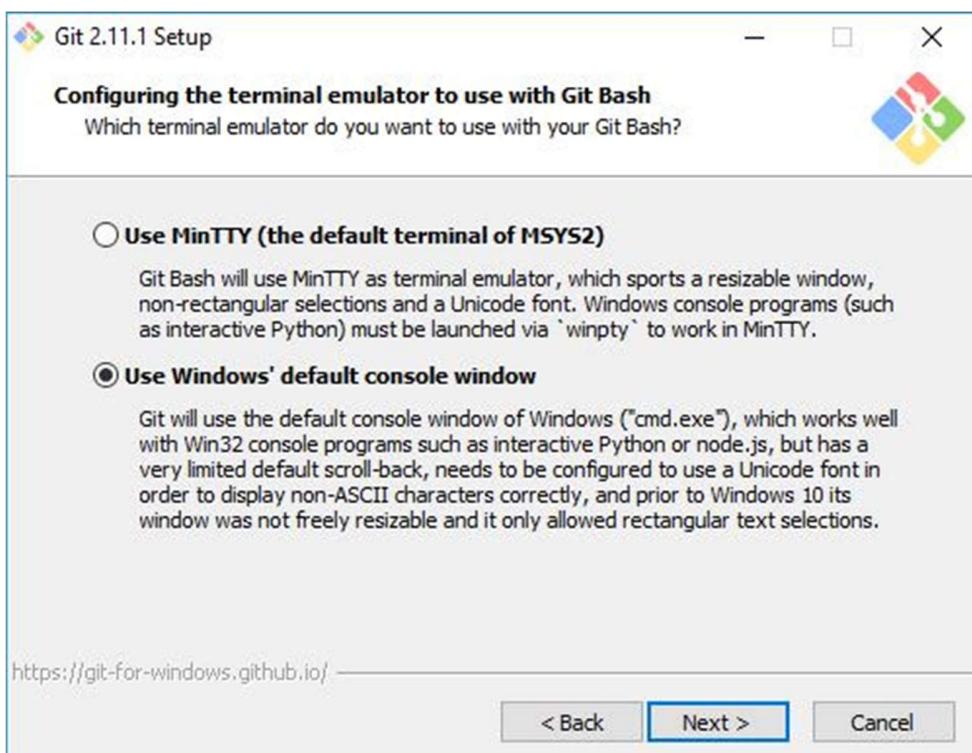


5. Selanjutnya klik next dan pilih lokasi install git



6. 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 instal lagi dengan pilihan default. Berikut ini gambar pilihan pilihan default installer git yang dapat anda ikuti.





 Git 2.11.1 Setup

**Configuring experimental options**  
Which bleeding-edge features would you like to enable?

**Enable experimental, builtin difftool**  
Use the experimental builtin difftool (fast, but only lightly tested).

<https://git-for-windows.github.io/>

< Back **Install** Cancel

 Git 2.11.1 Setup

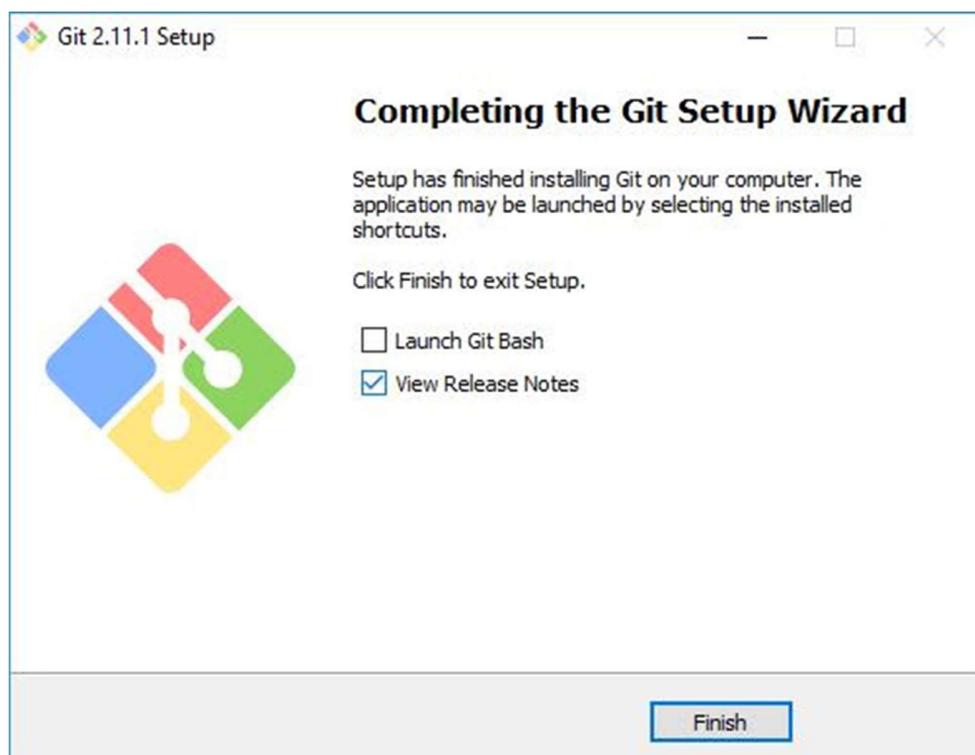
**Installing**  
Please wait while Setup installs Git on your computer.

Extracting files...  
C:\Program Files\Git\mingw64\lib\tcl8.6\tzdata\America\Antigua



<https://git-for-windows.github.io/>

Cancel



7. Selamat anda sudah berhasil melakukan install git pada sistem operasi windows. Langkah selanjutnya adalah melakukan konfigurasi terhadap git agar sesuai dengan akun yang anda miliki.
8. Buka terminal kemudian ketikkan perintah berikut

```
git config --list
```

9. Jika instalasi berhasil dilakukan akan keluar output berikut ini diterminal

```
~  
› git config --list  
core.symlinks=false  
core.autocrlf=true  
core.fscache=true  
color.diff=auto  
color.status=auto  
color.branch=auto  
color.interactive=true  
help.format=html  
rebase.autosquash=true  
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt  
http.sslbackend=openssl
```

```
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
credential.helper=manager
user.name=siubie
user.email=putraprima@gmail.com
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
```

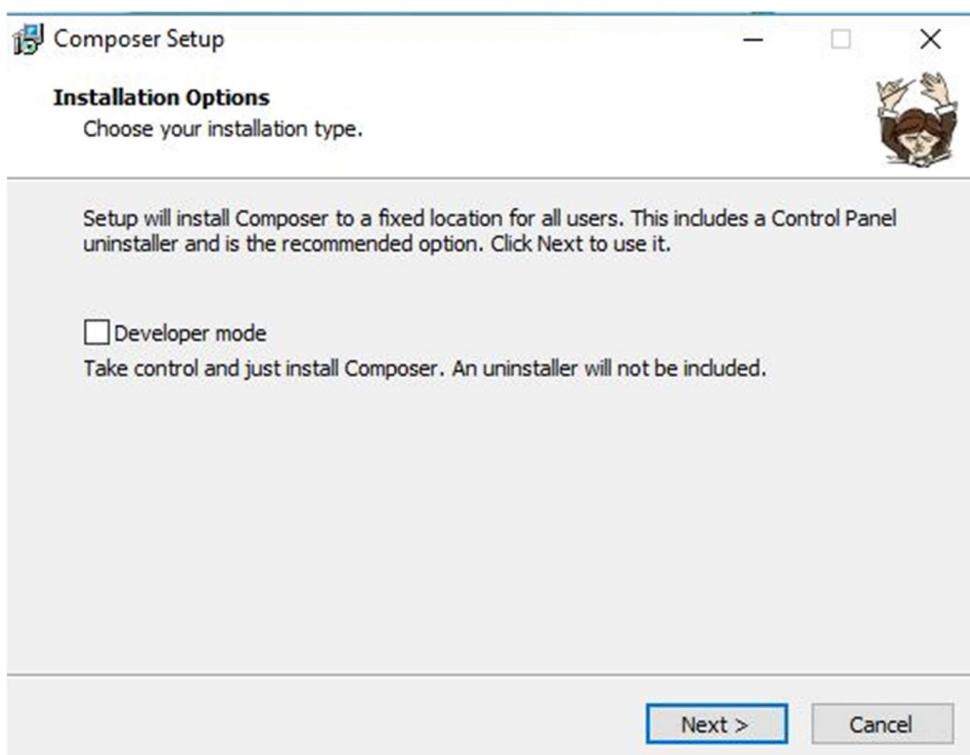
10. Perhatikan pada opsi user.name dan user.email field ini adalah field yang menentukan username dan email git yang akan digunakan pada saat menyimpan perubahan atau publish ke github pastikan opsi tersebut sudah sesuai dengan username dan email anda.
11. Jika belum di set nilai dari user.name dan user.email dapat di set dengan menggunakan perintah berikut ini :

