



Balai Besar Pengembangan Penjaminan
Mutu Pendidikan Vokasi
Bidang Otomotif dan Elektronika

MODUL WEB TECHNOLOGIES



Persiapan LKS Nasional

PEMBIMBING :

Dr. Eko Subiyantoro, S.Pd.,S.S.T.,M.T
David Naista, S.Kom., M.T.I

TIM PENYUSUN :

1. Yudi Febriandy, S.Kom
2. Agus, S.Pd,
3. Andy Rafi, S.Kom., M.Kom,
4. Agustinus Karaeng Kanaka, S.Kom.
5. Andrias Koto, S.Kom, MM.
6. Mohammad Zakaria Iba, S.Kom
7. Kadek Unggah Adi Nope, S.Pd.
8. Aspert Mohede, S.Pd

9. Dere Jalu Putro Nuswantoro, S.ST.
10. Christonus Joens Luturmas, S.Pd.Kom
- III. Feni Iriany, S.Kom,
12. Sanry Merciana Nisikoen, S.Kom
13. Yanuar Setyoningsih, S.Pd
14. Mukhamad Syafiullah,S.Pd., Gr.
15. Ilman Pradhana, S.Kom., M.Kom



MODUL I

WEB DEVELOPER DENGAN LARAVEL

A. APA ITU PHP FRAMEWORK

PHP framework adalah platform yang digunakan sebagai kerangka kerja untuk membangun aplikasi web berbasis PHP. PHP framework berisi kumpulan library dari kode yang sering digunakan, yang bisa membantu para developer untuk dapat membuat aplikasi web lebih cepat.

PHP framework memberikan kerangka dasar dalam membuat aplikasi web berbasis PHP. Framework bekerja untuk menstandarisasi dan mempercepat proses pengembangan, serta mengurangi kesalahan, dan keraguan dalam membangun aplikasi.

B. APA ITU LARAVEL

Laravel adalah sebuah framework PHP hasil pengembangan Taylor Otwell yang pertama kali resmi dirilis pada tahun 2011. Framework ini didesain dengan prinsip-prinsip yang membuat pengembangan aplikasi berbasis web lebih efisien, dinamis, dan powerful.

Framework ini merupakan salah satu dari beberapa framework PHP open-source yang menerapkan konsep arsitektur Model-View-Controller (MVC) di bagian back-end atau server-side dengan sangat baik. Dengan MVC, aplikasi terbagi menjadi tiga komponen utama, yakni komponen Model, View, dan Controller.

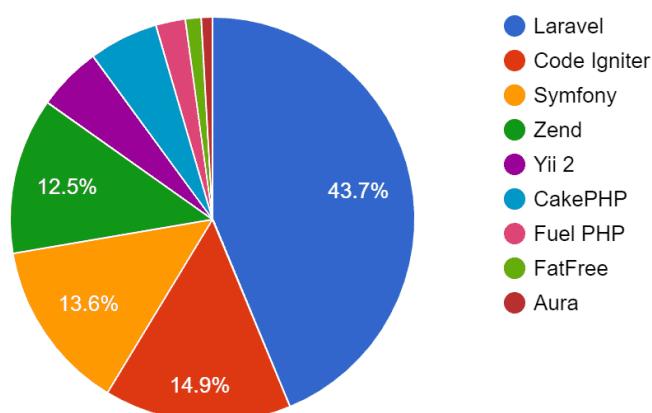
Laravel adalah framework atau platform yang berfungsi untuk mengembangkan aplikasi web dengan bahasa pemrograman PHP. Aplikasi web sendiri merupakan aplikasi yang dapat diakses melalui web browser saat tersambung dengan internet. Sehingga, pengguna tidak harus menginstal aplikasinya pada ponsel.

C. PERBEDAAN LARAVEL VS CODEIGNITER

Saat membicarakan [framework PHP](#), tentunya Laravel bukanlah satu-satunya pilihan. Bahkan, di dunia ini tercatat ada lebih dari 40 framework PHP.

Laravel cocok jika Anda ingin membangun aplikasi web yang canggih dengan fitur modern. Sedangkan CodeIgniter lebih cocok untuk membuat aplikasi web sederhana yang menghadirkan fitur lengkap.

Grafik Penggunaan Framework php di dunia



Jadi, jika Anda ingin membuat aplikasi web yang elegan dan kompleks, sebaiknya belajar Laravel. Namun jika aplikasi web Anda cukup sederhana, CodeIgniter bisa menjadi solusi.

D. APA YANG PERLU ANDA PAHAMAI SEBELUM BELAJAR LARAVEL?

1. PENGUNAAN HTML / CSS

HTML dan CSS merupakan dasar untuk membuat tampilan dan struktur alias front-end website. Dengan kata lain, HTML dan CSS membentuk kulit dan tulang website. Sedangkan untuk membangun otak alias logikanya, Anda bisa menggunakan PHP.

2. PENGGUNAAN PHP

Untuk belajar Laravel secara mendalam, pemahaman akan PHP sangat penting. Sebab, ini bahasa pemrograman yang Laravel gunakan. PHP adalah bahasa pemrograman dengan sistem server-side. Jadi, tugas PHP yaitu membangun back-end alias pemrograman pada sisi server.

PHP singkatan dari Hypertext Preprocessor, yaitu bahasa pemrograman yang dirancang khusus untuk aplikasi dan pengembangan website. Mulanya, PHP dikembangkan pada tahun 1994 dan rilis perdana pada 8 Juni 1995 oleh pengembang berketurunan Denmark-Kanada, Rasmus Lerdorf.

Awalnya PHP merupakan kependekan dari Personal Home Page (Situs personal). Dengan sifatnya yang open source, minat dari para developer untuk berkontribusi dalam pengembangan PHP semakin meningkat. Pada tahun 1995, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1995, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang PHP: Hypertext Preprocessor.

Biasanya, PHP disisipkan ke dalam dokumen HTML dan dipadukan dengan bahasa pemrograman lain seperti CSS.

a. Cara Kerja PHP

Client (web browser) mengirim sebuah permintaan HTTP ke web server, sebagai contoh index.php. PHP yang berada di web server akan memproses kode PHP untuk menghasilkan file HTML. Web server akan mengirim file HTML kembali ke client.

b. Kelebihan PHP

- Mudah dipelajari
- Terintegrasi dengan banyak database
- Open source
- Komunitas besar
- Mendukung layanan cloud
- Fleksibel
- Cross-platform

c. Kekurangan PHP

- Keamanan kurang baik
- Alat debugging terbatas
- Bukan bahasa serbaguna
- Tidak dapat mengubah core behavior

- Variabel PHP bersifat case-sensitive

d. Penulisan Script PHP

Penulisan script PHP bisa dilakukan dengan beberapa cara seperti di gambar berikut, Pengguna tinggal memilih cara yang diinginkan.

```
<? dan ?> atau
<?php dan ?> atau
<script language="php"> dan </script> atau
<% dan %>
```

Berikut merupakan penerapan script PHP.

```
<!DOCTYPE html>
<html lang="en">
<head>
|   <title>Document</title>
</head>
<body>
|   <?php
|       echo 'Hai, saya adalah contoh PHP!';
|   ?>
</body>
</html>
```

e. Variabel

- Digunakan untuk menyimpan sebuah value, data atau informasi
- Nama variabel diawali dengan tanda \$
- Panjang tidak terbatas
- Setelah tanda \$ diawali oleh huruf atau under-score (_). Karakter berikutnya bisa terdiri dari huruf, angka, dan karakter tertentu yang diperbolehkan (karakter ASCII dari 127 – 255).
- Bersifat case-sensitive.
- Tidak perlu dideklarasikan.
- Tidak boleh mengandung spasi.

f. Tipe Data

Tipe data adalah pengklasifikasian data berdasarkan jenis data tersebut yang digunakan dalam pemrograman. Adapun beberapa jenis tipe sebagai berikut:

- Boolean

Tipe data boolean adalah tipe data paling standar yang hanya menyatakan kebenaran, apakah TRUE (benar) atau FALSE (salah).

```
<?php
$x = false;
$y = true;
?>
```

- String

Tipe data string adalah tipe data yang berbentuk text. dan untuk cara penulisan tipe data string di letakkan di tengah-tengah tanda petik.

```
<?php
$tes = "Saya Pergi Kerja";
?>
```

- Integer

Tipe data integer adalah tipe data yang berbentuk angka yang berbentuk bilangan asli atau bilangan bulat. untuk penulisan tipe data Integer tidak perlu menambahkan tanda petik karena akan di baca sebagai string jika anda menambahkan tanda petik pada tipe data integer.

```
<?php
$bilangan_pertama = 12;
?>
```

- Float

Tipe data float atau di sebut juga tipe data double adalah tipe data yang berisi bilangan desimal. cara penulisannya hampir sama dengan tipe data integer karena tidak memerlukan tanda petik di awal dan di akhir isi variabel float.

```
<?php
$angka = 12.177;
?>
```

- Array

Array adalah sebuah tipe data yang menyimpan banyak isi di dalam sebuah variabel. jika kita ibaratkan array seperti sebuah tas yang di dalamnya bisa saja berisi pulpen, buku, penggaris dan lainnya. isi dari pada variabel array di tandai dengan masing-masing angka yang sudah di terapkan menurut urutannya.

```
<?php
$anggota = array("Andi","Budi","Joni");
?>
```

- Objek

Tipe data objek digunakan untuk membuat instan dari suatu kelas. Objek memiliki properti dan metode yang dapat diakses untuk melakukan berbagai tindakan.

```
<?php
class Mobil {
    public $warna;
    public $merk;
}
?>
```

- NULL

Tipe data NULL digunakan untuk menyimpan nilai kosong atau tidak ada. Tipe data ini sering digunakan ketika suatu variabel belum memiliki nilai yang ditetapkan.

```
<?php
$nama = NULL;
?>
```

3. DASAR FRAMEWORK MVC

Laravel menggunakan konsep MVC. MVC atau Model-View-Controller adalah konsep yang memisahkan aplikasi web menjadi tiga bagian:

- a. **Model** merupakan representasi dari database, table, ikon, dan file lainnya. Fungsinya untuk mengelola data, logika, dan pengaturan aplikasi;
- b. **View**, bertugas menyajikan tampilan yang bisa dipahami manusia sesuai perintah controller;
- c. **Controller**, berperan menghubungkan model dengan view. Controller akan menerima input dari view, mengolah komputasi dan data, kemudian memberikan respons untuk ditampilkan oleh view.

Dengan pemisahan antara tampilan (front-end) dengan controller (back-end), Anda akan lebih memahami bagaimana Laravel memproses setiap file dan resource yang digunakan.

E. SOFTWARE YANG DIBUTUHKAN & CARA INSTALASI

Supaya lancar saat menggunakan Laravel tutorial, pastikan server Anda memenuhi persyaratan berikut :

a. WEB SERVER

Web server yaitu sistem komputer untuk menyimpan data dan file penunjang website agar dapat ditampilkan pada web browser. Server ini akan Anda gunakan untuk men-deploy atau mengimplementasikan program.

Pada modul ini, akan dicontohkan cara untuk melakukan konfigurasi database menggunakan MySQL yang ada diaplikasi XAMPP. Sebelum melakukan konfigurasi database pada Laravel terlebih dahulu memastikan sudah melakukan instalasi XAMPP. Berikut ini adalah cara untuk melakukan instalasi XAMPP.

1. Download file master installer xampp disitus resmi:

Link ini adalah situs resmi untuk mendownload master installer xampp <https://www.apachefriends.org/download.html>, setelah mengunjungi halaman situs, kemudian pilihlah master installer yang versi terbaru, pada saat modul ini ditulis, versi xampp dapat dilihat pada gambar dibawah ini :



The screenshot shows the XAMPP for Windows download page. At the top, it says "XAMPP for Windows 8.0.30, 8.1.25 & 8.2.12". Below that is a table with three rows, each representing a different version of XAMPP with its corresponding PHP version and download links. The table has columns for Version, What's Included?, Checksum (MD5 and SHA1), and Size.

Version	What's Included?	Checksum	Size
8.0.30 / PHP 8.0.30	What's Included?	md5 sha1	Download (64 bit) 144 Mb
8.1.25 / PHP 8.1.25	What's Included?	md5 sha1	Download (64 bit) 148 Mb
8.2.12 / PHP 8.2.12	What's Included?	md5 sha1	Download (64 bit) 149 Mb

At the bottom of the page, there are links for "Requirements" and "More Downloads". A note at the bottom states: "Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms here."

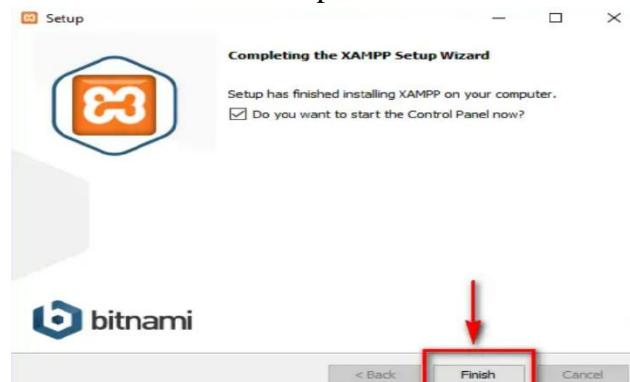
Pada gambar di atas dapat dilihat versi xampp yang terbaru yaitu 8.2.12 yang di mana versi Bahasa permrograman PHP dengan versi 8.2.12. untuk saat ini master installer xampp hanya dapat diinstall pada arsitektur operating system 64bit. Silahkan disesuaikan dengan operating system masing-masing.

2. Instalasi file master xampp

Setelah melakukan download file master xampp, selanjutnya melakukan instalasi. Untuk melakukankan instalasi dengan cara mengklik dua kali pada file xampp, kemudian menekan tombol Next sampai akhir instalasi.



Pada gambar di atas adalah tampilan awal untuk installasi xampp, silahkan klik tombok next, next, dan next sampai muncul tombol finish. Tombol finish muncul menandakan installasi aplikasi sudah selesai.



Untuk menyelesaikan installasi secara sempurna maka klik tombol finish, kemudian menjalankan beberapa service yang harus di aktifkan. Pada modul ini service yang harus diaktifkan yaitu apache dan mysql.



Pada gambar diatas menunjukkan service apache dan mysql sudah berjalan, untuk memastikan service-service tersebut dapat berfungsi dengan baik, maka cobalah mengakses link berikut : <http://localhost/phpMyadmin> di browser.

3. Jika terdapat pesan error pada saat mengklik tombol start, dapat melakukan beberapa perbaikan melalui tombol config. Salah satu contoh Ketika ingin menginstall Laravel, muncul pesan error seperti pada gambar dibawah ini:

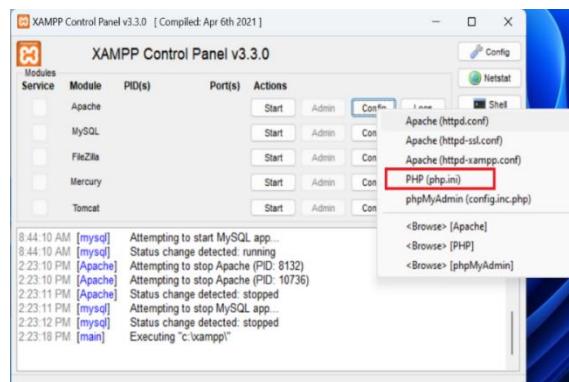
```

Note: command prompt
C:\Users\cad.. 
C:\>cd xampp\htdocs
C:\xampp\htdocs>composer create-project laravel/laravel sisinventario
Creating a "laravel/laravel" project at "/xampp/sisinventario"
Info from https://repo.packagist.org: #stashed
Installing laravel/laravel (v9.5.3)
  Failed to download laravel/laravel from dist: The zip extension and unzip() commands are both missing, skipping.
  The php.ini used by your PHP installation was: C:\xampp\php\php.ini
  The zip extension must be available from your system's PHP configuration.
  - Syncing [laravel/laravel] (v9.5.1) into cache
The following exception is caused by a process timeout.
Check https://getcomposer.org/doc/06-config.md#process-timeout for details.

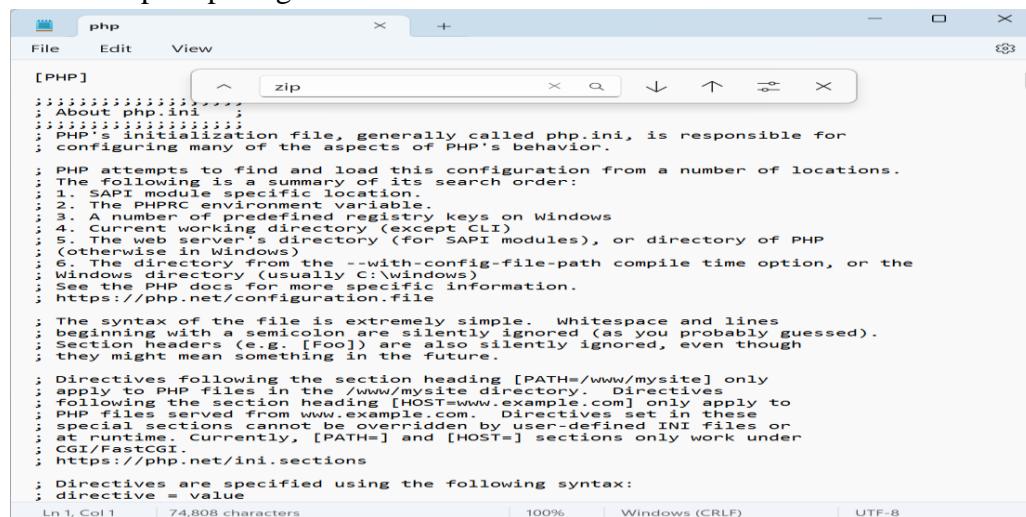
In Process.php line 1284:
  The process "git clone --mirror https://github.com/laravel/laravel.git C:/Users/DINESOTECH.COM/AppData/Local/Composer/vcs/github.com/laravel/laravel.git" exceeded the timeout of 300 seconds.

create-project [:s]:--stability STABILITY [:--prefer-source] [:--prefer-dist] [:--prefer-install PREFER-INSTALL] [:--repository REPOSITORY[:URL]] [:--add-repository] [:--dev] [:--no-dev] [:--no-custom-installers] [:--no-symlinks] [:--no-progress] [:--no-secure-http] [:--keep-vcs] [:--remove-vcs] [:--no-install] [:--no-modify] [:--add-format MODE] [:--no-interaction] [:--ignore-platform-requires] [:--ignore-platform-reqs] [:--no-ask] [:--no-verbose] [:--no-dev] [:--no-dev]
  