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

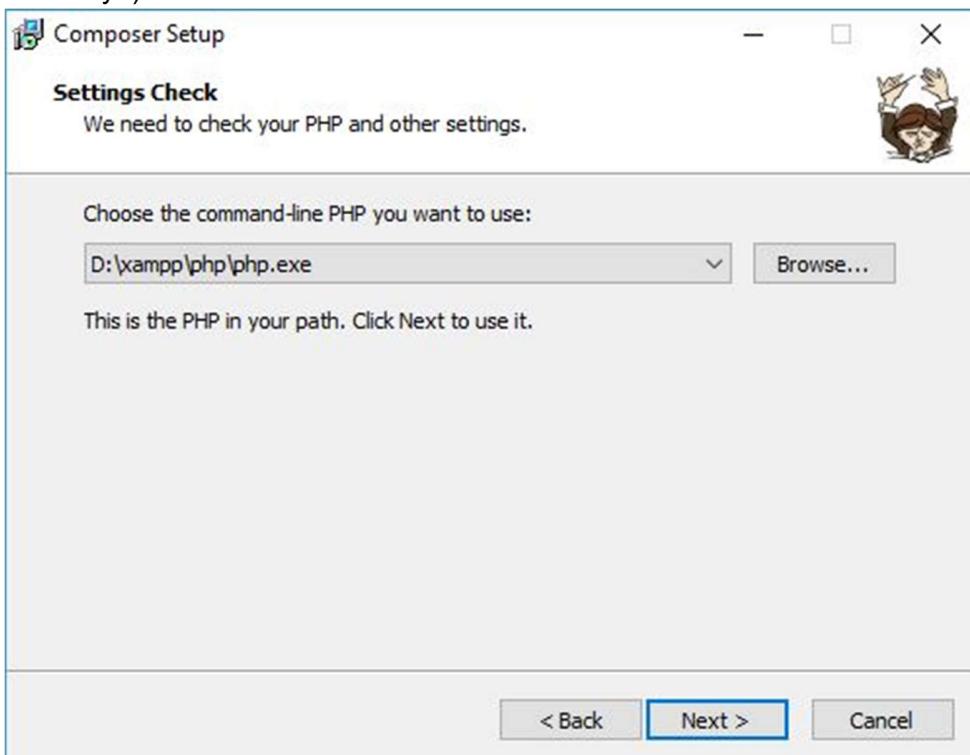
## Composer

Composer adalah sebuah package manager / dependency manager untuk library php. Composer menjadi alat wajib ketika akan membuat project laravel, dengan menggunakan composer anda tidak perlu melakukan copy / paste library secara manual, library akan didownload dan di setup oleh composer. Ketika berjalan composer membutuhkan koneksi internet, oleh karena itu pastikan laptop anda terhubung dengan internet ketika melakukan perintah perintah composer. Berikut ini langkah langkah melakukan instalasi Composer.

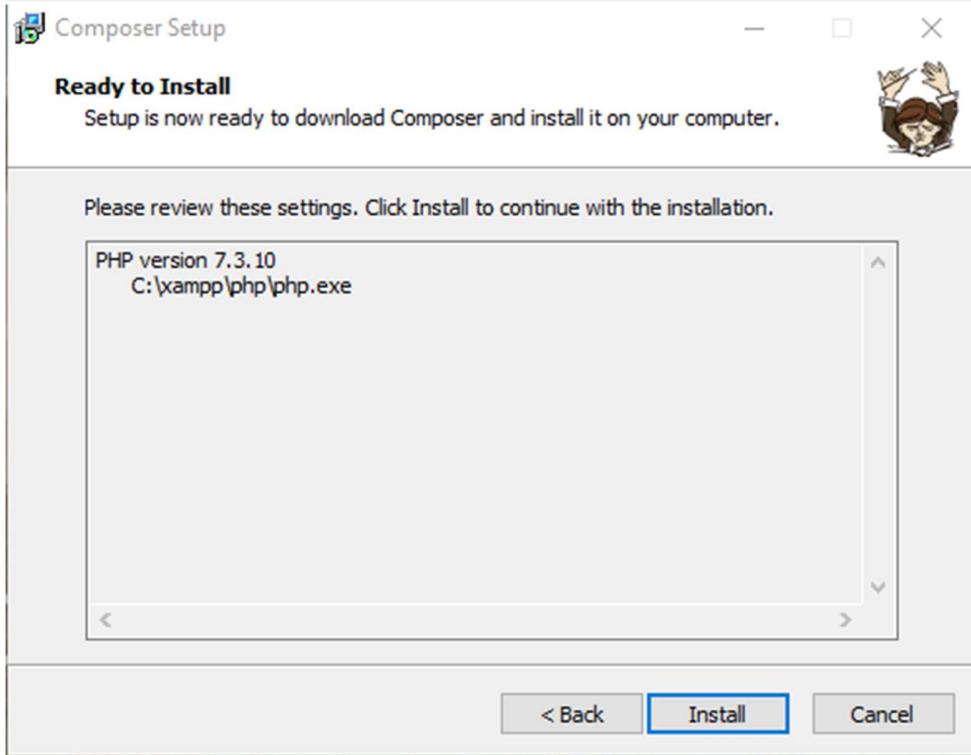
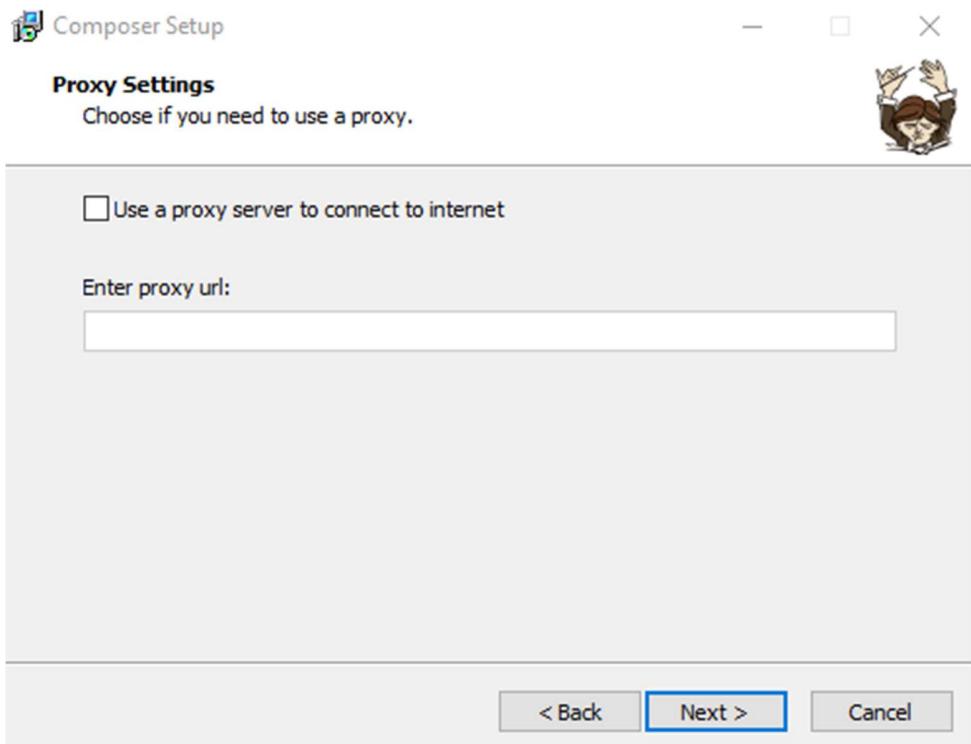
1. Download Composer di tautan berikut ini <https://getcomposer.org/Composer-Setup.exe>
2. Double click installer composer

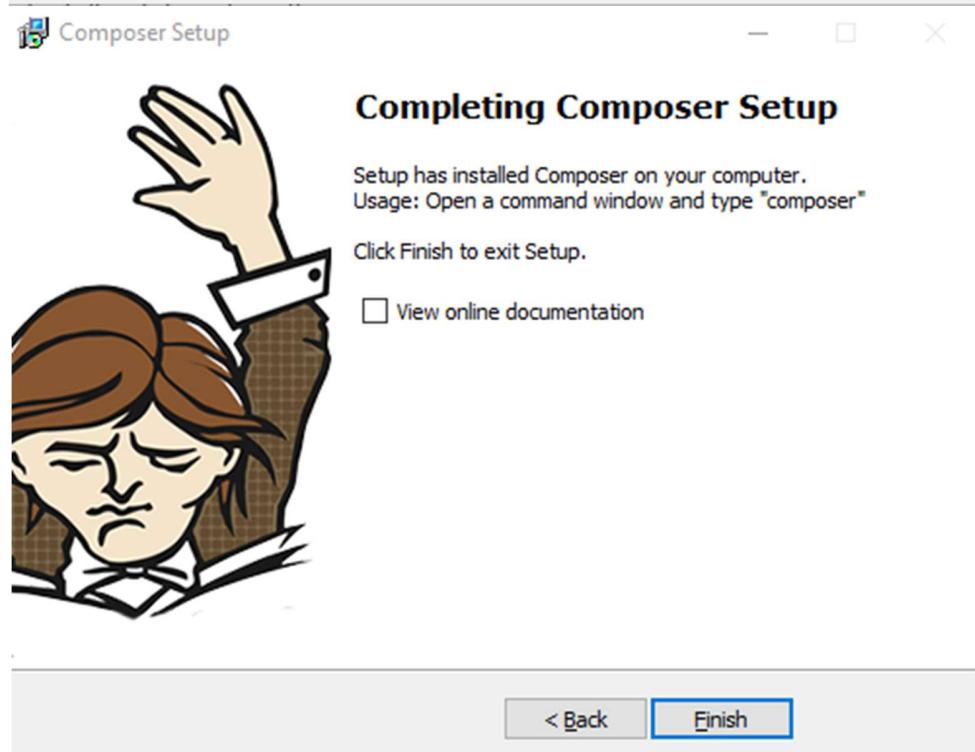
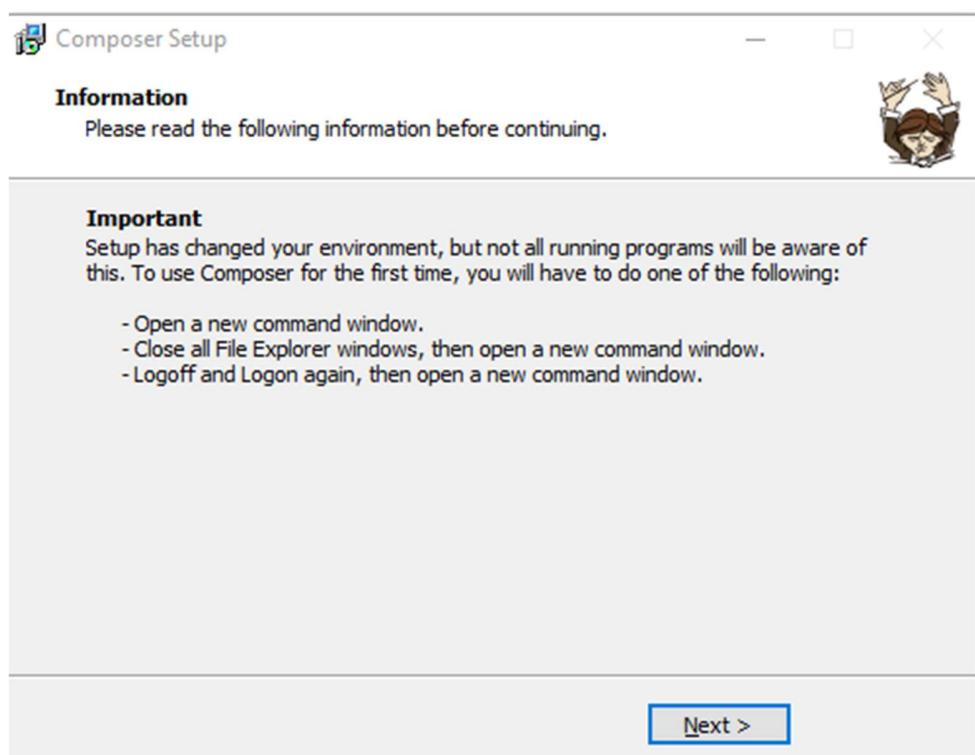


3. Composer membutuhkan lokasi dari file php.exe yang sebelumnya sudah di install (pastikan anda sudah menginstall php, mysql dan apache atau menginstall xampp sebelumnya)



4. Untuk sisa pilihan instalasi biarkan mengikuti nilai default





5. Cek Hasil Instalasi bukalah terminal dan ketikkan composer jika tampil output seperti dibawah ini maka proses instalasi composer telah selesai.

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.18362.418]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\dafit>composer

Composer version 1.9.0 2019-08-02 20:55:32

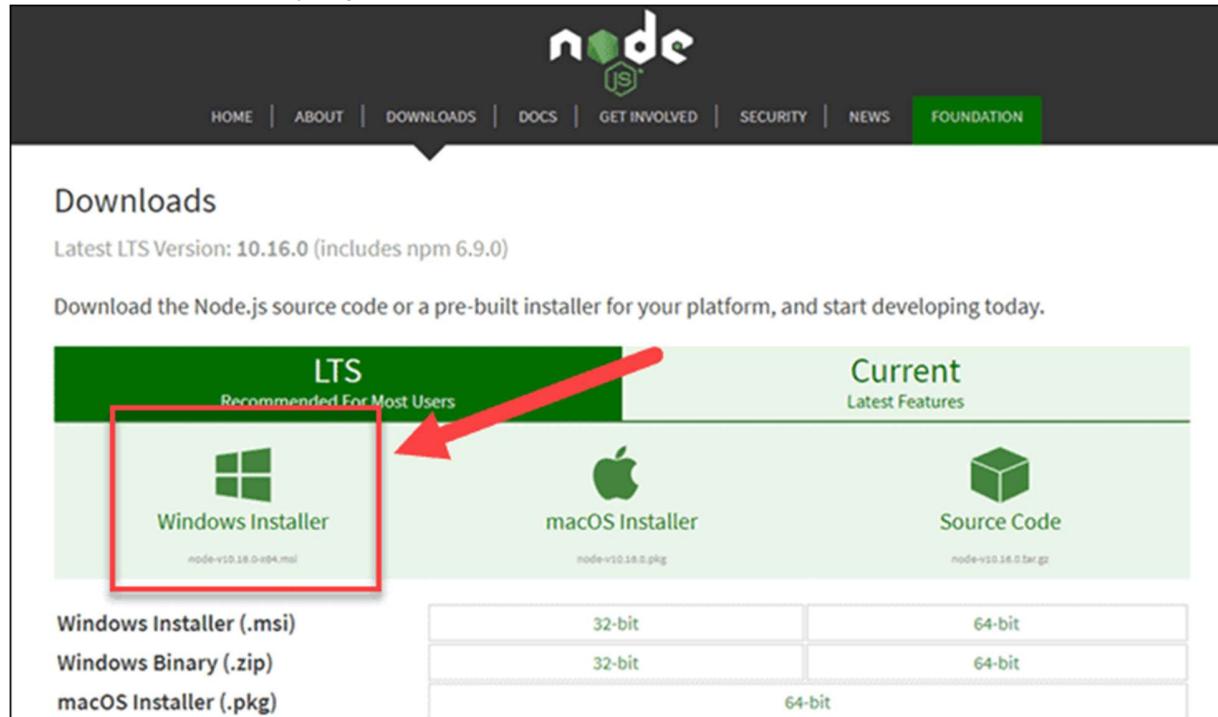
Usage:
  command [options] [arguments]