```

Pada gambar di atas muncul error dikarenakan ada extention yang perlu untuk diaktifkan pada file php.ini. cara untuk mengaktifkan bisa dilihat pada gambar dibawah ini:

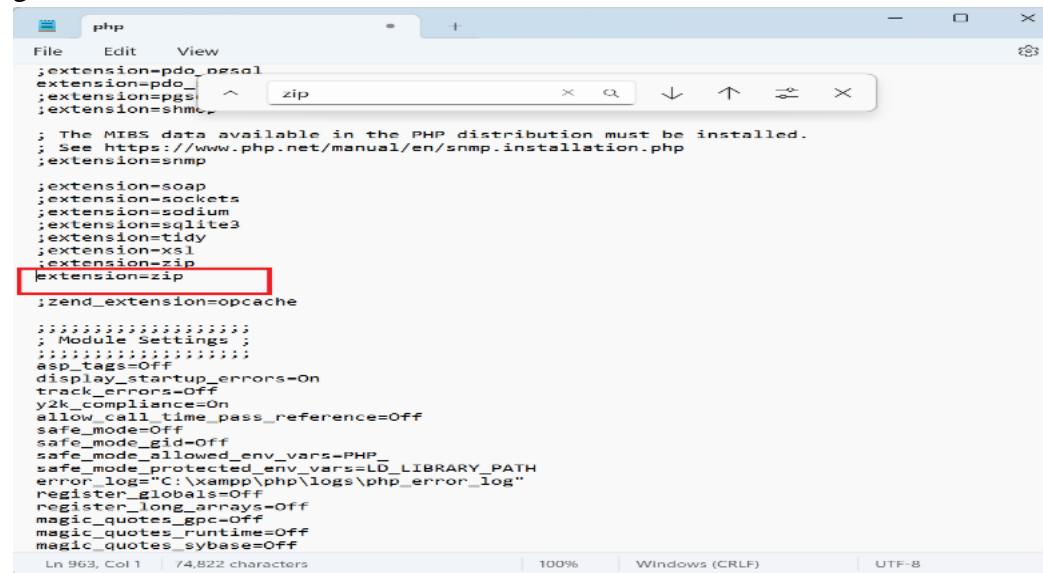


Pada gambar di atas, fokuslah pada tanda warna merah, Ketika diklik maka akan muncul seperti pada gambar dibawah ini:



Pada gambar diatas ketik CTR+F untuk memunculkan kolom pencarian, kemudian ketik zip sehingga nantinya langsung diarahkan pada baris extention yang diperlukan, setelah berada pada baris extention zip, hapuslah tanda titik koma pada

baris tersebut, sehingga extention zip menjadi aktif. Contohnya dapat dilihat pada gambar berikut ini :



```

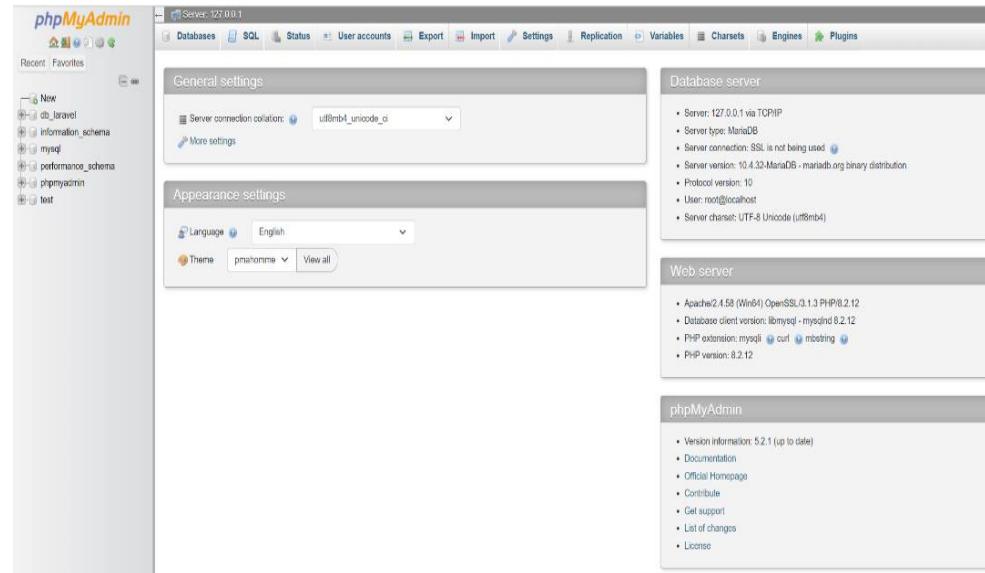
;extension=pdo_dbsql
;extension=pdo_
;extension=pgsql
;extension=shm_
;extension=snmp
;extension=soap
;extension=sockets
;extension=sodium
;extension=sqlite3
;extension=tidy
;extension=xsl
;extension=zip
zend_extension=opcache

;-----;
; Module Settings ;
;-----;
asp_tags=Off
display_startup_errors=On
track_errors=Off
y2k_compliance=On
allow_call_time_pass_reference=Off
safe_mode=Off
safe_mode_gid=Off
safe_mode_allowed_env_vars=PHP
safe_mode_protected_env_vars=LD_LIBRARY_PATH
error_log="C:\xampp\php\logs\php_error_log"
register_globals=Off
register_long_arrays=Off
magic_quotes_gpc=Off
magic_quotes_runtime=Off
magic_quotes_sybase=Off

```

Ln 963, Col 1 | 74,822 characters | 100% | Windows (CRLF) | UTF-8

Pada gambar diatas fokuslah pada tanda warna merah, dimana titik koma sudah dihapus. Kemudian simpanlah, cobalah Kembali menekan tombol Start. Setelah tidak ada masalah, maka cobalah akses link sesuai dengan penjelasan diawal. Setelah mengakses link, maka akan menampilkan seperti gambar dibawah ini.



General settings

- Server connection collation: utf8mb4_unicode_ci
- More settings

Appearance settings

- Language: English
- Theme: pmishonne
- View all

Database server

- Server: 127.0.0.1 via TCP/IP
- Server type: MySQL
- Server connection: SSL is not being used
- Server version: 10.4.32-MariaDB - mariadb.org binary distribution
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8mb4)

Web server

- Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12
- Database client version: libmysql - mysqld 8.2.12
- PHP extension: mysql curl mbstring
- PHP version: 8.2.12

phpMyAdmin

- Version information: 5.2.1 (up to date)
- Documentation
- Official Homepage
- Contribute
- Get support
- List of changes
- License

Pada gambar di atas menunjukkan bahwa instalasi xampp sudah berhasil dengan sempurna, untuk membuat database dan tabel akan dilakukan melalui framework Laravel, salah satu fitur Laravel yaitu Database Migration.

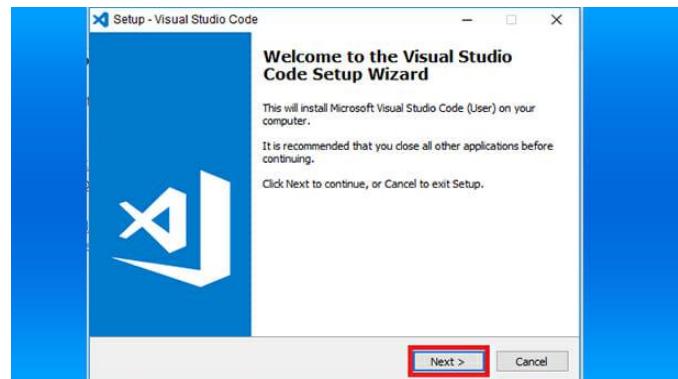
b. VISUAL STUDIO CODE

Berikut adalah cara untuk melakukan instalasi Composer di Sistem Operasi Windows:

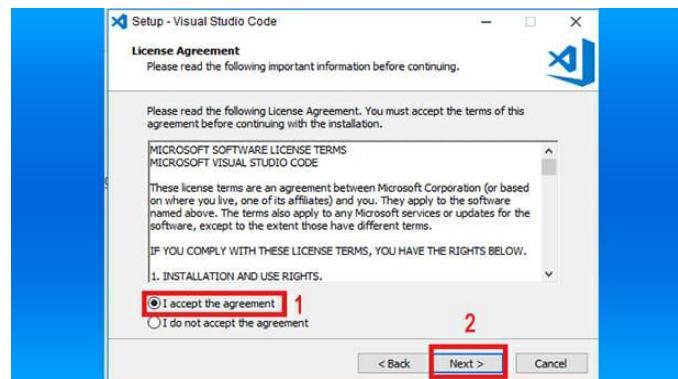
1. Unduh Visual Studio Code link <https://code.visualstudio.com/docs/setup/windows>



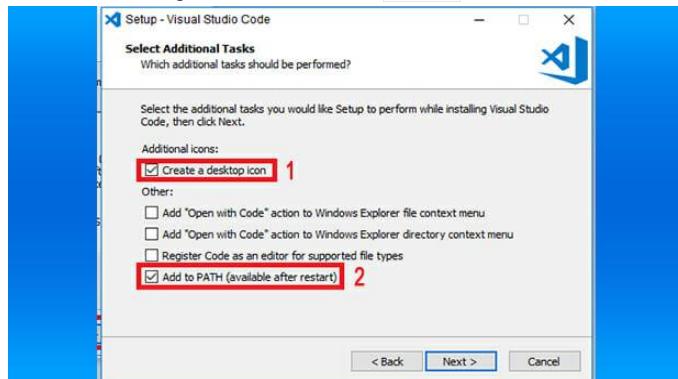
2. Jika sudah klik/buka Aplikasi VSCode yang sudah di download, Kemudian ikuti kotak merah pada gambar klik *Next*.



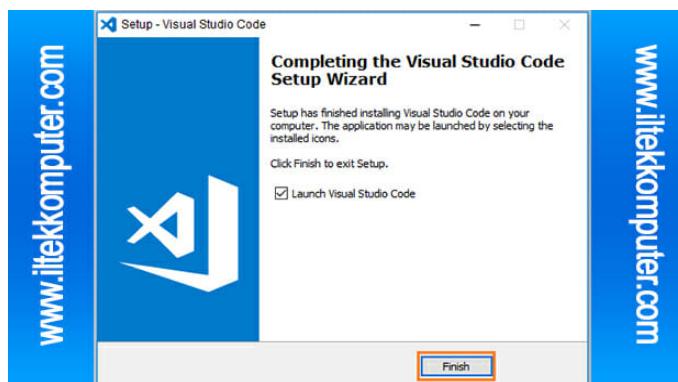
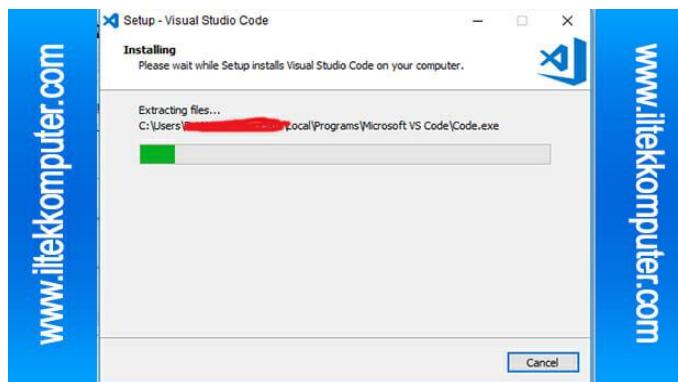
3. Kemudian pilih "*I accept the agreement*" untuk menyetujui kebijakan dari VSCode lalu klik *Next*.



- Kemudian klik *Next* lagi itu hanya penempatan direktori file programnya, lalu *Next* untuk membuat folder nya, Kemudian Ceklis “*Create Desktop Icon*” jika ingin membuat shortcut VSCode nya kemudian ceklis “*Add to PATH (available after restart)*” jika sudah klik *Next*.



- Kemudian Klik *Install* tunggu sampai “*Setup has finished installing Visual Studio Code on your komputer*” . kemudian klik *Finish* seperti gambar dibawah ini.



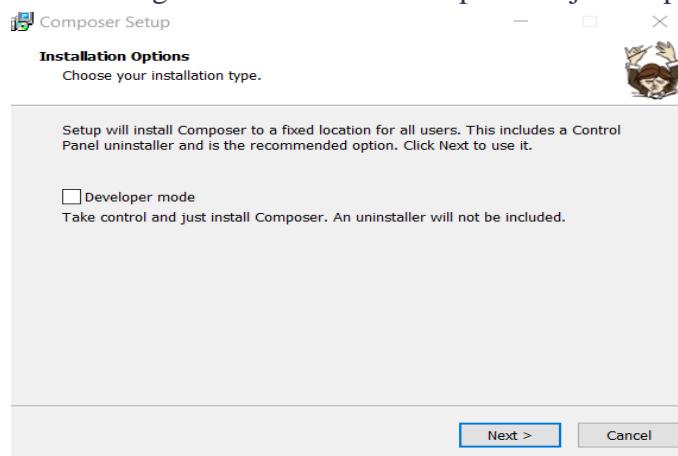
c. COMPOSER

Laravel menggunakan Composer untuk mengelola libraries yang dipakai Laravel dalam suatu proyek. Karena itu, Composer juga dijuluki sebagai *dependency* atau *library manager* untuk PHP.

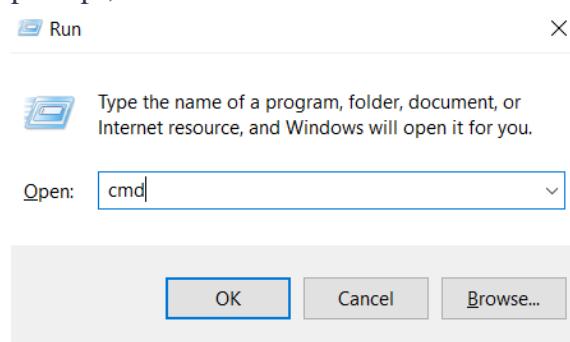
Dengan composer, pekerjaan Anda akan jauh lebih efisien. Sebab, composer mampu menyederhanakan pekerjaan sehingga Anda tidak perlu memberi perintah berulang kali. Selain itu, pemrograman PHP Anda juga jadi lebih rapi.

Berikut adalah cara untuk melakukan instalasi Composer di Sistem Operasi Windows:

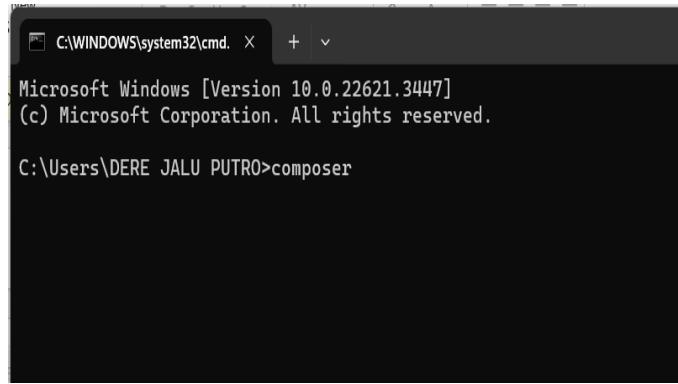
1. Unduh Composer versi terbaru pada link <https://getcomposer.org/Composer-Setup.exe>
2. Jalankan (run) **Composer** installation wizard. Abaikan jika Anda diminta untuk mengaktifkan mode developer. Lanjutkan proses install.



3. Muncul jendela yang akan meminta Anda untuk mencari baris perintah PHP. Secara default, baris perintah dapat ditemukan di **C:/xampp/php/pjp.exe**. Setelah itu, klik **Next**.
4. Permintaan **Pengaturan Proxy (Proxy Settings)** akan muncul. Kosongkan boks (tidak perlu ditandai) dan lewati bagian ini dengan meng-klik opsi **Next**. Klik **Install** di jendela berikutnya.
5. Buka command prompt. Tekan tombol **CTRL+R**, ketik “cmd” di dalam kolom prompt, dan klik **OK**.