Options:
  -h, --help          Display this help message
  -q, --quiet         Do not output any message
  -V, --version       Display this application version
  --ansi             Force ANSI output
  --no-ansi          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.
  -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
```

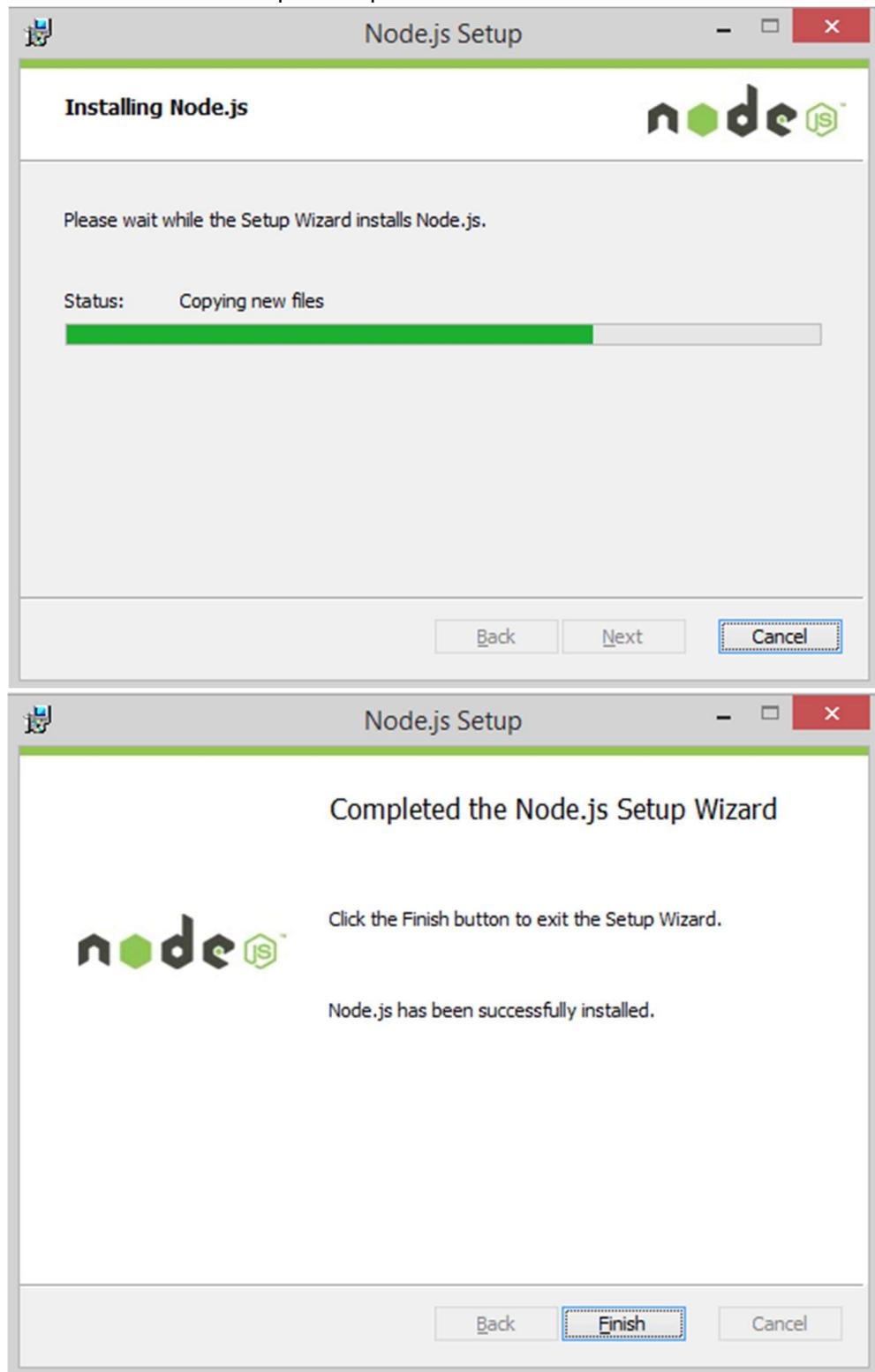
## Nodejs

Nodejs adalah library runtime javascript yang dapat digunakan untuk membuat berbagai macam aplikasi berbasis web, namun pada mata kuliah ini node js digunakan sebagai alat untuk melakukan build front end laravel pada tahap deploy. Berikut ini langkah langkah yang diperlukan untuk melakukan instalasi Nodejs.

1. Download nodejs dari tautan berikut ini <https://nodejs.org/en/download/> kemudian pilih sistem operasi yang sesuai



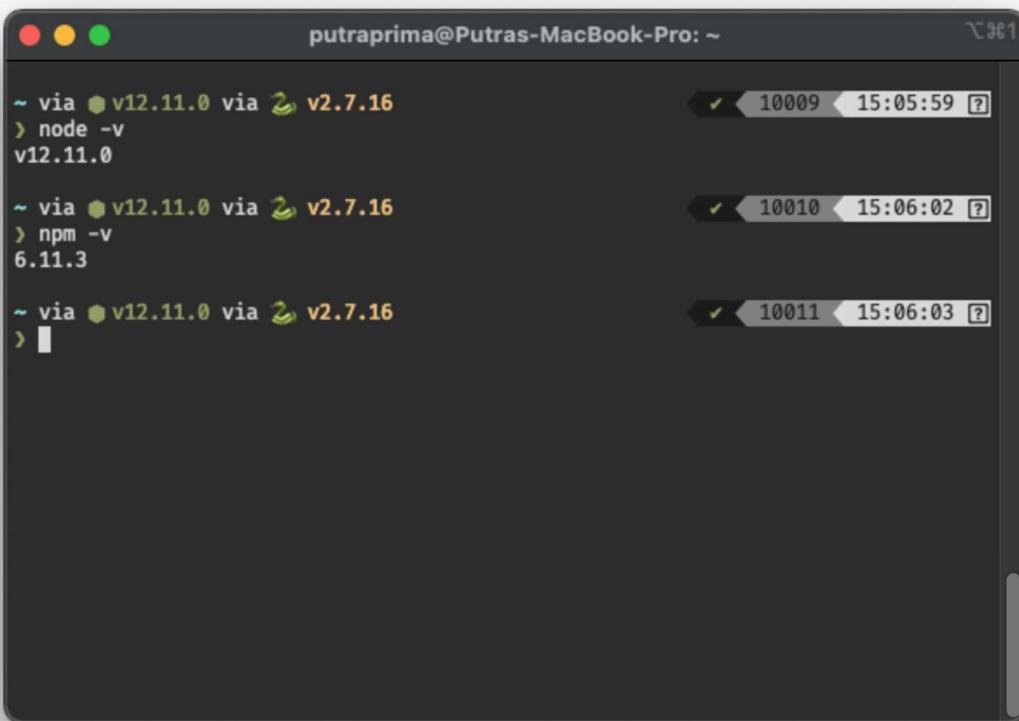
2. Double click installer dan pilihlah pilihan default



3. Periksa apakah nodejs sudah berhasil di install dengan mengetikkan command berikut pada terminal.

```
node -v  
npm -v
```

4. Jika keluar tampilan seperti gambar dibawah maka instalasi nodejs sudah selesai dilakukan. Perhatikan default package manager yang digunakan adalah npm namun anda bisa mengganti dengan package manager yang lain.



A screenshot of a macOS terminal window titled "putraprima@Putras-MacBook-Pro: ~". The window shows three separate command-line sessions:

- The first session shows the output of "node -v" and "npm -v".

```
~ via v12.11.0 via v2.7.16  
› node -v  
v12.11.0
```

```
~ via v12.11.0 via v2.7.16  
› npm -v  
6.11.3
```
- The second session shows the output of "node -v" again.

```
~ via v12.11.0 via v2.7.16  
› node -v  
v12.11.0
```
- The third session shows the output of "npm -v" again.

```
~ via v12.11.0 via v2.7.16  
› npm -v  
6.11.3
```

The terminal has a dark theme and shows timestamps (15:05:59, 15:06:02, 15:06:03) and process IDs (10009, 10010, 10011) for each session.

## Laravel

Laravel dapat diinstall dengan beberapa cara pada praktikum ini akan dijelaskan bagaimana cara melakukan instalasi laravel dengan menggunakan package laraver/installer. Langkah langkah instalasi laravel akan dilakukan pada pros

## Praktikum

### Praktikum 1 - Instalasi Web Framework Laravel

1. Sebelum memasuki langkah-langkah praktikum, periksa instalasi versi PHP (min >= 7.3) dan juga composer pada lingkungan kerja anda. Lanjutkan ke langkah berikutnya, jika telah memenuhi kebutuhan-kebutuhan tersebut.
2. Pada dokumentasi official instalasi Laravel digunakan Laravel installer dengan perintah

```
composer global require laravel/installer  
laravel new <project>
```

Pastikan perintah composer dikenal dalam \$PATH secara luas oleh sistem. Silahkan periksa <https://laravel.com/docs/8.x#installing-laravel> untuk dokumentasi lebih lengkap. Tetapi dikarenakan dalam perkuliahan ini akan digunakan versi 8 sehingga digunakan perintah sebagai berikut:

```
composer create-project --prefer-dist laravel/laravel=8.*  
<project>
```

- Sebelum mengetikkan perintah di atas, arahkan ke direktori dimana project akan disimpan, menggunakan perintah cd  
(misal: cd D:\POLINEMA\Kuliah\Pemrograman Web Lanjut)  
Ubah nilai <project> dengan nama cms (dalam perkuliahan ini, akan dibangun proyek cms sederhana sebagai studi kasus).
3. Tunggu semua proses instalasi sampai selesai. Kecepatan instalasi sangat tergantung dari kecepatan internet dan juga sistem.

## Praktikum 2 - Menjalankan Web Framework Laravel

Untuk menjalankan Laravel pada local development sebenarnya belum diperlukan web server. Dikarenakan Laravel sudah menyediakan wrapper built in web server yang dapat dijalankan dengan perintah `php artisan serve`. Sedangkan pada lingkungan production sebaiknya anda menggunakan web server semacam apache atau nginx.

1. Masuk ke directory di mana anda menyimpan project Laravel pada praktikum sebelumnya. Anda dapat memanfaatkan program terminal cmd atau terminal dalam Text Editor anda.
2. Periksa struktur project Laravel anda, pastikan file `.env` telah ada dalam project anda.
3. Jika anda tidak menemukan file `.env`, anda dapat menyalin isi file tersebut dari `.env.example`. File `.env` berformat key dan value yang digunakan untuk mengatur konfigurasi project Laravel. Dokumentasi lebih lanjut mengenai format ini, silahkan membuka <https://github.com/vlucas/phpdotenv>.
4. Buka isi file `.env` dan pastikan nilai APP\_KEY telah mempunyai nilai. Jika nilainya tidak ada maka diperlukan proses menggenerate nilai ini yang akan dijelaskan pada langkah selanjutnya.

- Pastikan anda masih dalam terminal dan lakukan perintah berikut

```
php artisan key:generate
```

- Untuk menjalankan Laravel ketik perintah berikut dalam terminal

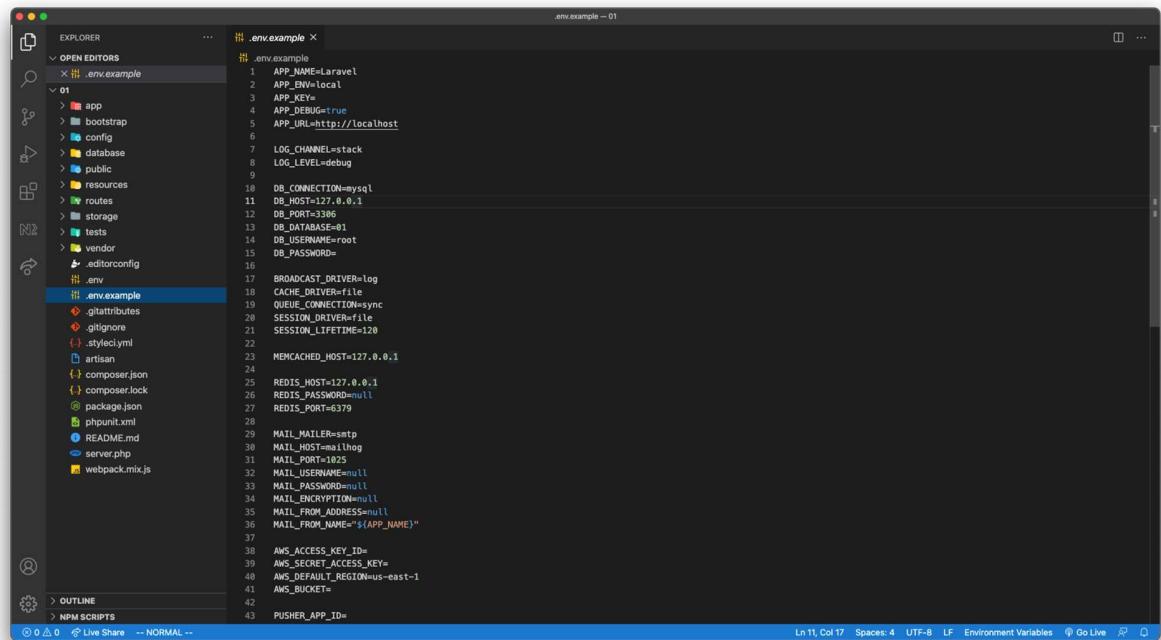
```
php artisan serve
```

- Silahkan eksplorasi nilai-nilai yang ada dalam file .env dengan prefix APP\_ secara mandiri.
- Pembahasan lebih lengkap dapat anda pelajari pada tautan dokumentasi official dari Laravel <https://laravel.com/docs/8.x#installation>. Selamat bereksplorasi.

## Praktikum 3 - Membuat Git Repository dan Publish Ke Github

Ikutilah langkah langkah berikut untuk mempublikasikan pekerjaan anda ke github.

- Mendaftar akun github ke <https://github.com/> , pilihlah username anda dengan bijaksana karena akun github anda akan sangat sering digunakan dan dapat dijadikan portofolio untuk mendaftar pekerjaan.
- Update konfigurasi git di komputer anda sesuai dengan username dan email yang anda buat di github.com (lihat materi tentang instalasi git langkah ke 11)
- Bukalah project laravel yang anda buat sebelumnya pada Praktikum 2 menggunakan Visual Studio Code



```

.env.example - 01
.env.example
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_LEVEL=debug
9
10 DB_CONNECTION=mysql
11 DB_HOST=127.0.0.1
12 DB_PORT=3306
13 DB_DATABASE=01
14 DB_USERNAME=root
15 DB_PASSWORD=
16
17 BROADCAST_DRIVER=log
18 CACHE_DRIVER=file
19 QUEUE_CONNECTION=sync
20 SESSION_DRIVER=file
21 SESSION_LIFETIME=120
22
23 MEMCACHED_HOST=127.0.0.1
24
25 REDIS_HOST=127.0.0.1
26 REDIS_PASSWORD=null
27 REDIS_PORT=6379
28
29 MAIL_MAILER=smtplib
30 MAIL_HOST=mailhog
31 MAIL_PORT=1025
32 MAIL_USERNAME=null
33 MAIL_PASSWORD=null
34 MAIL_ENCRYPTION=null
35 MAIL_FROM_ADDRESS=null
36 MAIL_FROM_NAME="${APP_NAME}"
37
38 AWS_ACCESS_KEY_ID=
39 AWS_SECRET_ACCESS_KEY=
40 AWS_DEFAULT_REGION=us-east-1
41 AWS_BUCKET=
42
43 PUSHER_APP_ID=

```

- Jika anda belum menginisialisasi repository git pada project tersebut akan keluar tampilan seperti ini ketika anda mengklik icon source control di sebelah kiri.

```

.env.example
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_LEVEL=debug
9
10 DB_CONNECTION=mysql
11 DB_HOST=127.0.0.1
12 DB_PORT=3306
13 DB_DATABASE=bd1
14 DB_USERNAME=root
15 DB_PASSWORD=
16
17 BROADCAST_DRIVER=log
18 CACHE_DRIVER=file
19 QUEUE_CONNECTION=sync
20 SESSION_DRIVER=file
21 SESSION_LIFETIME=120
22
23 MEMCACHED_HOST=127.0.0.1
24
25 REDIS_HOST=127.0.0.1
26 REDIS_PASSWORD=null
27 REDIS_PORT=6379
28
29 MAIL_MAILER=smtplib
30 MAIL_HOST=mailhog
31 MAIL_PORT=1025
32 MAIL_USERNAME=null
33 MAIL_PASSWORD=null
34 MAIL_ENCRYPTION=null
35 MAIL_FROM_ADDRESS=null
36 MAIL_FROM_NAME="${APP_NAME}"
37
38 AWS_ACCESS_KEY_ID=
39 AWS_SECRET_ACCESS_KEY=
40 AWS_DEFAULT_REGION=us-east-1
41 AWS_BUCKET=
42
43 PUSHER_APP_ID=

```

Ln 14, Col 16 Spaces: 4 UTF-8 LF Environment Variables Go Live

- Perhatikan ada dua tombol yaitu Initialize Repository dan Publish to Github, Klik lah tombol Initialize Repository. Sidebar akan berubah menjadi seperti berikut ini.