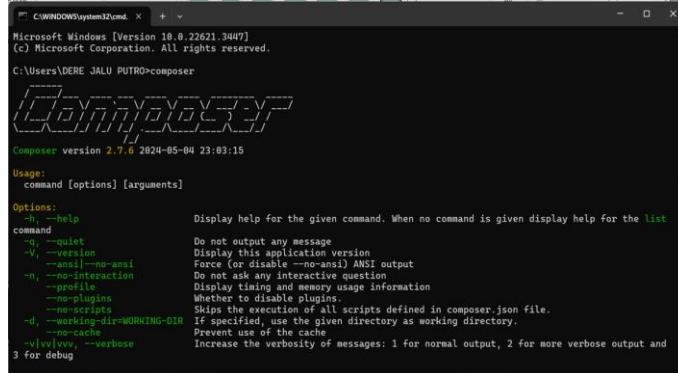
6. Setelah terinstall ketik composer pada CMD untuk mayakinkan bahwa composer telah berajalan dengan baik, Jalankan perintah berikut ini:



```
C:\WINDOWS\system32\cmd. X + v
Microsoft Windows [Version 10.0.22621.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DERE JALU PUTRO>composer
```

7. Berikut tampilan jika composere telah berjalan dengan baik.



```
C:\WINDOWS\system32\cmd. X + v
Microsoft Windows [Version 10.0.22621.3447]
(c) Microsoft Corporation. All rights reserved.

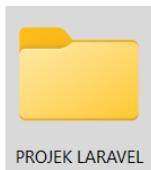
C:\Users\DERE JALU PUTRO>composer
 Composer version 2.7.6 2024-05-04 23:03:15
 Usage:
   command [options] [arguments]

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              Disable all available plugins
   --no-plugins              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
   -vvvvvvv, --verbose        Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and
   3 for debug
```

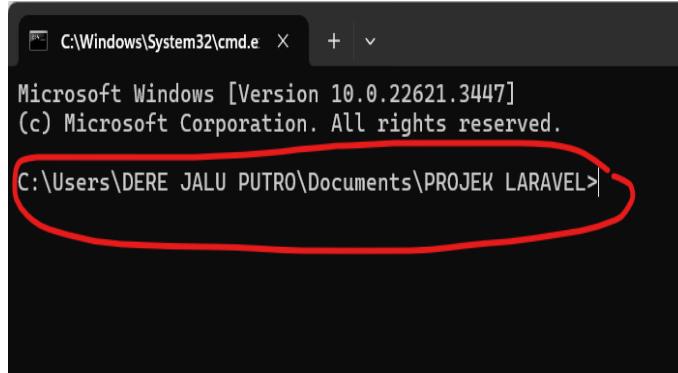
d. LARAVEL

Setelah Composer selesai terinstal, tahap berikutnya yaitu memasang Laravel.
Cara memasang laravel sebagai berikut :

1. Buat folder baru untuk projek kita dan masuk kedalam folder tersebut.



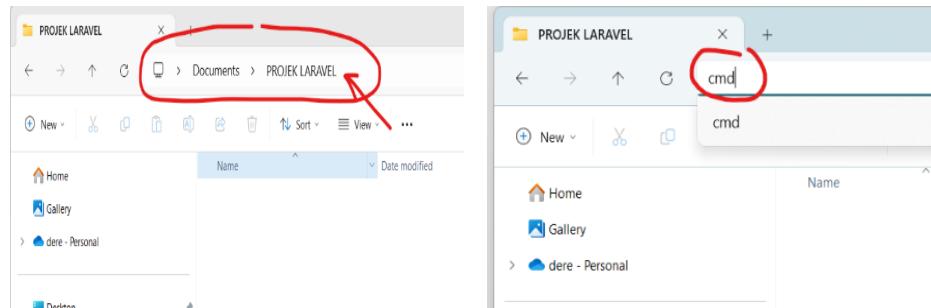
2. Buka CMD untuk masuk kedalam directory folder projek.



```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.22621.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DERE JALU PUTRO\Documents\PROJEK LARAVEL>
```

3. Shortcut tercepat untuk arahakan CMD ke directory adalah dengan mengklik link Alamat folder dan ketik CMD kemudian tekan Enter.



- Instal Laravel didalam folder projek yang sudah kita buat sebelumnya dengan perintah : composer create-project laravel/laravel *nama project*, dan pastikan computer kita terkoneksi internet dengan baik.

```
C:\Windows\System32\cmd.e > + <
Microsoft Windows [Version 10.0.22621.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DERE JALU PUTRO\Documents\PROJEK LARAVEL>composer create-project laravel/laravel sim_mini
```

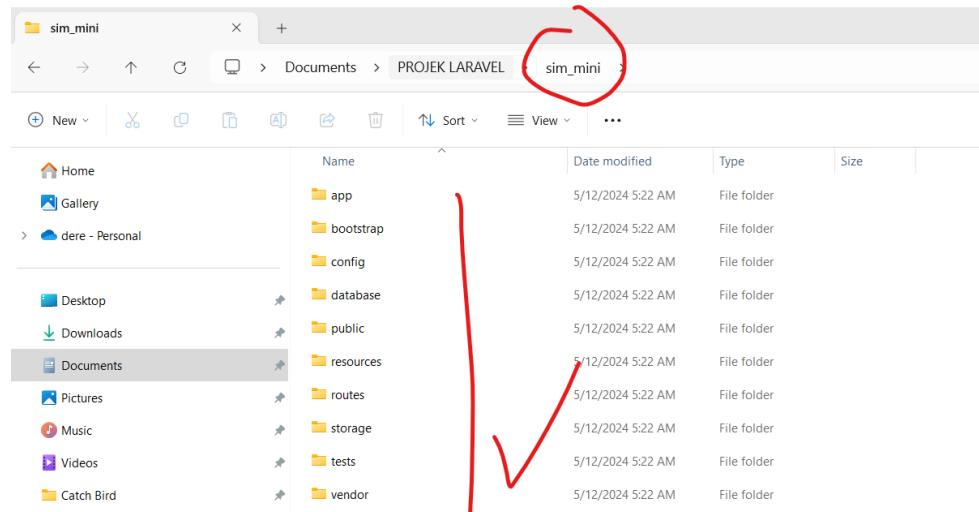
- Jika berhasil maka tampilan pada CMD akan seperti ini :

```
C:\Windows\System32\cmd.e > + <
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE

85 packages you are using are looking for funding.
Use the 'composer fund' command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force
[INFO] No publishable resources for tag [laravel-assets].
[INFO] No security vulnerability advisories found.
> @php artisan key:generate --ansi
[INFO] Application key set successfully.
> @php -r "file_exists('database/database.sqlite') || touch('database/database.sqlite');"
> @php artisan migrate --graceful --ansi
[INFO] Preparing database.
Creating migration table ..... 5.86ms DONE
[INFO] Running migrations.
0001_01_01_000000_create_users_table ..... 15.42ms DONE
0001_01_01_000001_create_cache_table ..... 4.47ms DONE
0001_01_01_000002_create_jobs_table ..... 10.67ms DONE

C:\Users\DERE JALU PUTRO\Documents\PROJEK LARAVEL>
```

- Tampilan dibawah ini adalah tampilan Ketika Laravel berhasil di download dan diterapkan pada folder projek kita.



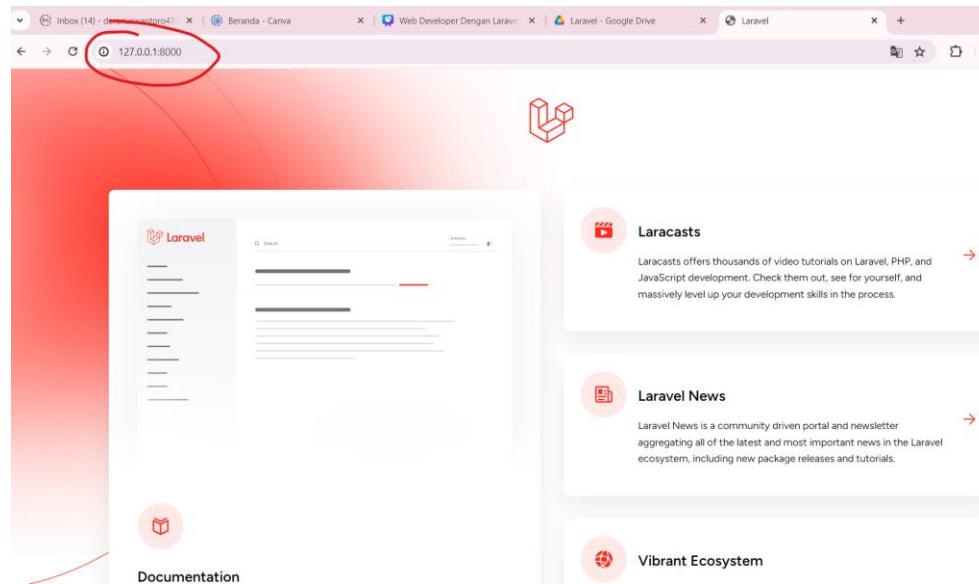
7. Kemudian kita masuk Kembali ke command prompt mengetik perintah dibawah ini, Untuk menjelaskan server localhost

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DERE JALU PUTRO\Documents\PROJEK LARAVEL\sim_mini>php artisan serve

[INFO] Server running on [http://127.0.0.1:8000].
Press Ctrl+C to stop the server
```

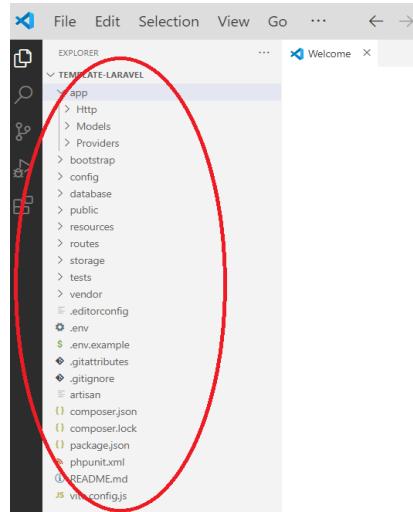
8. Tampilan awal saat anda memulai Laravel



MODUL II

DIRECTORY STRUCTURE LARAVEL

Laravel, sebagai salah satu framework PHP yang paling populer, memiliki struktur direktori yang terorganisir dengan baik. Laravel adalah salah satu framework PHP yang menggunakan konsep MVC (Model-View-Controller). Struktur direktori Laravel dirancang untuk memudahkan pengembangan aplikasi dengan mengorganisir kode berdasarkan fungsinya.

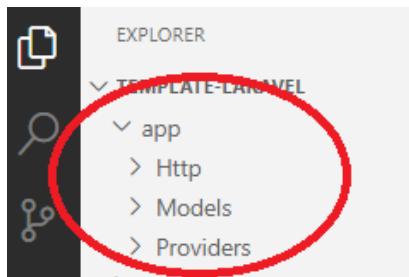


Berikut adalah penjelasan mengenai setiap direktori default pada Laravel dan bagaimana mereka berhubungan dengan konsep MVC:

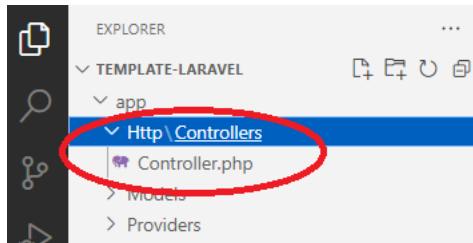
Struktur Direktori Laravel

- /app

Direktori ini berisi semua kode aplikasi, termasuk model, controller, dan sebagainya.



- /Http: Berisi controller, middleware, dan request.



- /Controllers: Berisi semua controller aplikasi.

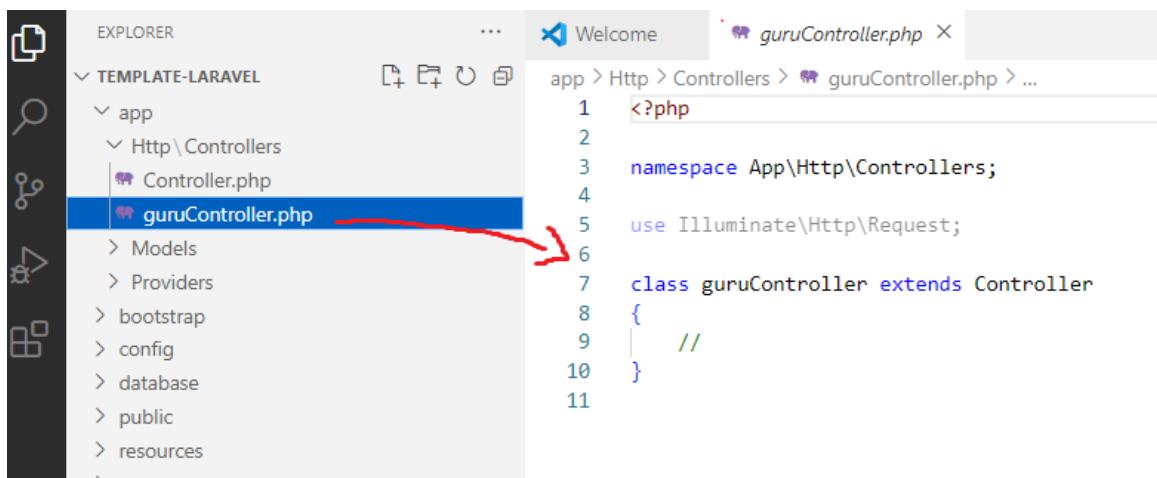
Perintah dibawah merupakan perintah untuk membuat file controller baru

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS DEVDB

```
PS C:\xampp\htdocs\template-laravel\template-laravel> php artisan make:controller guruController
```

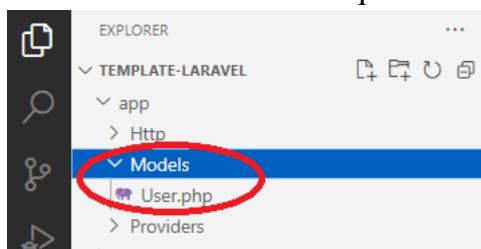
```
[INFO] Controller [C:\xampp\htdocs\template-laravel\template-laravel\app\Http\Controllers\guruController.php] created successfully.
```

Setelah perintah membuat controller baru sukses maka otomatis akan dibuatkan format controller.



- /models

Selanjutnya sub folder yang ada pada folder utama app yaitu folder Models. Pada folder ini berisi semua model Eloquent.



Di bawah ini merupakan perintah untuk membuat file model baru menggunakan perintah seperti diatas.

```
PS C:\xampp\htdocs\template-laravel\template-laravel> php artisan make:model guruModel
```

```
[INFO] Model [C:\xampp\htdocs\template-laravel\template-laravel\app\Models\guruModel.php] created successfully.
```

Setelah perintah membuat Model baru sukses maka otomatis akan dibuatkan format Model seperti di atas.

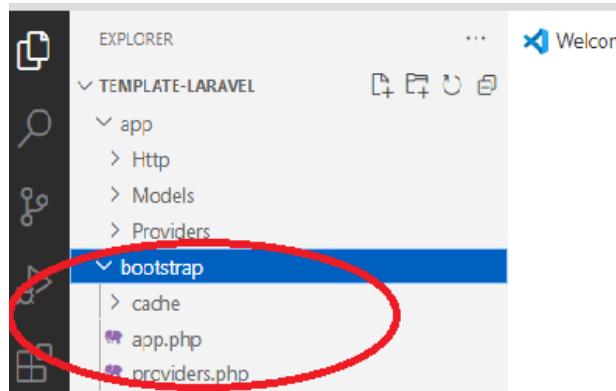
```