```

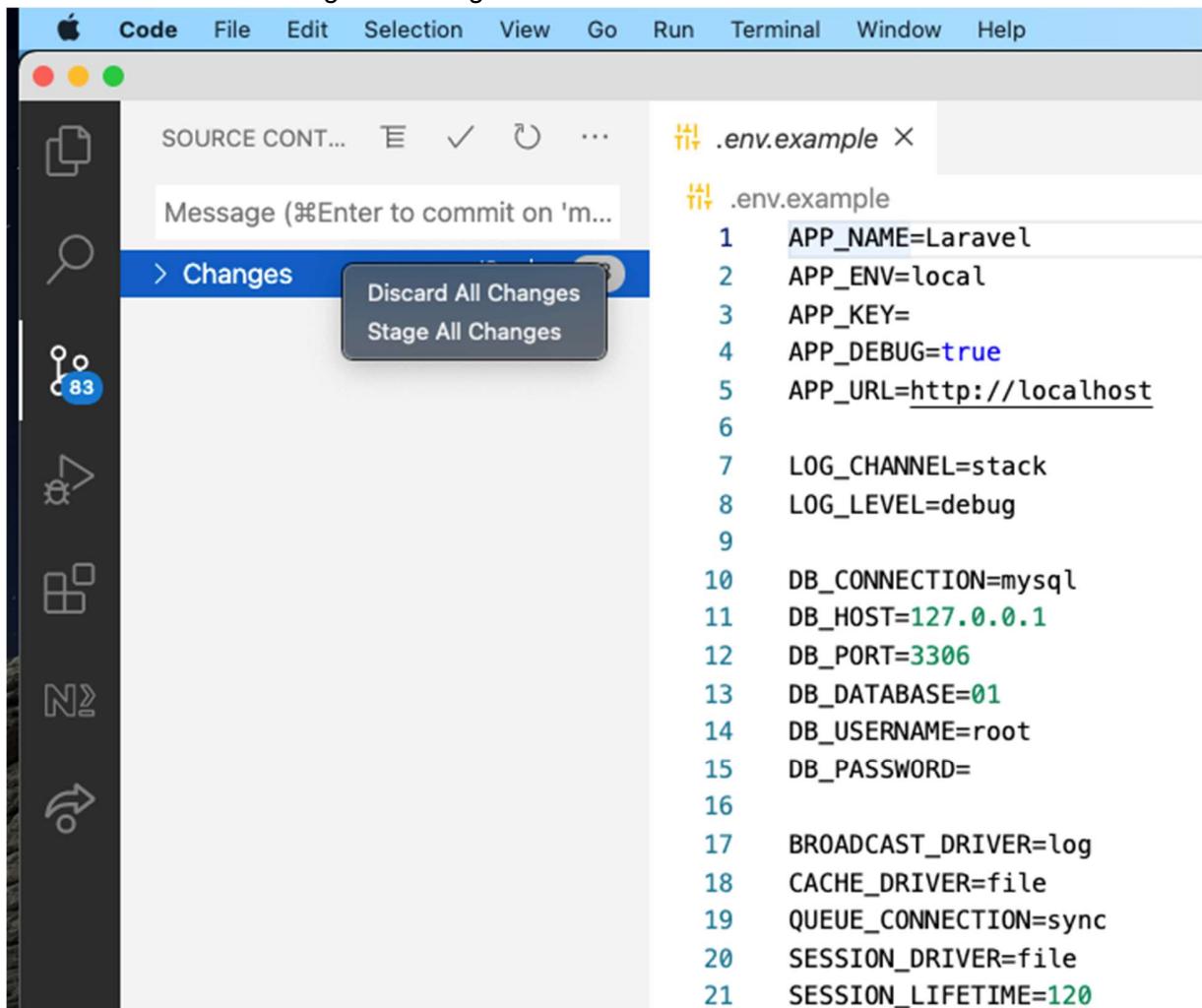
.env.example
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_LEVEL=debug
9
10 DB_CONNECTION=mysql
11 DB_HOST=127.0.0.1
12 DB_PORT=3306
13 DB_DATABASE=bd1
14 DB_USERNAME=root
15 DB_PASSWORD=
16
17 BROADCAST_DRIVER=log
18 CACHE_DRIVER=file
19 QUEUE_CONNECTION=sync
20 SESSION_DRIVER=file
21 SESSION_LIFETIME=120
22
23 MEMCACHED_HOST=127.0.0.1
24
25 REDIS_HOST=127.0.0.1
26 REDIS_PASSWORD=null
27 REDIS_PORT=6379
28
29 MAIL_MAILER=smtplib
30 MAIL_HOST=mailhog
31 MAIL_PORT=1025
32 MAIL_USERNAME=null
33 MAIL_PASSWORD=null
34 MAIL_ENCRYPTION=null
35 MAIL_FROM_ADDRESS=null
36 MAIL_FROM_NAME="${APP_NAME}"
37
38 AWS_ACCESS_KEY_ID=
39 AWS_SECRET_ACCESS_KEY=
40 AWS_DEFAULT_REGION=us-east-1
41 AWS_BUCKET=
42
43 PUSHER_APP_ID=

```

master\* 0 0 0 Live Share — NORMAL — Ln 15, Col 12 Spaces: 4 UTF-8 LF Environment Variables Go Live

- Perhatikan pada langkah ini kita sudah membuat repository git dan git mulai melacak perubahan yang ada pada root folder project kita. Perhatikan dengan seksama di sidebar terdapat daftar file yang berubah dibawah dropdown menu Changes dan status file nya memiliki status U dengan warna hijau.

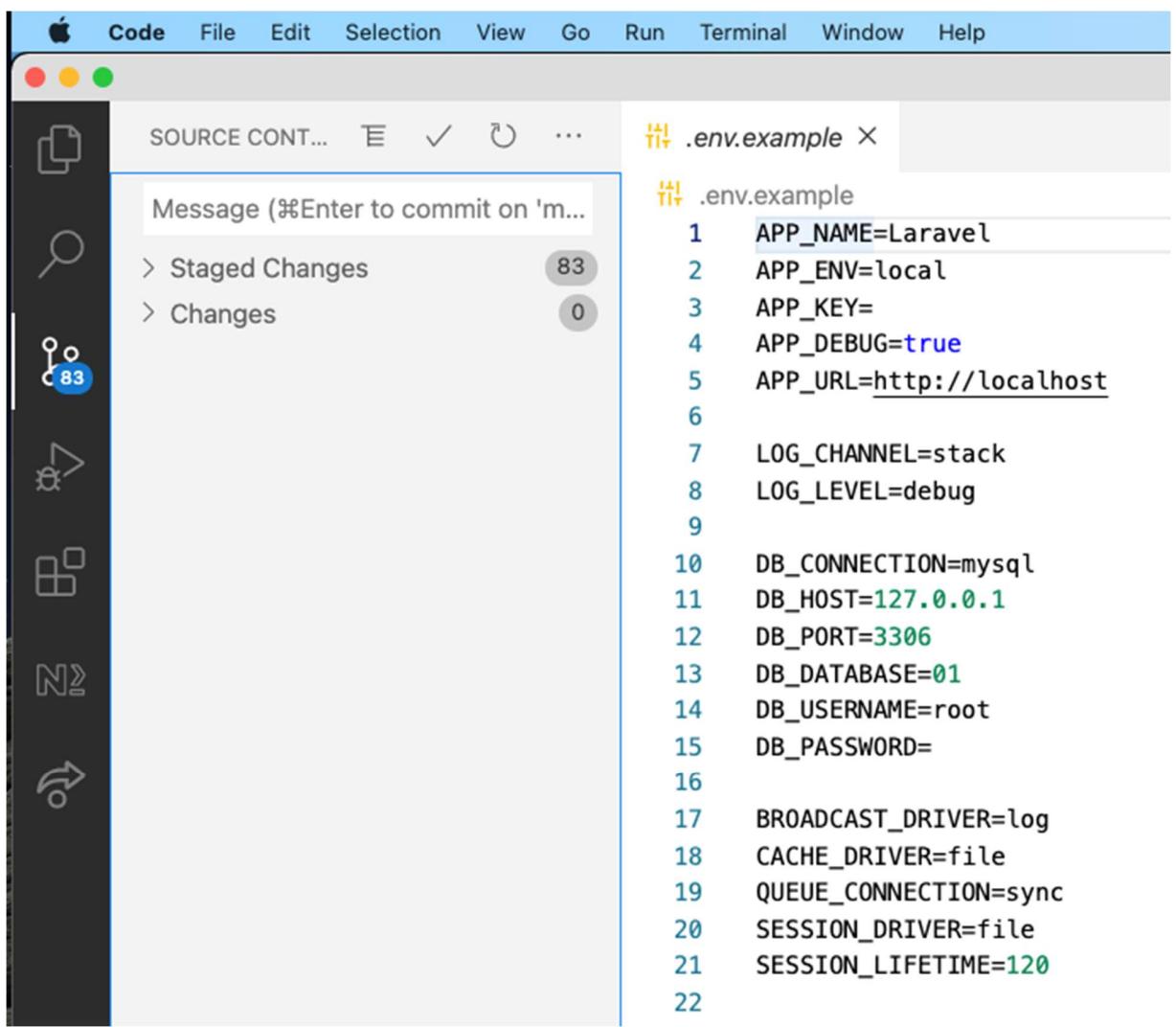
7. Klik dropdown changes ini sehingga daftar file yang berubah diminimize, kemudian klik kanan dan klik menu “Stage All Changes”



The screenshot shows the GitHub Desktop application interface. At the top, there's a menu bar with Apple, Code, File, Edit, Selection, View, Go, Run, Terminal, Window, and Help. Below the menu is a toolbar with icons for file operations like New, Open, Save, and Commit. On the left, there's a sidebar with various icons and a notification count of 83. The main area shows a file named '.env.example' with some configuration code. A context menu is open over the file, with 'Changes' selected. The submenu 'Stage All Changes' is highlighted with a blue border.

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=
APP_DEBUG=true
APP_URL=http://localhost
LOG_CHANNEL=stack
LOG_LEVEL=debug
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=01
DB_USERNAME=root
DB_PASSWORD=
BROADCAST_DRIVER=log
CACHE_DRIVER=file
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120
```

8. Setelah di klik git mencatat semua perubahan yang kita buat pada kode program dan sekarang semua file yang ada di Changes berpindah ke Dropdown Staged Changes.