1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class guruModel extends Model
9 {
10     use HasFactory;
11 }

```

➤ /bootstrap

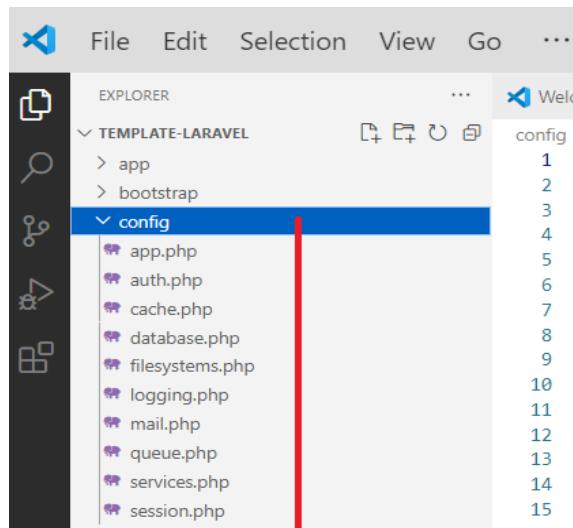
Bootstrap berisi file yang digunakan untuk bootstrap framework, seperti file app.php yang mem-bootstrapping framework.



➤ /cache: Berisi cache file aplikasi yang dibuat oleh framework.

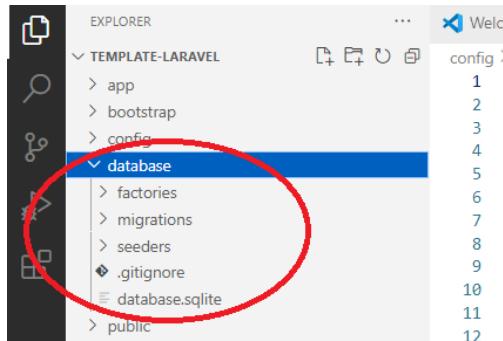
➤ /config

Config Berisi semua file konfigurasi aplikasi. Misalnya, app.php, database.php, mail.php, dll.



➤ /database

Berisi migrasi, seeder, dan factory untuk pengujian.



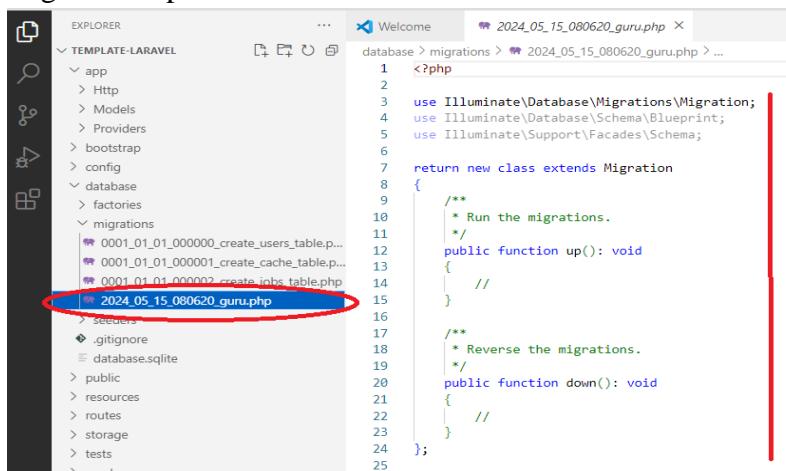
Untuk membuat file migration baru menggunakan perintah seperti dibawah :

PS C:\xampp\htdocs\template-laravel\template-laravel> php artisan make:migration guru

INFO Migration [C:\xampp\htdocs\template-laravel\template-laravel\database\migrations\2024_05_15_080620_guru.php] created successfully.

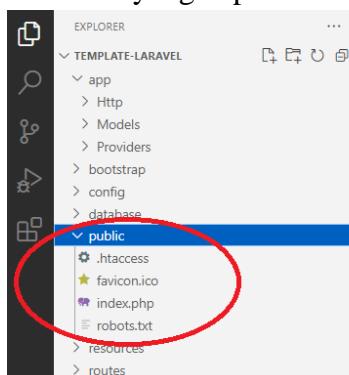
PS C:\xampp\htdocs\template-laravel\template-laravel> █

Setelah perintah membuat migration baru sukses maka otomatis akan dibuatkan format migration seperti diatas.



➤ /public

Direktori yang dapat diakses public



- File index.php utama yang menjadi entry point aplikasi, serta file asset publik seperti gambar, JavaScript, dan CSS.

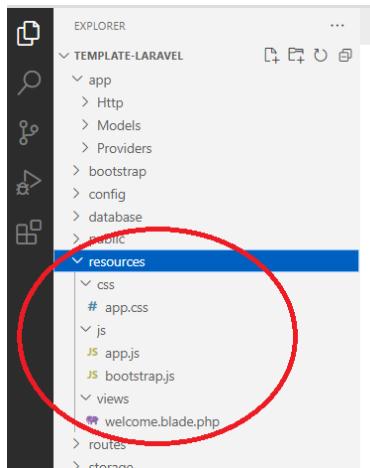
```

EXPLORER          ...  Welcome  index.php X
public > index.php > ...
1 <?php
2
3 use Illuminate\Http\Request;
4
5 define('LARAVEL_START', microtime(true));
6
7 // Determine if the application is in maintenance mode...
8 if (file_exists($maintenance = __DIR__.'/../storage/framework/maintenance.php')) {
9     require $maintenance;
10 }
11
12 // Register the Composer autoloader...
13 require __DIR__.'/../vendor/autoload.php';
14
15 // Bootstrap Laravel and handle the request...
16 (require_once __DIR__.'/../bootstrap/app.php')
17     ->handleRequest(Request::capture());
18

```

- /resource

/resource berisi resource view, file bahasa, dan asset yang tidak di-compile seperti SASS atau JavaScript. Pada folder view berisi file.blade.php yang dimana untuk pembuatan tampilan program.



- /views: Berisi semua template Blade (view) aplikasi.
- /js : Berisi file JavaScript yang digunakan dalam aplikasi.
- /css : Berisi file CSS (biasanya menggunakan preprocessor seperti SASS).

Perintah dibawah merupakan perintah untuk membuat file.blade.php baru.

```

PS C:\xampp\htdocs\template-laravel\template-laravel> php artisan make:view guru
INFO View [C:\xampp\htdocs\template-laravel\template-laravel\resources\views\guru.blade.php] created successfully.

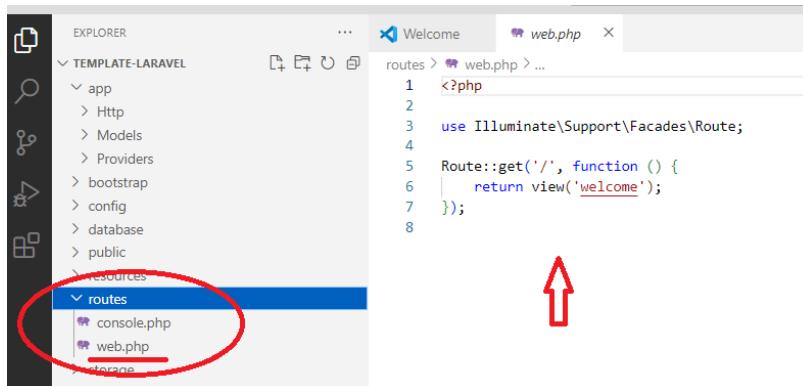
```

Setelah perintah membuat file blade baru sukses maka otomatis akan dibuatkan format .blade.php seperti di atas.



➤ /routes

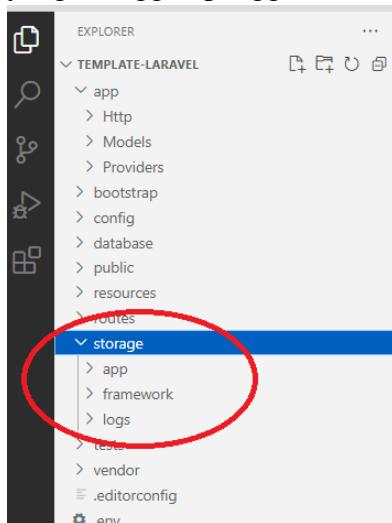
Berisi definisi rute aplikasi yang menghubungkan URL permintaan dengan controller yang tepat.



File web.php merupakan file yang digunakan untuk mendefinisikan rute-rute yang akan diakses melalui web browser. Di dalam file ini, Anda dapat mendefinisikan rute-rute untuk aplikasi Anda.

➤ /storage

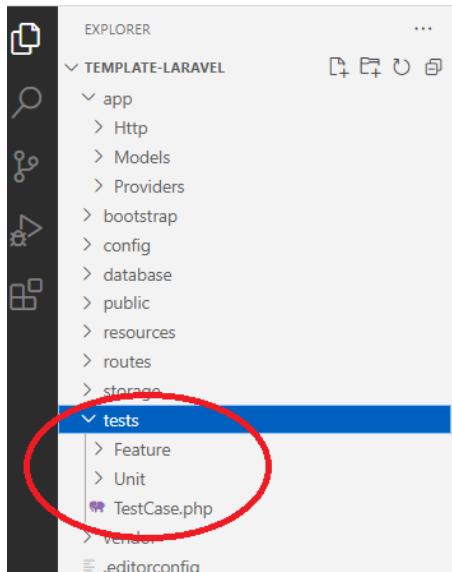
Direktori ini berisi file yang di-generate oleh framework, termasuk log, cache, dan file yang diunggah pengguna.



- /app : Berisi file aplikasi.
- /framework : Berisi file framework seperti cache dan sesi.
- /logs : Berisi log aplikasi.

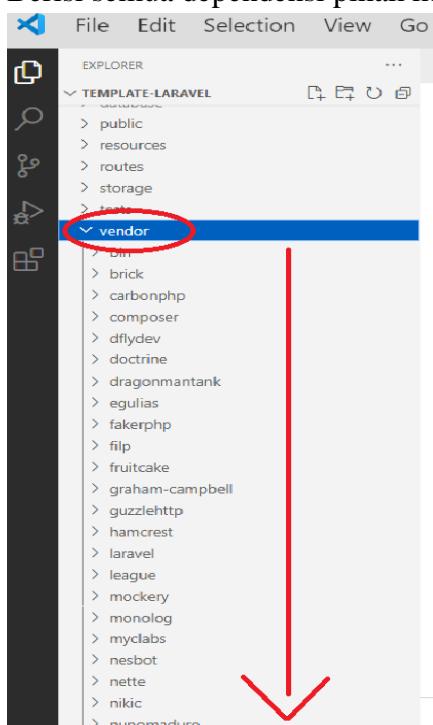
- /tests

Direktori ini berisikan test unit dan test fitur untuk aplikasi. Laravel menggunakan PHPUnit sebagai framework testing default.



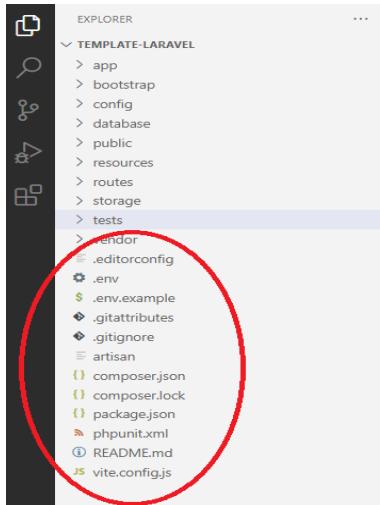
- /Feature: Berisi test fitur.
- /Unit : Berisi test unit
- /vendor

Berisi semua dependensi pihak ketiga yang di-install melalui Composer.



➤ / root directory

Berikut merupakan beberapa file yang ada pada root directory



- ❖ .env: File konfigurasi lingkungan yang berisi variabel-variabel seperti pengaturan database, konfigurasi mail, dan variabel-variabel lainnya.
- ❖ artisan: File ini adalah file CLI untuk menjalankan perintah-perintah Laravel.
- ❖ composer.json: File konfigurasi untuk Composer, yang digunakan untuk mengelola paket-paket PHP.

Dengan struktur ini memungkinkan pengguna menggunakan metode MVC (Model, View, Controller), dimana Laravel memisahkan logika aplikasi (model), logika tampilan (view), dan kontrol logika aplikasi (controller) secara jelas. Ini membuat pengembangan dan pemeliharaan aplikasi menjadi lebih mudah dan terorganisir.

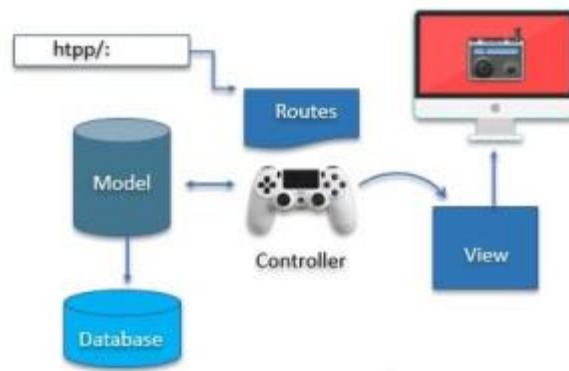
MODUL III

ELOQUENT ORM

A. DEFINISI ELOQUENT ORM

Eloquent adalah salah satu fitur di laravel, fitur ini untuk mengelola sebuah data yang ada pada database dan menjadikannya sangat mudah.

ORM (*Object Relational Mapping*) merupakan teknik yang digunakan untuk mengubah suatu table menjadi sebuah object, yang nantinya mudah untuk digunakan. Object yang dibuat memiliki property yang sama dengan field – field yang ada pada tabel dalam database.



Beberapa fitur utama dari Eloquent ORM:

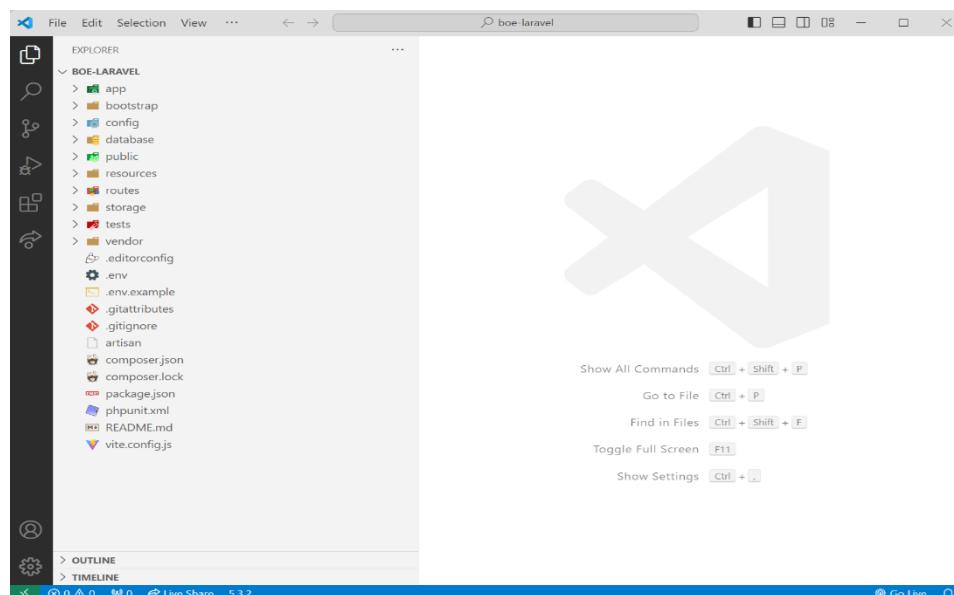
1. **Model:** Eloquent dapat merepresentasikan tabel dalam basis data. Pengembang dapat menentukan relasi antara model, seperti *one-to-one*, *one-to-many*, dan *many-to-many*.
2. **Query Builder:** Untuk membangun query basis data dengan sintaks yang ekspresif dan mudah dipahami. Ini memungkinkan pengembang untuk menulis query kompleks tanpa perlu menulis SQL mentah.
3. **Eloquent Collections:** Hasil query dikembalikan dalam bentuk koleksi yang merupakan objek terstruktur yang dapat diiterasi dan dimanipulasi dengan mudah.
4. **Soft Deletes:** memungkinkan penghapusan data dengan menandai data yang dihapus daripada menghapusnya secara permanen dari basis data.
5. **Event System:** Memungkinkan pengembang untuk menetapkan tindakan sebelum atau sesudah operasi dasar pada model, seperti menyimpan, menghapus, atau memperbarui.

B. MENERAPKAN ELOQUENT MODEL

Pastikan sudah menginstal Laravel dan mengkonfigurasi proyek. Jika belum melakukan penginstalan , dapat mengikuti panduan instalasi resmi laravel di : (<https://laravel.com/docs>).

C. Konfigurasi Database di Laravel 11

Sebelum melakukan konfigurasi tentunya harus melakukan instalasi framework terlebih dahulu, untuk dapat melakukan instalasi, ada beberapa prasyarat yang harus dilakukan terlebih dahulu yaitu mendownload dan menginstal aplikasi Composer dan NodeJS. Untuk kedua aplikasi ini akan dijelaskan pada bagian tersendiri. Pada konfigurasi database ini diharapkan sudah melakukan instalasi framework Laravel. Dibawah ini merupakan gambar apabila sudah berhasil melakukan instalasi Laravel.



Gambar diatas dapat dilihat struktur folder dan file pada Laravel, akan tetapi fokus pada file **.env**, folder **Migrations** yang ada didalam folder **Database/Migrations** dan folder **Model** yang ada didalam folder **App/Model**.

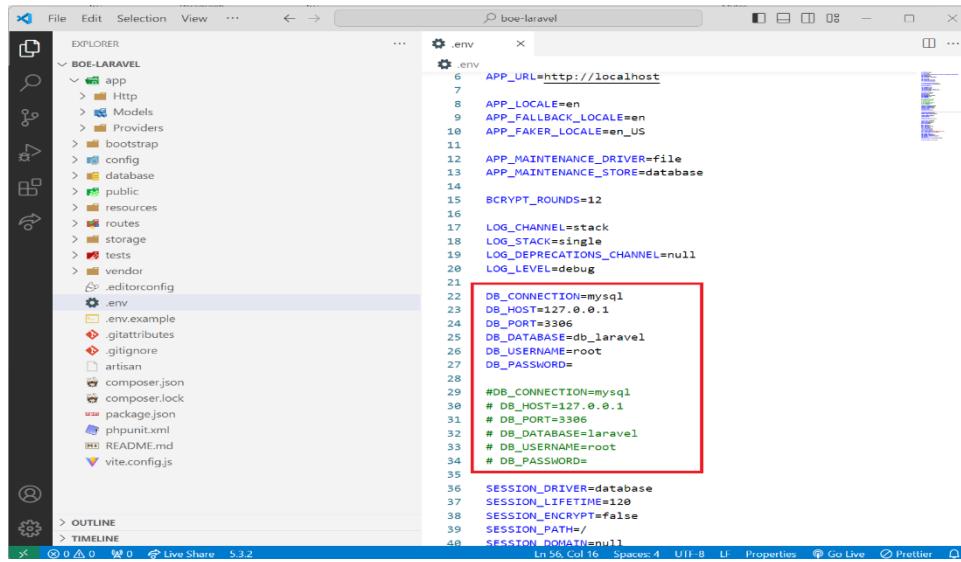
1. Konfigurasi file .env

Untuk pertama kalinya sebelum Laravel dapat terkoneksi ke database mysql yang ada di xampp, maka perlu melakukan deklarasi didalam file **.env**. Didalam file **.env** ini ada beberapa yang harus dirubah seperti :

- Menuliskan nama aplikasinya misalkan mysql, postgresql, sqlite, mongodb, dan lain-lain.

- b. Melakukan Uncomment pada baris yang masih ter comment secara default oleh Laravel.

Untuk lebih jelasnya lihatlah pada gambar dibawah ini.



```
File Edit Selection View ... ↵ ↶ boe-laravel
EXPLORER
BOE-LARAVEL
app
  Http
  Models
  Providers
  bootstrap
  config
  database
  public
  resources
  routes
  storage
  tests
  vendor
.editorconfig
.env
.env.example
.gitattributes
.gitignore
artisan
composer.json
composer.lock
package.json
phpunit.xml
README.md
vite.config.js

env
6 APP_URL=http://localhost
7
8 APP_LOCALE=en
9 APP_FALLBACK_LOCALE=en
10 APP_FAKE_LOCALE=en_US
11
12 APP_MAINTENANCE_DRIVER=file
13 APP_MAINTENANCE_STORE=database
14
15 BCRYPT_ROUNDS=12
16
17 LOG_CHANNEL=stack
18 LOG_STACK=single
19 LOG_DEPRECATED_CHANNEL=null
20 LOG_LEVEL=debug
21
22 #DB_CONNECTION=mysql
23 #DB_HOST=127.0.0.1
24 #DB_PORT=3306
25 #DB_DATABASE=db_laravel
26 #DB_USERNAME=root
27 #DB_PASSWORD=
28
29 #DB_CONNECTION=mysql
30 # DB_HOST=127.0.0.1
31 # DB_PORT=3306
32 # DB_DATABASE=laravel
33 # DB_USERNAME=root
34 # DB_PASSWORD=
35
36 SESSION_DRIVER=database
37 SESSION_LIFETIME=120
38 SESSION_ENCRYPT=false
39 SESSION_PATH=/
40 SESSION_DOMAIN=null

Ln 5b, Col 16 Spaces: 4 Ulf-8 U Properties Go Live Prettier
```

Pada gambar di atas fokuslah pada area yang tersorot warna merah, merupakan contoh bagian yang di comment dengan yang tidak comment, untuk bagian yang di comment itu tidak akan dieksekusi oleh Laravel. Untuk bagian yang tidak dicomment merupakan sebuah deklarasi dari nama database yang akan digunakan. Lihatlah penjelasan Pada tabel dibawah ini.

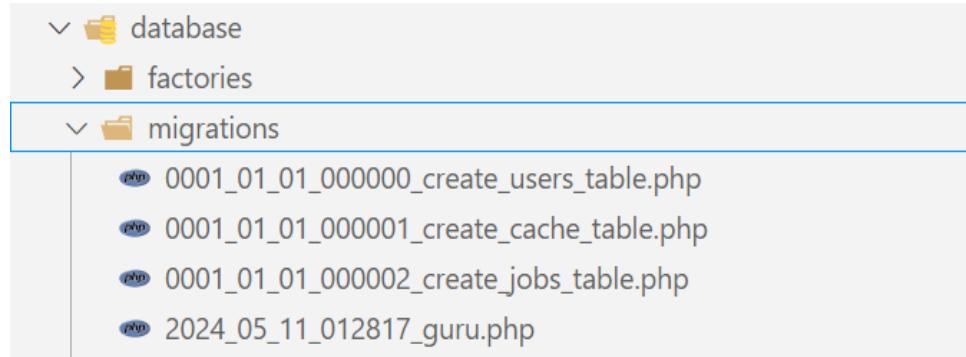
DB_CONNECTION=mysql	Mendefinisikan koneksi dengan nama aplikasi RDBMS yang digunakan, disini yang digunakan yaitu mysql
DB_PORT=3306	Nomor Port 3306 merupakan nomor default bawan dari mysql.
DB_HOST=127.0.0.1 atau DB_HOST=localhost	Mendefinisikan host, bisa dengan alamat IP atau dengan nama localhost, bisa dipilih salah satu
DB_DATABASE=db_laravel	Mendefinisikan nama database yang ingin dibuat, misalnya db_laravel atau db_sekolah atau db_kasir
DB_USERNAME=root	Untuk masuk kedalam mysql, username default nya itu root
DB_PASSWORD=	Untuk password defaultnya “null” atau tidak isi apapun.

2. Membuat tabel dengan nama tabel “**guru**” melalui artisan

Untuk membuat tabel disini bisa melalui Command Prompt (CMD) atau bisa melalui VS CODE, kemudian bisa mengetikkan perintah pada gambar dibawah ini :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS DEVDB powershell + 
PS C:\Users\Woodysnail\Desktop\boe-laravel> php artisan make:migration guru
```

Kemudian tekan Enter, maka hasilnya dapat dilihat pada folder **Database/Migrations**, seperti pada gambar dibawah ini :



Pada tahap ini, selanjutnya yaitu melakukan generate database serta tabelnya melalui Laravel, sebelum melakukan generate, terlebih dahulu melakukan deklarasi field-field apa saja yang ada di tabel guru, pada modul ini field yang akan digunakan sebagai contoh yaitu filed id, nama, mata_pelajaran. Bukalah file **2024_0511_012817_guru.php**, Listing kode programnya bisa dilihat pada gambar dibawah ini :

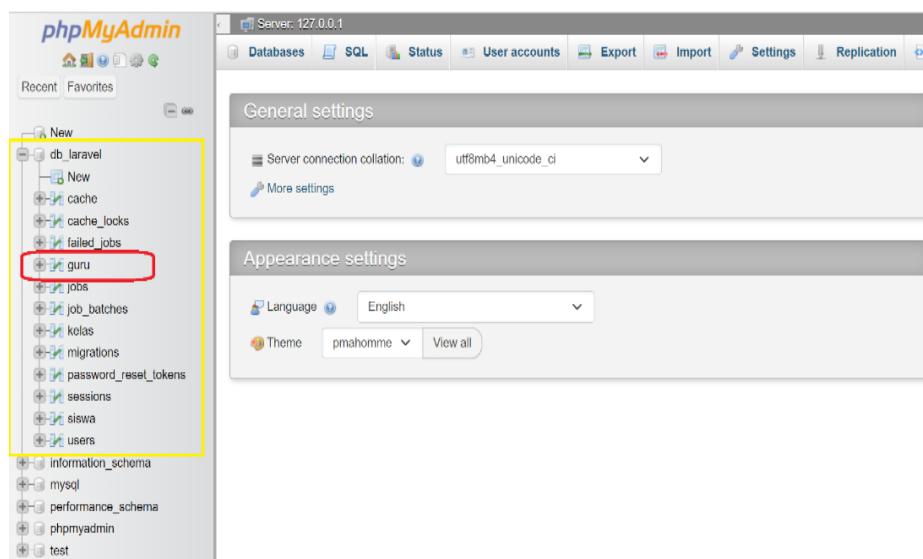
```
2024_05_11_012817_guru.php
database > migrations > 2024_05_11_012817_guru.php > class > up
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         /**
15          * Schema::create('guru', function (Blueprint $table) {
16          *     $table->id();
17          *     $table->string('nama');
18          *     $table->string('mata_pelajaran');
19          *     $table->timestamps();
20          * });
21      }
22
23      /**
24       * Reverse the migrations.
25       */
26     public function down(): void
27     {
28         /**
29          * Schema::drop('guru');
30      }
31 };
32 
```

Pada gambar diatas, fokuslah pada Listing kode yang ditandai warna merah, untuk syntax **Schema::create** merupakan syntax baku atau syntax yang sudah menjadi aturan dari Laravel itu sendiri tinggal mengikuti saja perintahnya. Untuk lebih jelasnya silahkan baca dokumentasi resmi dari Laravel pada bagian Database Migrations <https://laravel.com/docs/11.x/migrations>. Setelah medeklarasikan field-field tersebut, selanjutnya melakukan generate databse, tabel serta field-fielnya dengan perintah seperti pada gambar dibawah ini :



```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    DEVDB
PS C:\Users\Woodysnail\Desktop\boe-laravel> php artisan migrate
```

Untuk membuktikan apakah generate database, tabel, dan field berhasil, cobalah kembali membuka phpMyAdmin seperti link yang sudah dijelaskan, maka akan nampak sebuah database dengan nama **db_laravel**, dan tabel **guru**, adapun tabel-tabel lain yang ada di database **db_laravel** merupakan tabel default yang memang harus ada dan di migrate pertama kali. Apabila tabel-tabel tersebut ingin di hapus ataupun ingin melakukan custimisasi setelah melakukan migrate menjadi opsi masing-masing, silahkan perhatikan gambar dibawah ini :



Pada gambar diatas dapat dilihat pada tanda warna kuning dan merah, cobalah fokus ke warna merah, kemudian klik icon + untuk melihat struktur file yang sudah didefinisikan sebelumnya pada Laravel, coba perhatikan gambar dibawah ini :

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	nama	varchar(255)	utf8mb4_unicode_ci		No	None		Change Drop More	
3	mata_pelajaran	varchar(255)	utf8mb4_unicode_ci		No	None		Change Drop More	
4	created_at	timestamp			Yes	NULL		Change Drop More	
5	updated_at	timestamp			Yes	NULL		Change Drop More	
6	kelas_id	int(11)			Yes	NULL		Change Drop More	

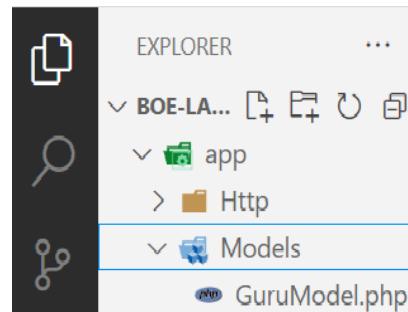
Sampai pada tahap ini artinya sudah berhasil melakukan migrations pada database, selanjutnya beralih untuk Langkah berikutnya yaitu membuat Model.

6. Membuat model dengan nama **GuruModel** melalui artisan

Untuk dapat membuat Model dengan nama GuruModel bisa membuat melalui Menu File VS CODE atau melalui Command Prompt (CMD) atau melalui Terminal yang ada di VS CODE, pada modul ini akan dicontohkan menggunakan Terminal VS CODE, untuk perintahnya dapat dilihat pada gambar dibawah ini :

```
PS C:\Users\Woodysnail\Desktop\boe-laravel> php artisan make:model GuruModel
```

Pada gambar diatas merupakan perintah untuk membuat sebuah file Model yang dimana nama Model nya adalah **GuruModel**, setelah menekan tombol **Enter**, maka cobalah perhatikan perubahan pada folder **App/Models**, akan terciptat sebuah file dengan nama **GuruModel**, hasilnya dapat dilihat pada gambar dibawah ini :



Pada gambar diatas menunjukan pembuatan file GuruModel sudah berhasil, selanjutnya bukalah file GuruModel yang selanjutnya mendefiniskan sebuah variable dengan nama tabel, untuk penamaan variable didahului dengan tanda dolar **\$tabel** , contoh dapat dilihat pada gambar dibawah ini :



```
 1  <?php
 2
 3  namespace App\Models;
 4
 5  use Illuminate\Database\Eloquent\Factories\HasFactory;
 6  use Illuminate\Database\Eloquent\Model;
 7  use Illuminate\Database\Eloquent\Relations\HasOne;
 8
 9  class GuruModel extends Model
10 {
11     use HasFactory;
12
13     protected $table = 'guru';
14
15
16
17 }
18
```

Pada gambar diatas dapat dilihat baris-barsi kode program, baris kode diatas secara automatis dibuat oleh Laravel kecuali pada baris kode **protected \$table = 'guru'**, baris kode tersebut ditulis secara manual yang artinya terlebih dahulu mendeklarasikan nama tabel yang sudah di migrate, kemudian untuk dapat berintraksi kedalam database mysql, sehingga untuk dapat **menyimpan, melihat, mengubah** dan **menghapus record** akan dijelaskan pada bagian terpisah.

MODUL IV

BACK END (MEMBUAT API)

A. Definisi Back End

Backend adalah bagian dari sebuah aplikasi atau sistem komputer yang berfungsi untuk menangani logika bisnis, penyimpanan data, dan interaksi dengan pengguna. Secara umum, backend bertanggung jawab atas pemrosesan data, menjalankan perhitungan, mengakses dan memanipulasi database, serta menyediakan layanan atau API (Application Programming Interface) untuk interaksi dengan frontend atau aplikasi klien lainnya.

Perbedaan Backend dan Frontend

Backend	Frontend
bagian yang tidak terlihat oleh pengguna	bagian yang terlihat dan dapat diakses pengguna
bertanggung jawab untuk menyimpan, memproses, dan mengirimkan data ke frontend	bertanggung jawab untuk menampilkan data kepada pengguna, menangani input pengguna, dan membuat interface pengguna yang interaktif dan menarik secara visual
Contoh bahasa yang digunakan: Java, Ruby, Phyton, dan .Net	Contoh bahasa yang digunakan: HTML, CSS, dan JavaScript

B. Definisi API

API (Application Programming Interface) adalah sekumpulan aturan, protokol, dan alat yang memungkinkan berbagai aplikasi perangkat lunak untuk berinteraksi satu sama lain. API menyediakan cara bagi pengembang perangkat lunak untuk menggunakan fungsionalitas atau layanan yang disediakan oleh aplikasi lain atau oleh sistem yang lebih besar.



Jenis – Jenis API :

API dapat berbentuk berbagai jenis, termasuk:

- ❖ RESTful API: Menggunakan HTTP request (GET, POST, PUT, DELETE) untuk mengakses dan memanipulasi data melalui URI (Uniform Resource Identifier).
- ❖ GraphQL: Sebuah query language dan runtime untuk mengambil data dari server. Menggunakan struktur tipe dan field untuk memungkinkan pengguna untuk menentukan struktur data yang mereka inginkan.
- ❖ SOAP (Simple Object Access Protocol): Sebuah protokol yang mendefinisikan cara pemanggilan prosedur atau metode dalam jaringan komputer, khususnya dalam web services.
- ❖ API Client Libraries: Kumpulan kode yang disediakan oleh pembuat API untuk mempermudah penggunaan API mereka dalam berbagai bahasa pemrograman.

C. Menerapkan Back End dan Restful API

1. Membuat File Api di dalam Laravel

Untuk membuat file Api.php di dalam Laravel, kita biasanya akan membuatnya sebagai bagian dari rute-rute API yang terpisah dari rute-rute web biasa. File Api.php akan berisi definisi rute-rute API yang akan digunakan dalam aplikasi Laravel kita.

Berikut adalah langkah-langkahnya:

- Buka terminal atau command prompt.
- Navigasikan ke direktori proyek Laravel kamu menggunakan perintah cd.
- Setelah berada di dalam direktori proyek Laravel, gunakan perintah Artisan untuk membuat file Api.php di dalam direktori routes.

bash

```
php artisan make:controller Api
```

Perintah ini akan membuat file Api.php di dalam direktori routes

2. Buka file routes/api.php dan tambahkan rute-rute yang telah kamu definisikan di dalam file Api.php. Contoh:

```

File Edit Selection View Go Run Terminal Help ⏎ → belajar-laravel
EXPLORER routes > api.php > ...
OPEN EDITORS web.php HomeController.php tambah.blade.php home.blade.php api.php layout.blade.php
BELAJAR-LARAVEL routes > ...
> database
> public
> resources
> views
> storage
> tests
> vendor
> .editorconfig
> .env
$ .env.example
> .gitattributes
> .gitignore
> artisan
routes > api.php
api.php
console.php
web.php
> storage
> tests
> vendor
> .editorconfig
> .env
$ .env.example
> .gitattributes
> .gitignore
> artisan
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS DEVDB
One new database migration has been published. Would you like to run all pending database migrations? (yes/no) [yes]:
> yes
INFO Running migrations.
    
```

Pada file Api. Php, kita buat route baru dengan nama api_guru

```

File Edit Selection View Go Run Terminal Help ⏎ → belajar-laravel
EXPLORER routes > api.php > ...
OPEN EDITORS web.php HomeController.php tambah.blade.php home.blade.php api.php layout.blade.php
BELAJAR-LARAVEL routes > ...
> database
> public
> resources
> views
> storage
> tests
> vendor
> .editorconfig
> .env
$ .env.example
> .gitattributes
> .gitignore
> artisan
routes > api.php
api.php
console.php
web.php
> storage
> tests
> vendor
> .editorconfig
> .env
$ .env.example
> .gitattributes
> .gitignore
> artisan
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS DEVDB
One new database migration has been published. Would you like to run all pending database migrations? (yes/no) [yes]:
> yes
    
```

Setelah membuat route, di file HomeController.php kita buat fungsi api_guru seperti gambar di bawah ini.

```

File Edit Selection View Go Run Terminal Help ← → belajar-laravel
EXPLORER ... web.php HomeController.php tambah.blade.php home
OPEN EDITORS
BELAJAR-LARAVEL
app > Http > Controllers > HomeController.php > ...
11 class HomeController extends Controller
12 {
13     public function index(Request $request)
14     {
15         $guru = GuruModel::get();
16
17         $data = array();
18         foreach($guru as $g) {
19             array_push($data, array(
20                 'nama' =>$g->nama,
21                 'mata_pelajaran' => $g->mata_pelajaran
22             ));
23         }
24
25         return response()->json($data);
26     }
27
28 }
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136

```

Melakukan pengecekan Restful Api kita, apakah sudah berjalan dengan baik dan manampilkan data, jika sudah akan muncul tampilan seperti gambar di bawah ini. Pengecekan di menggunakan aplikasi Postman