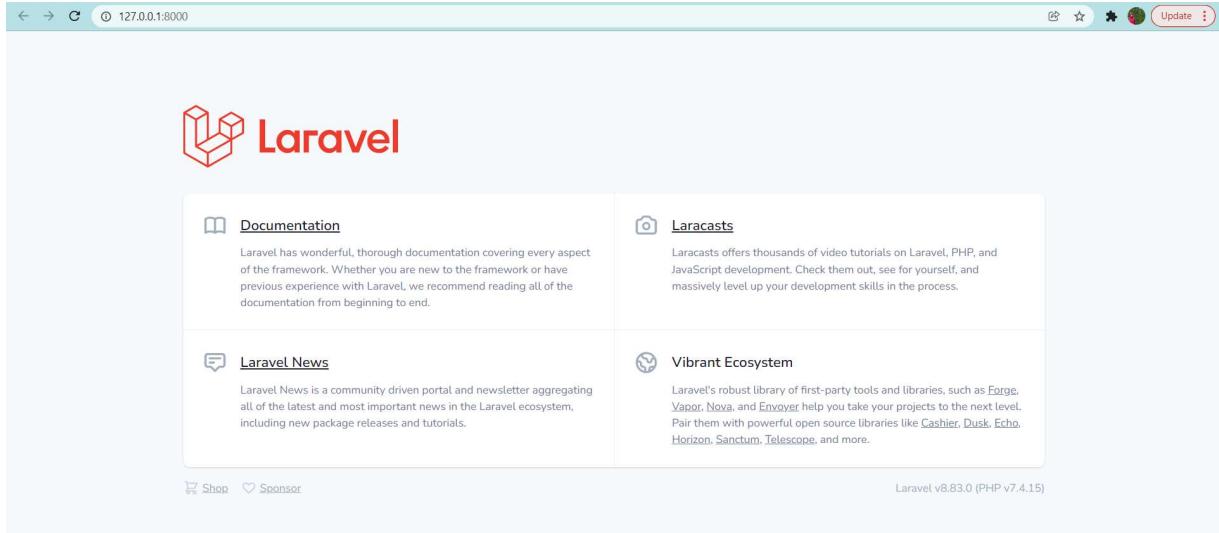


```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=
APP_DEBUG=true
APP_URL=http://localhost
LOG_CHANNEL=stack
LOG_LEVEL=debug
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=01
DB_USERNAME=root
DB_PASSWORD=
BROADCAST_DRIVER=log
CACHE_DRIVER=file
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120
```

9. Isilah pesan tentang perubahan kode yang kita buat, pada kolom isian Message, kemudian klik tombol centang di atas.

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=
APP_DEBUG=true
APP_URL=http://localhost
LOG_CHANNEL=stack
LOG_LEVEL=debug
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=01
DB_USERNAME=root
DB_PASSWORD=
BROADCAST_DRIVER=log
CACHE_DRIVER=file
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120
```

10. Jika di klik menu “Staged Change” akan hilang dan semua perubahan sudah di catat oleh git.
11. Selanjutnya silahkan serve aplikasi laravel anda dengan mengetik php artisan serve kemudian bukalah web nya catat apakah terdapat perubahan.



12. Selanjutnya bukalah file "welcome.blade.php" pada folder resources/views/welcome.blade.php kemudian ubahlah kode html pada bagian body sehingga tinggal seperti ini (Gantilah nama yang tertulis dengan nama, kelas, dan nim anda sendiri)

```

1  <!DOCTYPE html>
2  <html lang="{{ str_replace('_', '-', app()->getLocale()) }}>
3      <head>
4          <meta charset="utf-8">
5          <meta name="viewport" content="width=device-width, initial-scale=1">
6
7          <title>Laravel</title>
8
9          <!-- Fonts -->
10         <link href="https://fonts.googleapis.com/css2?family=Nunito:wght@400;600;700&display=swap" rel="stylesheet">
11
12         <!-- Styles -->
13         <style>
14             /*! normalize.css v8.0.1 | MIT License | github.com/necolas/normalize.css */
15             html{line-height:1.15;-webkit-text-size-adjust:100%}body{margin:0}a{background-color:
16
17             <style>
18                 body {
19                     font-family: 'Nunito', sans-serif;
20                 }
21             </style>
22         </head>
23         <body class="antialiased">
24             <div class="relative flex items-top justify-center min-h-screen bg-gray-100 dark:bg-gray-900 sm:items-center py-4 sm:pt-0">
25                 <div class="ml-4 text-lg leading-7 font-semibold">
26                     <h1 class="text-grey-900 dark:text-white">
27                         Milyun Ni'ma Shoumi <br><br>
28                         Kelas: 2M, NIM: 129076325345
29                     </h1></div>
30                 </div>
31             </div>
32         </body>
33     </html>
```

13. Setelah mengganti cek tampilan web dengan membuka kembali url <http://localhost:8000/>



14. Periksalah kembali tab version control di sidebar dari visual studio code anda kemudian anda akan melihat perubahan yang terjadi pada file tersebut dan berapa jumlah file yang berubah.