Search Postman

GET http://127.0.0.1:8000/api/guru

HTTP http://127.0.0.1:8000/api/guru

GET http://127.0.0.1:8000/api/guru

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON ↕

```

1 [
2   {
3     "nama": "Dere Jalu Putro Nuswantoro",
4     "mata_pelajaran": "Fotografi"
5   },
6   {
7     "nama": "Jualianty Ingani Rhapsudyah",
8     "mata_pelajaran": "Photoshop"
9   },
10  {
11    "nama": "Arya Adzan Amirulmu'minin",
12    "mata_pelajaran": "Corel Draw"
13  },
14  {
15    "nama": "Arsya Alqorni Nuswantoro",
16    "mata_pelajaran": "Matematika"
17  },
18  {
19    "nama": "Bayu Putra Anoraga",
20    "mata_pelajaran": "Video"
21  },
22  {
23    "nama": "Rizki Maulida",
24    "mata_pelajaran": "Pemrograman"
25  }
26 ]
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136

```

MODUL V

INSTALASI DAN KONFIGURASI REACT JS

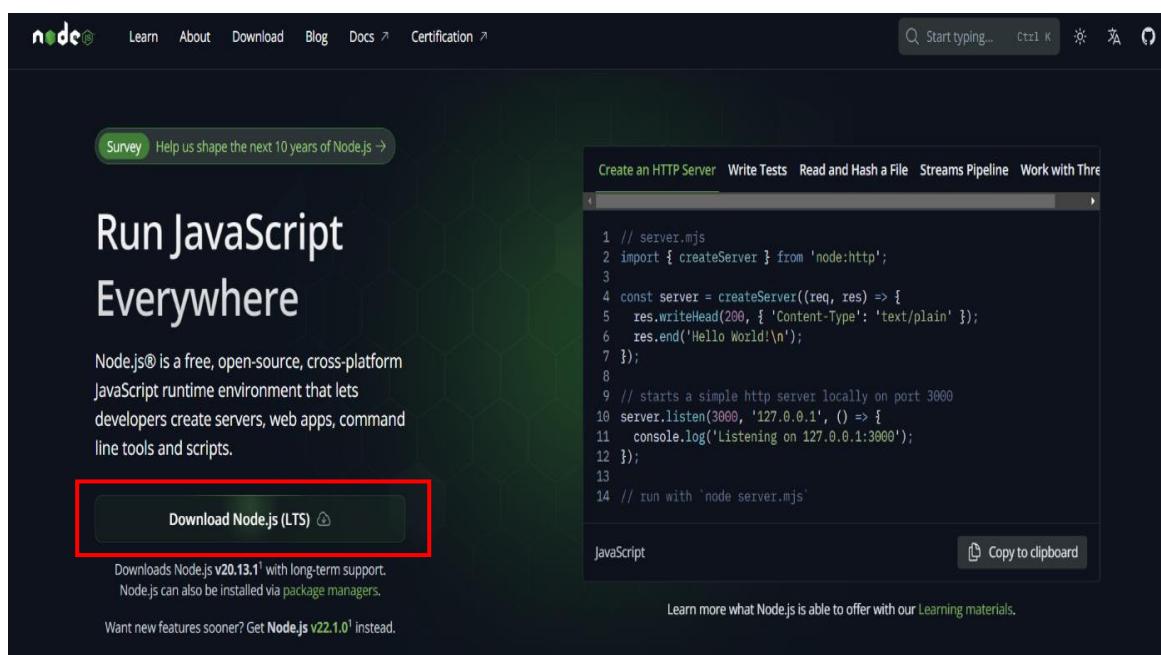
ReactJS adalah library JavaScript popular yang dibuat oleh Facebook yang digunakan dalam proses pengembangan aplikasi mobile dan web. untuk memfasilitasi pembuatan daripada komponen antarmuka yang interaktif, stateful, serta mudah untuk digunakan ulang. ReactJS sangatlah cocok digunakan untuk rendering antarmuka yang kompleks dengan performa tinggi (Kumar & Singh, 2016).

Dalam pengembangan, ReactJS memiliki berbagai macam kemampuan seperti dapat digunakan untuk membuat animasi suatu objek dengan efek transisi, menjalankan permainan di web browser yang sepenuhnya diprogram dengan menggunakan React. Selain itu, ReactJS juga mampu melakukan validasi form secara real-time saat pengguna mengisikan data pada form tersebut.

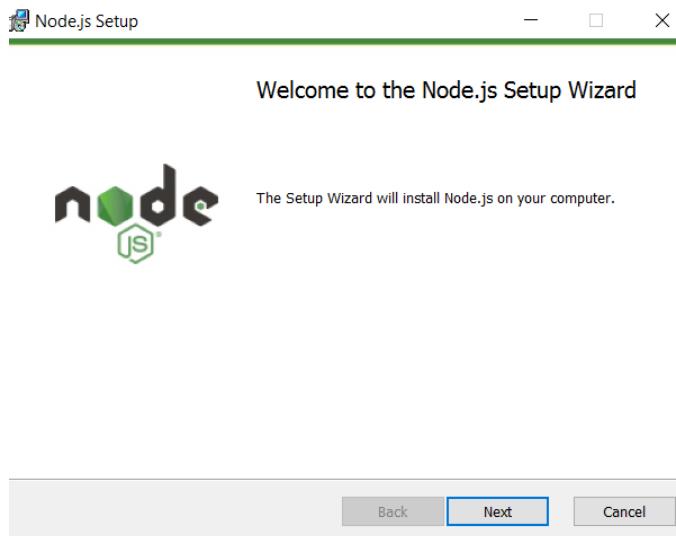
React bergantung pada Node.js dan npm (Node Package Manager). Untuk menjalankan hal tersebut perlu dilakukan proses untuk menginstal aplikasi Node.js seperti pada langkah-langkah dibawah ini:

Langkah 1: Instal Node.js dan npm

1. Kunjungi situs resmi Node.js di <https://nodejs.org/>
2. Unduh dan instal versi LTS (Long Term Support).



3. Selesai melakukan download aplikasi Node.js, dilanjutkan menginstal aplikasi tersebut hingga proses selesai



4. Setelah instalasi selesai, verifikasi instalasi dengan menjalankan perintah **node -v** dan **npm -v** di terminal atau command prompt:



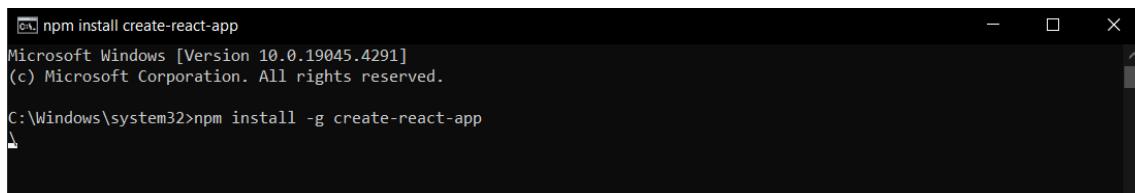
```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>node -v
v20.12.2

C:\Windows\system32>npm -v
10.7.0
```

Langkah 2: Instal Create React App

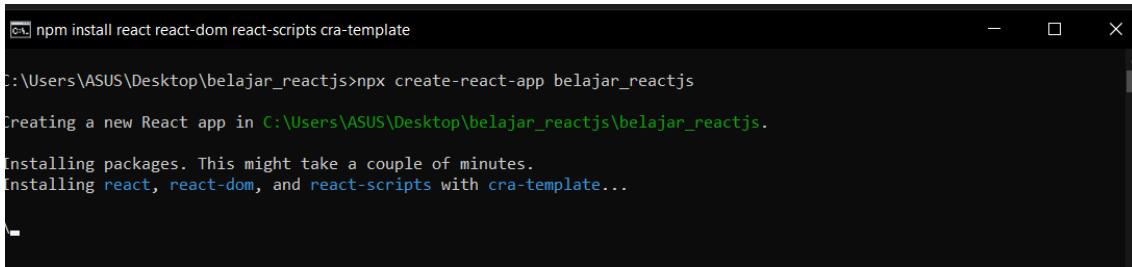
1. Buka terminal atau command prompt.
2. Jalankan perintah **npm install -g create-react-app** untuk menginstal Create React App secara global:



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>npm install -g create-react-app
\
```

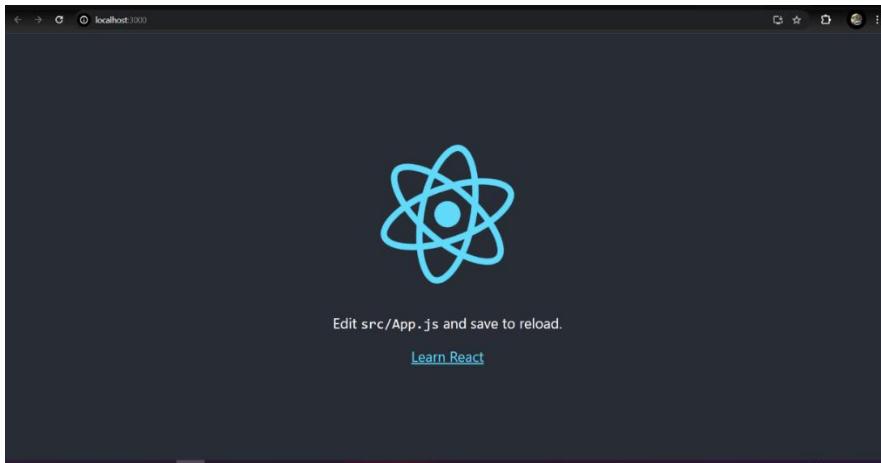
3. Buka direktori tempat anda akan membuat project react menggunakan cmd
4. kemudian ketikkan perintah **npx create-react-app (nama folder project)**



```
npm install react react-dom react-scripts cra-template
: \Users\ASUS\Desktop\belajar_reactjs>npx create-react-app belajar_reactjs
Creating a new React app in C:\Users\ASUS\Desktop\belajar_reactjs\belajar_reactjs.
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

```

5. Jalankan Visual Studio Code, kemudian open folder yang sudah di install React.
6. Untuk melihat hasil tampilan, buka terminal pada visual studio code dan ketikkan **npm start** sehingga akan muncul tampilan pada browser

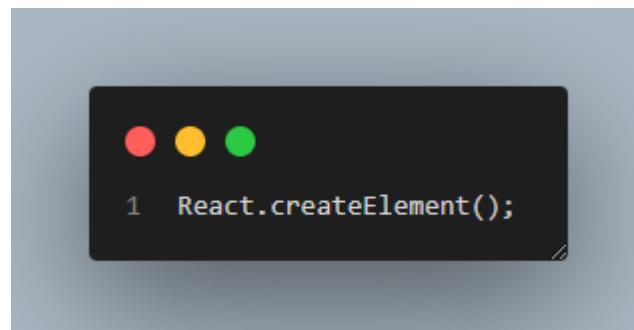


MODUL VI

PENGENALAN JSX

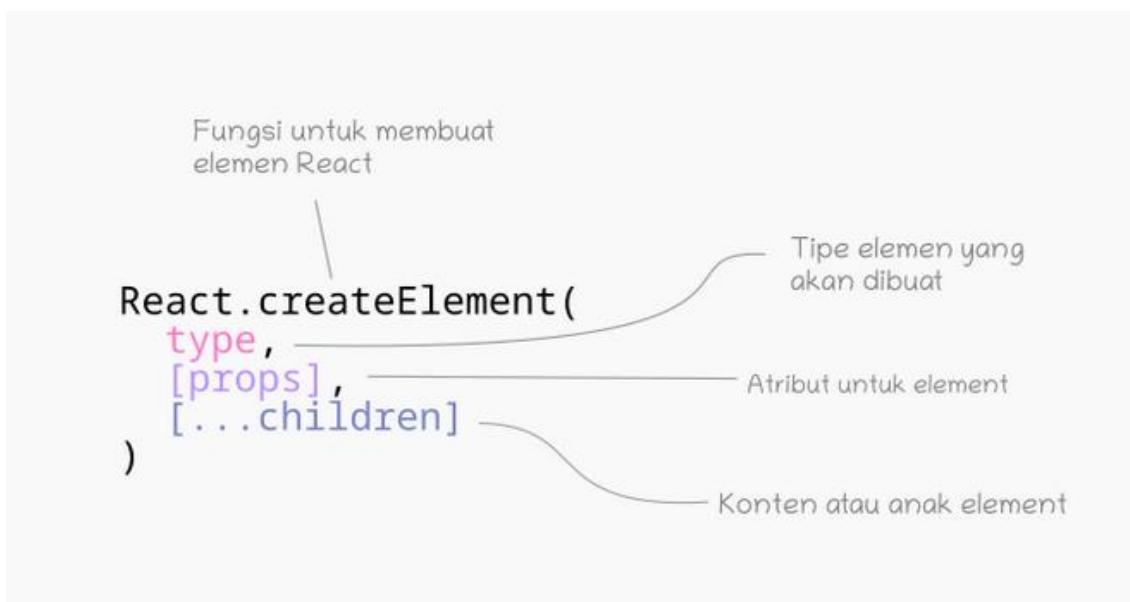
JSX adalah singkatan dari JavaScript XML merupakan sintaks ekstensi untuk JavaScript yang memungkinkan penulisan kode HTML di dalam React. Penulisan menggunakan JSX memudahkan dalam menulis elemen HTML dan menempatkannya di DOM tanpa harus menggunakan fungsi createElement() atau appendChild() sehingga mengubah tag HTML menjadi react elemen. Penggunaan JSX pada dasarnya bukan suatu keharusan atau kewajiban namun JSX mempermudah dalam penulisan coding pada React.

Penggunaan tanpa JSX, pengguna dapat membuat elemen React dengan method seperti berikut :

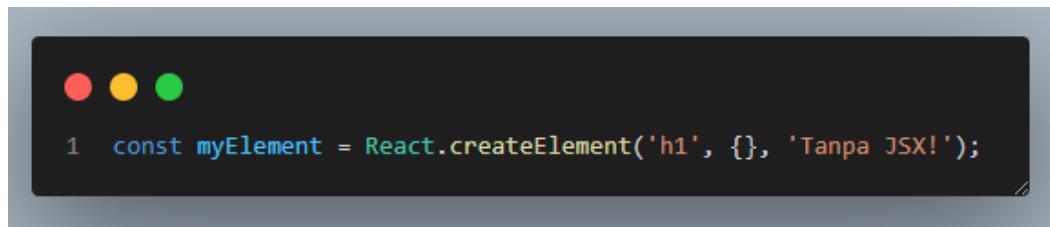


Dalam method diatas memiliki tiga parameter utama yang wajib untuk diisi yaitu:

1. Type elemen dalam bentuk string.
2. Properti element dalam bentuk object.
3. Children dalam bentuk string dan juga object react element.

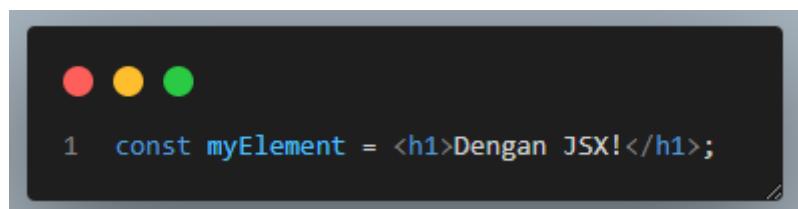


Untuk lebih jelasnya, berikut adalah contoh penulisan sintax React tanpa menggunakan JSX



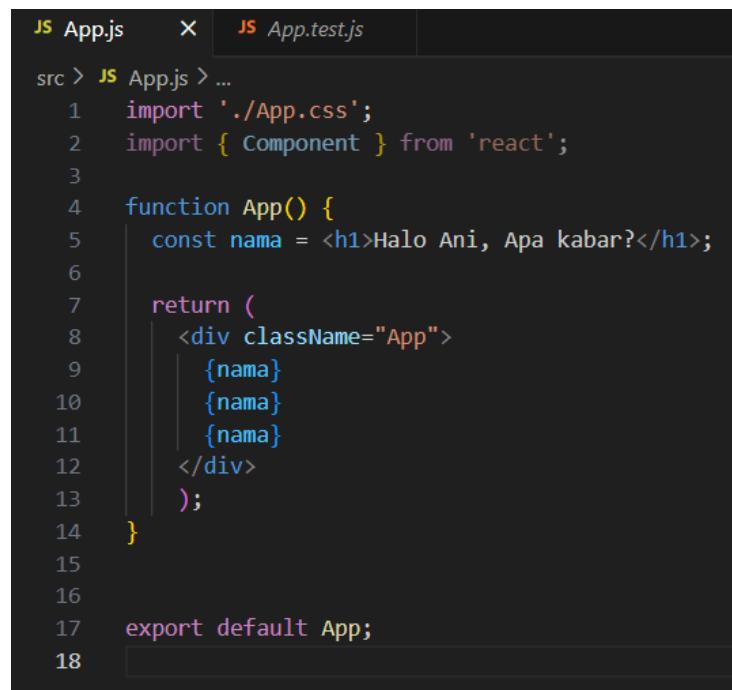
```
1 const myElement = React.createElement('h1', {}, 'Tanpa JSX!');
```

Adapun untuk penulisan penggunaan JSX yang benar adalah sebagai berikut



```
1 const myElement = <h1>Dengan JSX!</h1>;
```

Dari gambar diatas terlihat bahwa penulisan sintax atau script menggunakan JSX pada React lebih mempermudah atau menyingkat penulisan sintax. Disisi lain penggunaan JSX memiliki manfaat lain yaitu dapat dipanggil berulang-ulang seperti berikut :



```
JS App.js   X JS App.test.js
src > JS App.js > ...
1 import './App.css';
2 import { Component } from 'react';
3
4 function App() {
5   const nama = <h1>Halo Ani, Apa kabar?</h1>;
6
7   return (
8     <div className="App">
9       {nama}
10      {nama}
11      {nama}
12     </div>
13   );
14 }
15
16
17 export default App;
18
```

MODUL VII

COMPONENT, PROPS DAN STATE

1. Component (Komponen)

Komponen adalah blok bangunan dasar dalam React yang memungkinkan pemisahan UI menjadi bagian-bagian yang terisolasi. Komponen sendiri merupakan bagian terpenting dalam React karena React tersusun dari komponen-komponen itu sendiri.

Pembuatan komponen pada ReactJS dapat dilakukan dengan dua cara yaitu menggunakan fungsi dan yang kedua menggunakan class. Untuk komponen yang dibuat dengan fungsi disebut juga dengan function components dan untuk komponen yang dibuat menggunakan class disebut dengan class components.

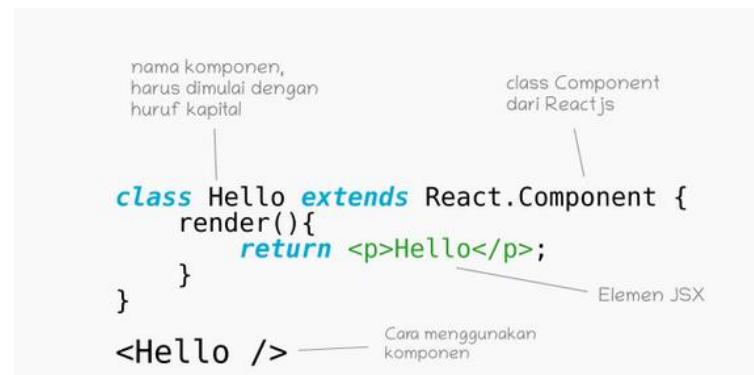
Penulisan komponen yang dibuat menggunakan fungsi adalah sebagai berikut :



Berdasarkan gambar diatas ada beberapa hal yang perlu diperhatikan karena didalam komponen terdapat beberapa hal yaitu

- Terdapat perintah function
- Terdapat nama komponen yang harus dimulai dengan huruf capital
- Terdapat elemen JSX
- Serta cara penggunaan komponen ketika dipanggil.

Adapun penulisan komponen berdasarkan class perlu melakukan extends dari class dengan cara sebagai berikut



Berdasarkan gambar diatas terdapat beberapa hal yang perlu diperhatikan dalam penulisan komponen yaitu :

- Terdapat perintah class
- Memiliki nama komponen dan dimulai dengan huruf capital
- Terdapat class component dari ReactJs
- Terdapat elemen JSX
- Cara penggunaan komponen

Kedua komponen diatas pada dasarnya sama namun yang membedakan adalah cara penulisannya. Untuk komponen pertama dibuat menggunakan fungsi dan komponen kedua menggunakan class.

Di dalam komponen terdapat sebuah method atau fungsi khusus yang terdapat pada class yang akan dieksekusi saat object dibuat atau pada saat komponen digunakan yaitu constructor. Constructor biasanya digunakan untuk melakukan inisialisasi komponen dan hanya pada komponen yang dibuat dengan class. Adapun penulisan constructor dan cara membuatnya didalam komponen adalah sebagai berikut :

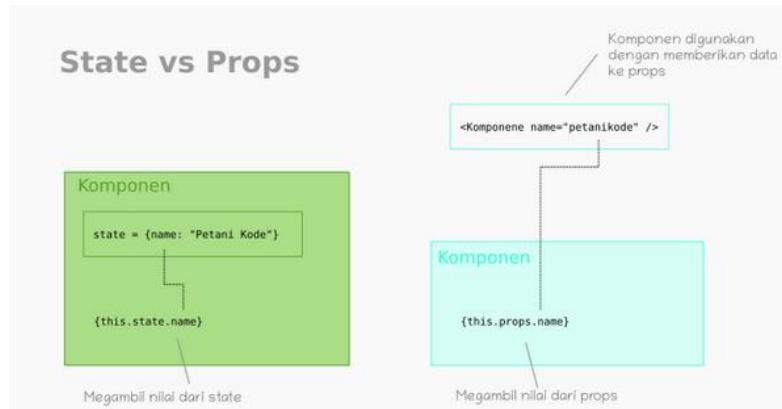
<pre> ● ● ● 1 class Data extends Component{ 2 constructor(props){ 3 super(props) 4 this.state={ 5 data : [6 {nama : 'David', nilai : 8}, 7 {nama : 'Aldo', nilai : 9}, 8 {nama : 'Rendi', nilai : 4}, 9] 10 } 11 } 12 render = () => { 13 return(14 <div class="list"> 15 <h2>Menampilkan Daftar Nilai</h2> </pre>	<pre> 15 <h2>Menampilkan Daftar Nilai</h2> 16 <table border="1" align="center"> 17 <tr> 18 <th>Nama</th> 19 <th>Nilai UTS</th> 20 </tr> 21 {this.state.data.map((uts)=> 22 <tr> 23 <td>{uts.nama}</td> 24 <td>{uts.nilai}</td> 25 </tr> 26)} 27 </table> 28 </div> 29); 30 } 31 } </pre>
---	---

Konstruktor dibuat dengan fungsi constructor() . Didalam konstruktor, pengguna bisa mendefinisikan nilai awal untuk variable maupun state. Pada konstruktor juga terdapat pemanggilan fungsi super() yang berada didalam fungsi konstruktor dan akan mengeksekusi konstruktor dari induk class yaitu Component dengan tujuan agar pengguna dapat mengakses semua method yang ada pada class induk.

2. Props dan State

State dan props adalah **objek khusus yang menyimpan data** untuk komponen. Kedua objek ini memiliki cara kerja yang berbeda. State adalah objek yang digunakan untuk

menyimpan **data di dalam komponen**, sedangkan props adalah objek yang digunakan untuk menyimpan **data yang diterima dari luar komponen**. Data yang disimpan di dalam state dapat di ubah-ubah, sedangkan data yang disimpan di dalam props tidak bisa diubah karena sifatnya *read-only*. Adapun konsep dari Props dan State adalah sebagai berikut



Adapun cara membuat props adalah sebagai berikut

```

1  class Menyapa extends Component{
2    render = () =>{
3      return(
4        <div>
5          Halo, saya {this.props.nama} menyapa dari class
6        </div>
7      )
8    }
9  }

```

Nilai pada props tidak bisa diubah karena sifat dari props yaitu hanya dapat dibaca saja atau *read-only*.

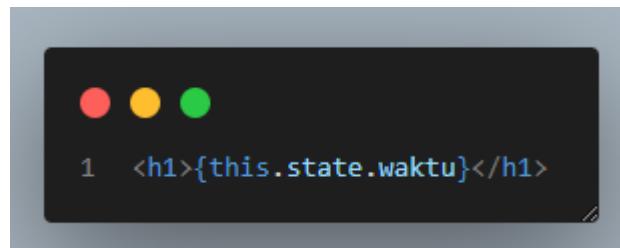
Sedangkan untuk penulisan state yang mana nilainya dapat berubah-ubah adalah sebagai berikut :

```

1  class Timer extends Component{
2    constructor(props){
3      super(props)
4      this.state={
5        waktu:0
6      }
7    }

```

Pengguna dapat mengambil nilai dari state dan elemen tersebut akan di render ulang apabila nilai state berubah dengan cara penulisan seperti berikut :



MODUL VIII

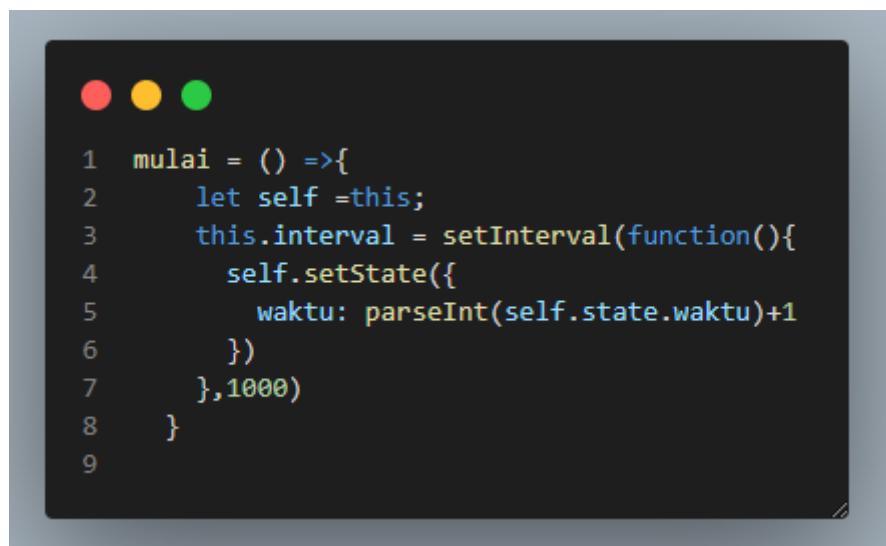
EVENT HANDLING

Setelah sebelumnya telah dipelajari terkait tentang props dan state, pada modul ini akan dibahas terkait event handling pada React Js. Dalam setiap aplikasi web, pengguna akan berinteraksi dengan elemen-elemen yang berada didalam web seperti button (tombol) atau formulir dan perlu ada proses untuk menangani interaksi tersebut. Event handling merupakan salah satu cara untuk menangani interaksi pengguna pada elemen-elemen tersebut.

Event handling adalah proses menangani **action** atau **interaction** yang dilakukan user pada element-element di dalam aplikasi web. Dalam [ReactJS](#), event handling bisa dilakukan dengan menambahkan **event listener** pada element tersebut.

Untuk menambahkan event handler pada component, kita perlu membuat function yang akan menangani interaksi user pada element seperti tombol, formulir, dan lain-lain. Kemudian, kita dapat menambahkan **event listener** pada elemen tersebut dan menghubungkannya dengan function event handler yang sudah dibuat.

Sebagai contoh, kita akan membuat sebuah tombol yang akan menampilkan pesan saat diklik. Pertama, kita buat sebuah function bernama mulai() yang akan menampilkan detik di saat tombol diklik:



```
1 mulai = () =>{
2     let self =this;
3     this.interval = setInterval(function(){
4         self.setState({
5             waktu: parseInt(self.state.waktu)+1
6         })
7     },1000)
8 }
9
```

Kemudian, kita tambahkan event listener pada element tombol dan hubungkan dengan function mulai() yang sudah dibuat:



```
● ● ●  
1 render = () =>{  
2  
3   return(  
4     <div>  
5       <h1>{this.state.waktu}</h1>  
6       <button onClick={this.mulai}>Mulai</button>  
7       <button onClick={this.pause}>Pause</button>  
8       <button onClick={this.stop}>Stop</button>  
9     </div>  
10   )  
11 }  
/
```

Dalam contoh ini, dapat menggunakan prop onClick untuk menambahkan event listener pada tombol. Prop ini akan menangkap event click pada tombol dan menjalankan function mulai(). Berhubung penulisan sintaks yang digunakan menggunakan class component maka penanganan event dapat dilakukan menggunakan method yang sudah disediakan oleh react yaitu this.setState().

MODUL IX

REACT-ROUTER

React Router adalah sebuah library yang digunakan dalam aplikasi React untuk membuat routing dan navigasi antar halaman. Dengan React Router, sebuah aplikasi web akan dapat memiliki banyak halaman dengan URL yang berbeda-beda. React Router sangat membantu dalam mengatur halaman-halaman tersebut dan memastikan bahwa setiap halaman ditampilkan dengan tepat.

Tanpa React Router, aplikasi React yang memiliki banyak halaman akan menjadi sangat sulit untuk dikelola. Pengembang harus membuat komponen untuk setiap halaman dan memastikan bahwa setiap komponen ditampilkan pada URL yang tepat. Dengan menggunakan React Router, memudahkan dalam memisahkan logika routing dari logika komponen sehingga dapat fokus pada pengembangan aplikasi itu sendiri.

Sebelum menggunakan React Router perlu dilakukan penginstalan React Router dengan menggunakan npm (Node Package Manager). Berikut ini adalah langkah-langkahnya:

1. Buka Terminal atau Command Prompt
2. Masukkan Perintah berikut : **npm install react-router-dom**
3. Tunggu hingga proses instalasi selesai

```
PS D:\SMK3\Pelatihan\Didik Web Technologi\Praktikum\belajar-reactjs2> npm i react-router-dom
added 3 packages, and audited 1559 packages in 13s

262 packages are looking for funding
  run `npm fund` for details

8 vulnerabilities (2 moderate, 6 high)

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
```

Setelah proses instalasi react router selesai dilakukan, maka dalam penulisan sintaks tidak akan bisa berjalan karena perlu dilakukan import library dari react router dengan menulis sintaks berikut pada project :

```
1 import { BrowserRouter, Route, Routes } from 'react-router-dom';
```

Sebelum lanjut membuat route, perlu dibuat beberapa tiga file js yang akan diroute yaitu DataGuru.js dan About.js dan setiap halaman akan menampilkan beberapa teks sederhana seperti berikut :

```
1 import { Component } from 'react';
2
3 class About extends Component{
4     render = () => {
5         return (
6             <div>
7                 <h1>About</h1>
8             </div>
9         )
10    }
11 }
12
13 export default About;
```

Pada sintaks diatas dapat disimpan dengan nama About.js sebab halaman tersebut yang akan di route nantinya. Hal yang sama juga dilakukan pada file dengan nama DataGuru.js seperti dibawah ini

```

 1 import { Component } from 'react';
 2
 3 class DataGuru extends Component{
 4     render = () => {
 5         return (
 6             <div>
 7                 <h1>Data Guru</h1>
 8             </div>
 9         )
10     }
11 }
12
13 export default DataGuru;

```

Selanjutnya yaitu mengintegrasikan React Router ke dalam proyek yang telah dikerjakan. Dalam komponen App.js akan ditentukan rute-rute yang akan digunakan dalam aplikasi yang dibuat dengan sintaks dibawah ini :

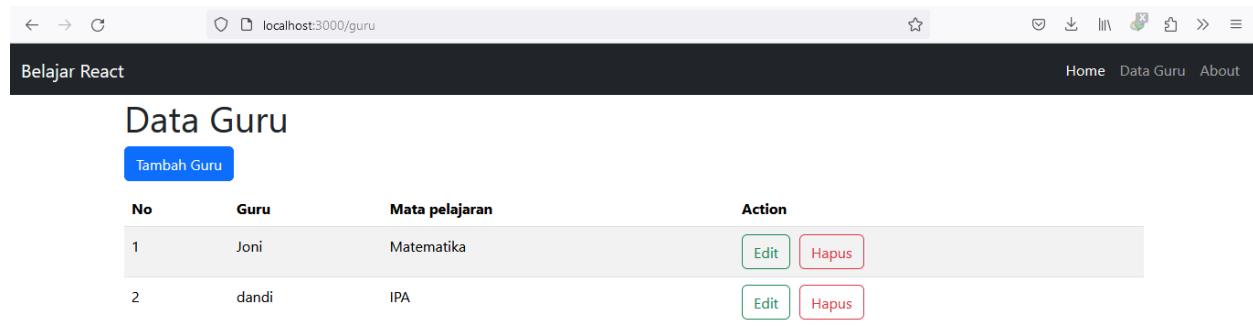
```