```

diff --git a/welcome.blade.php b/welcome.blade.php
--- a/welcome.blade.php
+++ b/welcome.blade.php
@@ -25,10 +25,10 @@ @if (Route::has('login'))
    <div class="hidden fixed top-0 right-0 px-4 sm:block">
        @auth
            <a href="{{ url('/home') }}" class="text-sm text-gray-700 font-semibold">
-                @if (Route::has('register'))
+                <a href="{{ route('register') }}" class="text-sm text-gray-700 font-semibold">
                    <span>Register</span>
                @endif
            @endauth
        </div>
    @endif
@@ -39,7 +39,7 @@ <div class="max-w-6xl mx-auto sm:px-6 lg:px-8">
    <div class="flex justify-center pt-8 sm:justify-start sm:pt-0" style="background-color: #f0f0f0; padding: 10px; border-radius: 10px; width: fit-content; margin: auto; height: fit-content; border: 1px solid #ccc; position: relative; z-index: 1; position: relative; z-index: 1;">
        <div class="relative flex items-top justify-center min-h-screen bg-gray-100 dark:bg-gray-900 sm:items-center sm:pt-0" style="position: absolute; top: 0; left: 0; width: 100%; height: 100%; background-color: white; z-index: 0;">
            <div class="ml-4 text-lg leading-7 font-semibold">H1 Class</div>

```

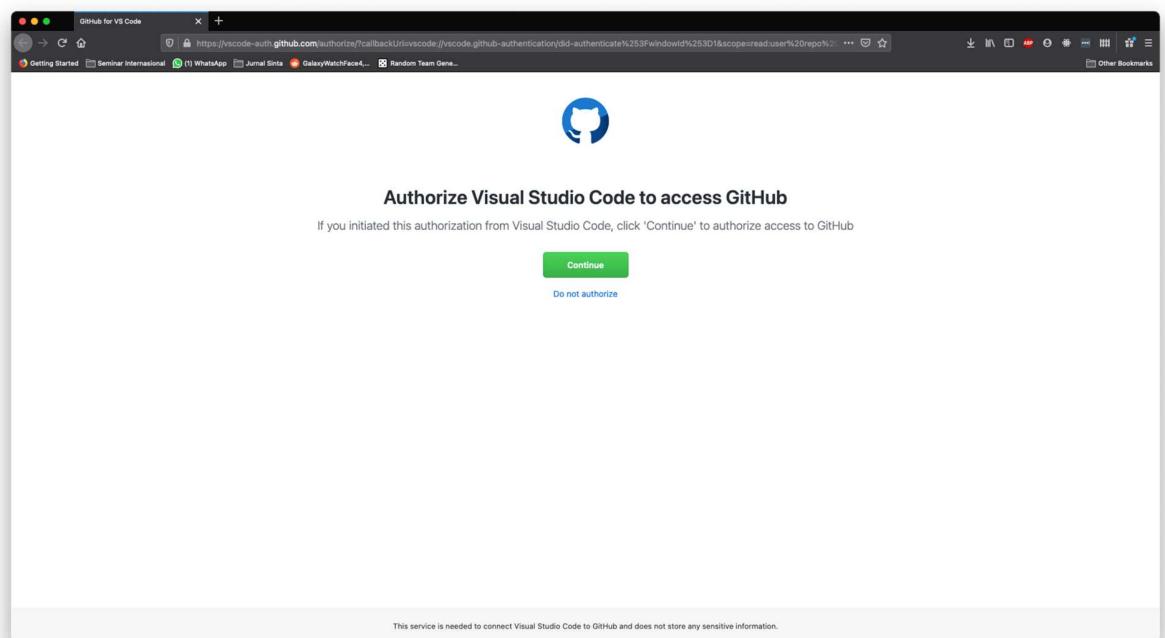
15. Di version control ini anda dapat melihat kode program sebelumnya di sebelah kiri dan kode program sekarang, pada bagian kode sebelumnya yang berwarna merah berarti dihapus sedangkan disebelah kanan baris yang berwarna hijau berarti mengalami perubahan.
16. Lanjutkan proses ini dengan melakukan staging dan berilah pesan yang sesuai dengan perubahan yang anda buat. (ulangi langkah 7 sampai dengan 10)
17. Selamat anda sudah berhasil menggunakan git pada repository lokal.
18. Selanjutnya untuk melakukan publikasi repository lokal ke github lakukan perintah berikut ini tekan Ctrl + Shift + P kemudian ketik publish to github

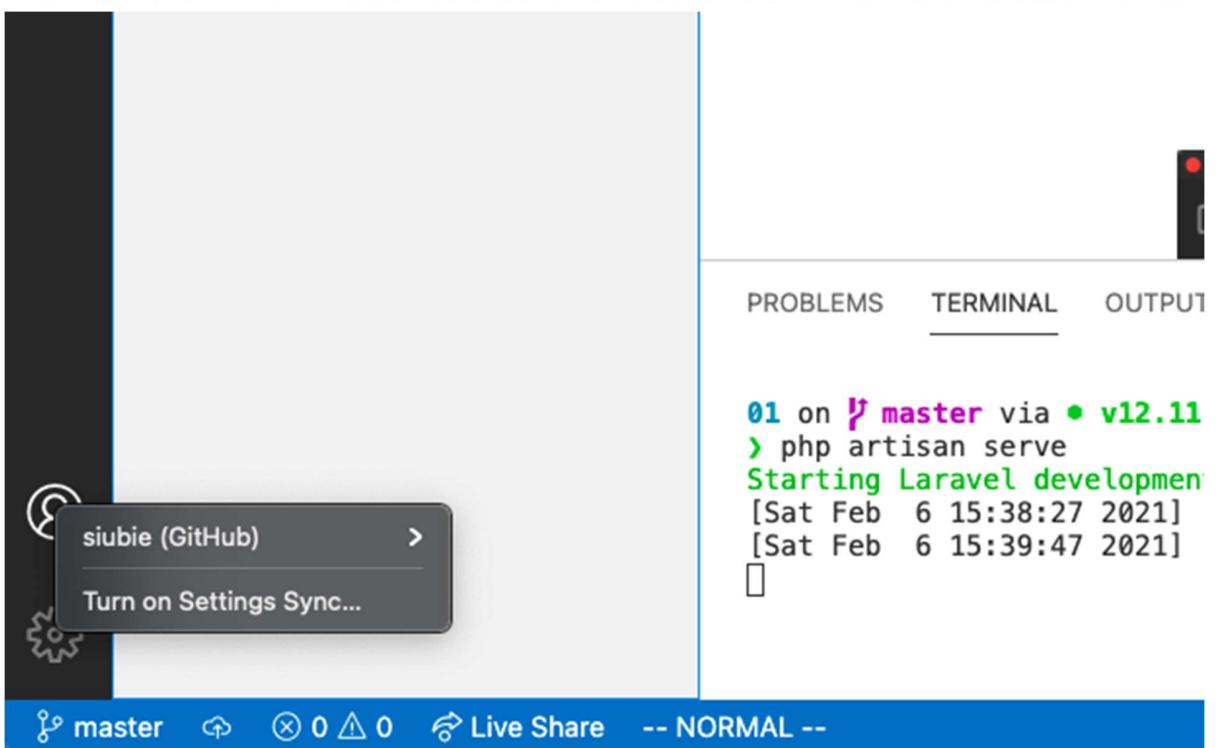
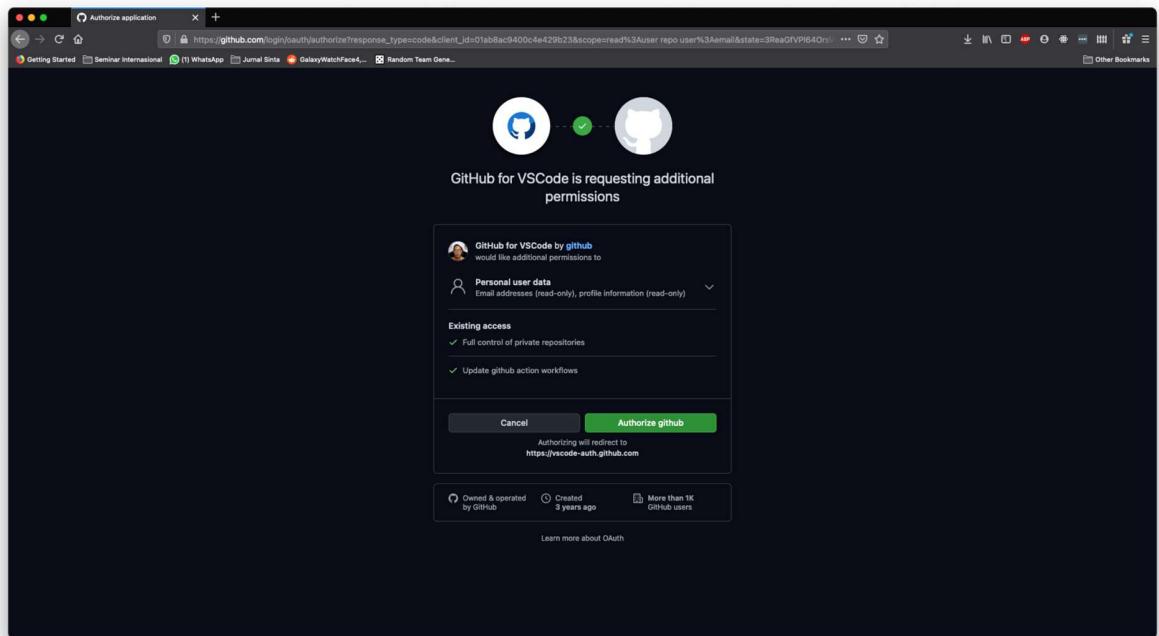
A screenshot of the Visual Studio Code interface. The code editor shows a portion of a CSS file with the following content:

```
ip >
  welcor Publish to GitHub
  html> Git: Push
  ={{ str_ Git: Add Remote...
  eta char Preferences: Color Theme
  eta name Unfold All
  Fold All
  itle>Lara XML Tools: Format as XML
  -- Fonts Format Document
  ink href= Add gitignore
  Fold Recursively
  -- Styles Expand Selection
  tyle> Collapse Folders in Explorer
  /*! no WakaTime: Api Key
  style> Copy With Syntax Highlighting
  tyle>
  body {
    font-family: 'Nunito';
  }
```

A context menu is open over the CSS selector ".font-family". The menu includes options like "Publish to GitHub", "Git: Push", "Format Document", and "Copy With Syntax Highlighting". The "recently used" tab of the menu bar is selected.

19. Visual studio code akan membuka browser dan meminta otorisasi untuk akun github





20. Setelah otorisasi berhasil anda dapat melanjutkan kembali command Ctrl + Shift + P dan pilih publish to github kemudian berilah nama laravel-hello-world

welcome.blade.php — 01

laravel-hello-world

elcor  Publish to GitHub private repository siubie/laravel-hello-world  
str  Publish to GitHub public repository siubie/laravel-hello-world

```
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
```

## Bahan Bacaan

1. Dokumentasi Laravel <https://laravel.com/docs/8.x/installation>
2. Instalasi Git <https://www.petanikode.com/git-install/>
3. Instalasi Composer <https://www.niagahoster.co.id/blog/cara-install-composer/>
4. Instalasi NodeJs <https://www.dumetschool.com/blog/cara-install-node-js-dan-npm-pada-windows>
5. Install Laravel <https://www.niagahoster.co.id/blog/cara-install-laravel-di-windows/> atau <https://blogs.masterweb.com/cara-install-laravel-di-windows/>
6. Git di visual studio code <https://medium.com/aisy-rozsidhy/sharing-menggunakan-dan-memanfaatkan-fitur-git-di-visual-studio-code-76e833b28b7d>