1 import { BrowserRouter, Route, Routes } from 'react-router-dom';
2 import DataGuru from './DataGuru';
3 import About from './About';
4 import TambahGuru from './TambahGuru';
5
6 function App() {
7
8     return (
9         <div className="">
10             <nav className="navbar navbar-expand-lg navbar-dark bg-dark">
11                 <div className="container">
12                     <a className="navbar-brand" href="#">Belajar React</a>
13                     <button className="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
14                         <span className="navbar-toggler-icon"></span>
15                     </button>
16                     <div className="collapse navbar-collapse d-flex justify-content-end" id="navbarNav">
17                         <ul className="navbar-nav">
18                             <li className="nav-item">
19                                 <a className="nav-link active" aria-current="page" href="/">Home</a>
20                             </li>
21                             <li className="nav-item">
22                                 <a className="nav-link" href="/guru">Data Guru</a>
23                             </li>
24                             <li className="nav-item">
25                                 <a className="nav-link" href="/about">About</a>
26                             </li>
27                         </ul>
28                     </div>
29                 </div>
30             </nav>
31             <div className="container">
32                 <BrowserRouter>
33                     <Routes>
34                         <Route path="/" element={''} />
35                         <Route path='/guru' element={[ <DataGuru /> ]}> />
36                         <Route path='/about' element={[ <About /> ]}> />
37                         <Route path='tambah_guru' element={[ <TambahGuru /> ]}> />
38                     </Routes>
39                 </BrowserRouter>
40             </div>
41         </div>
42     );
43 }
44
45 export default App;

```

Pada kode diatas akan menampilkan beberapa komponen dimana untuk URL dari element Home adalah “/” atau disebut juga dengan nama root. Untuk URL dari element DataGuru adalah /dataguru dan adapun hasilnya adalah sebagai berikut :

Activate Windows
Go to Settings to activate Windows.



No	Guru	Mata pelajaran	Action
1	Joni	Matematika	<button>Edit</button> <button>Hapus</button>
2	dandi	IPA	<button>Edit</button> <button>Hapus</button>



About

Dari gambar diatas terlihat bahwa ketika pengguna memilih data guru maka akan pindah ke halaman data guru, begitu pula ketika memilih about ataupun home maka akan diarahkan sesuai tujuan yang dipilih.

MODUL IX

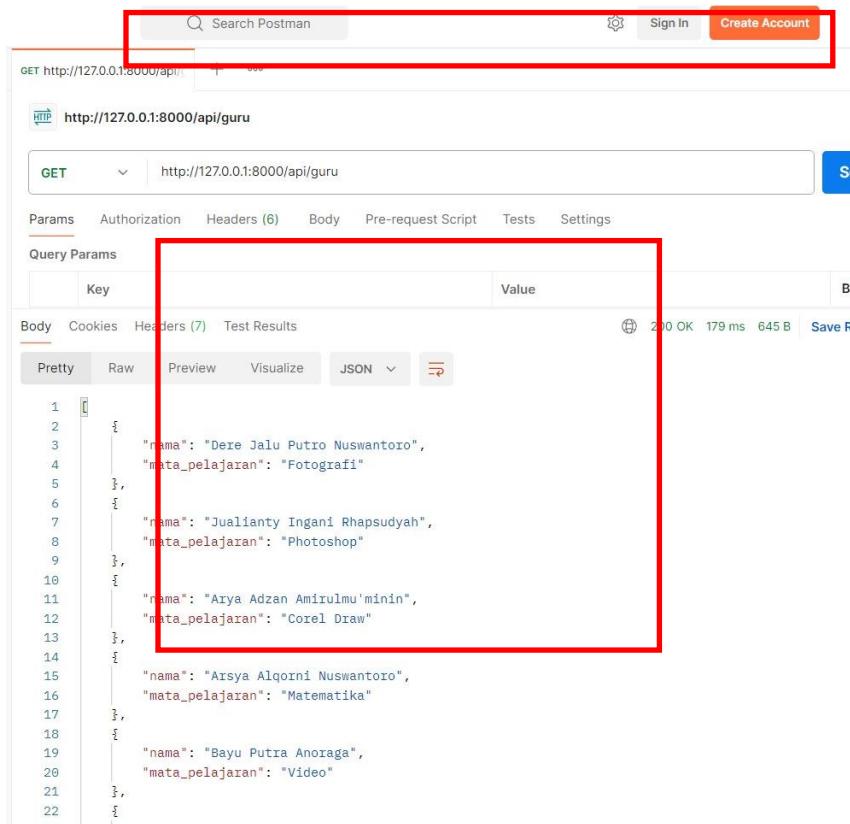
PARSING DATA MENGGUNAKAN API

Aplikasi web modern biasanya menggunakan API (Application Programming Interface) untuk menghubungkan berbagai sistem dan layanan. Dalam materi ini akan membahas bagaimana menggunakan [ReactJS](#) dengan API untuk mengambil data dari server dan menampilkannya dalam komponen.

API adalah sebuah protokol yang digunakan untuk menghubungkan antara dua sistem yang berbeda dan memungkinkan sistem tersebut untuk berinteraksi satu sama lain. Pada aplikasi web, API digunakan untuk mengambil data dari server dan menampilkan data tersebut di aplikasi. Dengan menggunakan API, aplikasi web dapat berinteraksi dengan layanan lainnya seperti mengirim email, melakukan pembayaran online, dan lain sebagainya. Dengan begitu, API memungkinkan aplikasi web untuk mengakses berbagai sumber daya yang lebih luas daripada hanya mengandalkan data yang tersimpan di dalam aplikasi itu sendiri.

1. Mengambil data dari API dengan menggunakan fetch()

Sebelum mengambil data menggunakan fungsi **fetch()**, pengecekan Restful Api dilakukan apakah sudah berjalan dengan baik dan menampilkan data, jika sudah akan muncul tampilan seperti gambar di bawah ini. Pengecekan menggunakan aplikasi Postman



The screenshot shows the Postman interface with a red box highlighting the response body. The response body is a JSON array of objects:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
{
  "nama": "Dere Jalu Putro Nuswantoro",
  "mata_pelajaran": "Fotografi"
},
{
  "nama": "Jualianty Ingani Rhapsudyah",
  "mata_pelajaran": "Photoshop"
},
{
  "nama": "Aiya Adzan Amirulmu'minin",
  "mata_pelajaran": "Corel Draw"
},
{
  "nama": "Aisyah Alqorni Nuswantoro",
  "mata_pelajaran": "Matematika"
},
{
  "nama": "Bayu Putra Anozaga",
  "mata_pelajaran": "Video"
}
```

Fetch adalah sebuah fungsi baru pada JavaScript yang digunakan untuk mengambil data dari server dengan mengirimkan permintaan HTTP. Fetch mengembalikan sebuah respon, sehingga kita dapat menangani respon data yang dikembalikan oleh server dengan menggunakan `then()` dan `catch()`.

Berikut adalah contoh penggunaan `fetch()` untuk mengambil data dari sebuah API:

```
● ● ●  
1 componentDidMount = () =>{  
2     fetch("http://127.0.0.1:8000/api/guru")  
3         .then(response => response.json())  
4         .then(data => this.setState({data:data}))  
5     }  
}
```

2. Mengelola data pada State

Anda akan mengubah respons JSON menjadi objek JavaScript dengan menggunakan `.json()` yang selanjutnya akan di simpan data dari API di dalam state komponen menggunakan `setState` yang nantinya object **data** bisa dipanggil untuk dimanipulasi.

```
● ● ●  
1 constructor(props) {  
2     super(props);  
3     this.state = {  
4         data: []  
5     }  
6 }  
7  
8 componentDidMount = () =>{  
9     fetch("http://127.0.0.1:8000/api/guru")  
10        .then(response => response.json())  
11        .then(data => this.setState({data:data}))  
12    }  
}
```

Sekarang data sudah tersedia dalam state komponen, Anda dapat menggunakan data tersebut dalam rendering komponen React Anda. Dalam contoh ini, **data** di-mapping untuk menampilkan elemen-elemen JSX untuk setiap item dalam array data

```

1  import { Component } from 'react';
2
3  class TambahGuru extends Component{
4
5      constructor(props) {
6          super(props);
7          this.state = {
8              data: []
9          }
10     }
11
12     componentDidMount = () =>{
13         fetch("http://127.0.0.1:8000/api/tambah")
14             .then(response => response.json())
15             .then(data => this.setState({data:data}))
16     }
17
18
19     render = () => {
20         return (
21             <div className='container'>
22                 <h1>Tambah Guru</h1>
23                 <form>
24                     <table className='table'>
25                         <tr>
26                             <td>Nama Guru</td>
27                             <td></td>
28                             <td><input type="text" name="nama" className="form-control"/></td>
29                         </tr>
30                         <tr>
31                             <td>Mata Pelajaran</td>
32                             <td></td>
33                             <td><input type="text" name="mata_pelajaran" className="form-control"/></td>
34                         </tr>
35                         <tr>
36                             <td>Kelas</td>
37                             <td></td>
38                             <td>
39                                 <select name="kelas" className="form-control">
40                                     {this.state.data.map((kelas) =>
41                                         <option value="">(kelas.nama)</option>
42                                     )}
43                                 </select>
44                             </td>
45                         </tr>
46                         <tr>
47                             <td></td>
48                             <td></td>
49                             <td><button type="submit" className="btn btn-success">Simpan</button></td>
50                         </tr>
51                     </table>
52                 </form>
53             </div>
54         )
55     }
56 }
57
58 export default TambahGuru;

```

Adapun tampilan halaman dari penggunaan API yang telah dipraktekkan diatas adalah sebagai berikut

No	Guru	Mata pelajaran	Action
1	Joni	Matematika	<button>Edit</button> <button>Hapus</button>
2	dandi	IPA	<button>Edit</button> <button>Hapus</button>

Apabila tabel dari data yang berada di backend bisa ditampilkan maka API berfungsi dengan baik dan front end nya berfungsi sesuai dengan yang diharapkan.

PENULIS



Ilman Pradhana, S.Kom., M.Kom adalah salah satu guru yang mengajar di SMKN 3 Palu, Sulawesi Tengah. Sebagai Guru yang mengajar pada Program keahlian Pengembangan Perangkat Lunak dan GIM ini merupakan alumnus dari STMIK Adhiguna Palu tahun 2014 dan juga alumnus Universitas Islam Indonesia tahun 2019. Bergabung dalam kegiatan Upskilling dan Reskilling Angkatan IV Konsentrasi Rekayasa Perangkat Lunak dengan teman Teknik Design Laman (Web Tehcnologies) yang diselenggarakan oleh BBPPMPV BOE Malang pada tanggal 02 - 31 Mei 2024.



Kadek Unggah Adi Nope, S.Pd adalah guru Pengembangan Perangkat Lunak dan Gim (PPLG) dari SMK Negeri 1 Denpasar, Kota Denpasar Provinsi Bali. Merupakan alumnus Universitas Pendidikan Ganesha, Jurusan Pendidikan Teknik Informatika tahun 2020. Bergabung dalam kegiatan Upskilling dan Reskilling Angkatan IV Konsentrasi Rekayasa Perangkat Lunak dengan teman Teknik Design Laman (Web Tehcnologies) yang diselenggarakan oleh BBPPMPV BOE Malang pada tanggal 02 - 31 Mei 2024



Yanuar Setyoningsih, S.Pd. adalah guru Rekayasa Perangkat Lunak dari SMK PGRI 3 Malang. Lahir di Bangkalan pada 26 Januari 1993. Merupakan alumni dari Universitas Negeri Malang Program Studi S1 Pendidikan Teknik Informatika dan selalu bersemangat untuk meningkatkan kemampuan diri demi lebih memaksimalkan kemampuan mengajar dengan mengikuti kegiatan Upskilling dan Reskilling Pelatihan Guru Kejuruan Teknik Desain Laman (Web Technologies) angkatan 4 pada 02 Mei 2024 hingga 31 Mei 2024



Mukhamad Syafiullah, S.Pd.,Gr adalah guru di SMKN Tambakboyo Kab. Tuban, Jawa Timur. Merupakan Alumnus dari Universitas Negeri Malang Pada Tahun 2016. Serta Memperoleh gelar sebagai guru profesional pada tahun 2023 di Universitas Negeri Malang. Bergabung dalam kegiatan Upskilling dan Reskilling Angkatan IV Konsentrasi Rekayasa Perangkat Lunak dengan teman Teknik Design Laman (Web Tehcnologies) yang diselenggarakan oleh BBPPMPV BOE Malang pada tanggal 02 - 31 Mei 2024.



Yudi Febriandy, S.Kom adalah Guru SMKN 2 Mataram Provinsi Nusa Tenggara Barat. Merupakan Alumnus Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Bumigora Mataram Tahun 2015. Bergabung dalam kegiatan Upskilling dan Reskilling Angkatan IV Konsentrasi Rekayasa Perangkat Lunak dengan teman Teknik Design Laman (Web Tehcnologies) yang diselenggarakan oleh BBPPMPV BOE Malang pada tanggal 02 - 31 Mei 2024. Dengan harapan menambah semangat baru bagi guru-guru Teknik Komputer & Informatika se-Indonesia serta memantapkan slogan " Coding Harga Mati"



Mohammad Zakaria Iba, S.Kom adalah guru dari SMKN 1 Pariwisata Jayapura, Papua. Merupakan alumnus Universitas Sains dan Teknologi Jayapura Prodi Teknik Informatika 2019. Bergabung dalam kegiatan Upskilling dan Reskilling Angkatan IV Konsentrasi Rekayasa Perangkat Lunak dengan teman Teknik Design Laman (Web Tehcnologies) yang diselenggarakan oleh BBPPMPV BOE Malang pada tanggal 02 - 31 Mei 2024.



Feni Iriany, S.Kom adalah Guru Dari SMKN 3 Gorontalo Kota Gorontalo Merupakan Alumni Dari ICHSAN GORONTALO Teknik Informatika Bergabung dalam kegiatan Upskilling dan Reskilling Angkatan IV Konsentrasi Rekayasa Perangkat Lunak dengan teman Teknik Design Laman (Web Tehcnologies) yang diselenggarakan oleh BBPPMPV BOE Malang pada tanggal 02 - 31 Mei 2024.



Andy Rafi, S.Kom., M.Kom. adalah Guru di SMKS Mutiara Ilmu Makassar, Sulawesi Selatan. Merupakan alumnus STMIK Dipanegara Makassar 2001 dan STMIK Handayani Makassar 2017. Bergabung dalam kegiatan Upskilling dan Reskilling Angkatan IV Konsentrasi Rekayasa Perangkat Lunak dengan teman Teknik Design Laman (Web Tehcnologies) yang diselenggarakan oleh BBPPMPV BOE Malang pada tanggal 02 - 31 Mei 2024.



Dere Jalu Putro Nuswantoro, S.ST adalah guru dari SMK Negeri 3 Teknologi dan Rekayasa Merauke, Papua Selatan. Merupakan alumnus Politeknik Elektronika Negeri Surabaya, Bidang Studio Teknik Informatika Komputer, Tahun 2009. Bergabung dalam kegiatan Upskilling dan Reskilling Angkatan IV Konsentrasi Rekayasa Perangkat Lunak dengan teman Teknik Design Laman (Web Tehcnologies) yang diselenggarakan oleh BBPPMPV BOE Malang pada tanggal 02 - 31 Mei 2024.



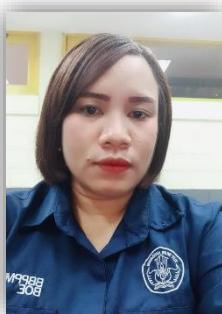
Aspert Mohede, S.Pd adalah guru di SMK NEGERI 1 AIRMADIDI Kabupaten Minahasa Utara, Provinsi Sulawesi Utara. Merupakan alumnus Universitas Negeri Manado 2013. Dengan semangat yang tak kenal lelah untuk terus berkembang dan memperluas wawasan dalam pemrograman yang berkembang pesat. Bergabung dalam kegiatan Upskilling dan Reskilling Angkatan IV Konsentrasi Rekayasa Perangkat Lunak dengan teman Teknik Design Laman (Web Tehcnologies) yang diselenggarakan oleh BBPPMPV BOE Malang pada tanggal 02 - 31 Mei 2024.



Andrias Koto, S.Kom, MM adalah guru di SMK Negeri 2 Kota Ternate. Merupakan alumnus BINUS tahun 1995 Jurusan Teknik Informatika dan Universitas Ibnu Khaldun Bogor Jurusan Manajemen Sistem Informasi tahun 2012. Bergabung dalam kegiatan Upskilling dan Reskilling Angkatan IV Konsentrasi Rekayasa Perangkat Lunak dengan teman Teknik Design Laman (Web Tehcnologies) yang diselenggarakan oleh BBPPMPV BOE Malang pada tanggal 02 - 31 Mei 2024



Agus, S.Pd adalah guru dari SMKN Labuang, Sulawesi Barat. Merupakan alumnus dari Universitas Negeri Makassar Prodi Pendidikan Teknik Elektronika. Bergabung dalam kegiatan Upskilling dan Reskilling Angkatan IV Konsentrasi Rekayasa Perangkat Lunak dengan teman Teknik Design Laman (Web Tehcnologies) yang diselenggarakan oleh BBPPMPV BOE Malang pada tanggal 02 - 31 Mei 2024.



Sanry Merciana Nisikoen, S.Kom adalah Guru dari SMKN 6 Kupang, Prov Nusa Tenggara Timur. Merupakan alumnus dari STIKOM UYELINDO Kupang Prodi Sistem Informasi tahun 2010. Bergabung dalam kegiatan Upskilling dan Reskilling Angkatan IV Konsentrasi Rekayasa Perangkat Lunak dengan teman Teknik Design Laman (Web Tehcnologies) yang diselenggarakan oleh BBPPMPV BOE Malang pada tanggal 02 - 31 Mei 2024.



Christonus Joens Luturmas, S.Pd.Kom adalah guru di SMK NEGERI 6 AMBON kota Ambon, Provinsi Maluku. Merupakan alumnus Universitas Kristen Satya Wacana 2020. Bergabung dalam kegiatan Upskilling dan Reskilling Angkatan IV Konsentrasi Rekayasa Perangkat Lunak dengan teman Teknik Design Laman (Web Tehcnologies) yang diselenggarakan oleh BBPPMPV BOE Malang pada tanggal 02 - 31 Mei 2024. Motto : "Orang yang mendidik anak-anak itu lebih dihormati daripada orang tua, jika orang tua hanya memberikan nafkah hidup, maka Guru memberikan seni kehidupan dengan baik."



Agustinus Karaeng Kanaka', S.Kom adalah guru dari SMKN 2 Kolaka Kabupaten Kolaka, Provinsi Sulawesi Tenggara. Merupakan alumnus Universitas Klabat Manado 2010 Fakultas Ilmu Komputer Prodi Sistem Informasi. Merupakan alumnus Universitas Kristen Satya Wacana 2020. Bergabung dalam kegiatan Upskilling dan Reskilling Angkatan IV Konsentrasi Rekayasa Perangkat Lunak dengan teman Teknik Design Laman (Web Tehcnologies) yang diselenggarakan oleh BBPPMPV BOE Malang pada tanggal 02 - 31 Mei 2024.

.:: -- 000 -- ::.