



Mata Kuliah : Pemrograman Web Lanjut (PWL)
Program Studi : D4 – Teknik Informatika / D4 – Sistem Informasi Bisnis
Semester : 4 (empat)
Pertemuan ke- : 1 (satu)
Nama : Nizam El Mulky Assalam
Repository : <https://github.com/nzm-el/PemogramanWebLanjut>

JOBSHEET 01

PENGENALAN WEB FRAMEWORK LARAVEL

➤ Teori

1. Pengenalan Web Development Modern

Web development telah berkembang pesat dari halaman statis sederhana menjadi aplikasi web yang kompleks dan interaktif. Pada era modern ini, pengembangan web melibatkan berbagai teknologi, tools, dan metodologi yang terintegrasi untuk menciptakan aplikasi yang responsif, aman, dan scalable.

Evolusi Web Development

Era	Tahun	Deskripsi
1.0	1990 - 2000	<i>Static Web</i> <ul style="list-style-type: none">– Halaman HTML statis– Konten yang tidak berubah– Interaksi minimal dengan pengguna– Contoh: halaman profil perusahaan sederhana
2.0	2000 - 2010	<i>Dynamic Web</i> <ul style="list-style-type: none">– Konten yang dinamis dan database-driven– Interaksi pengguna meningkat (form, komentar)– Penggunaan JavaScript untuk interaktivitas– Contoh: blog, forum, e-commerce awal
3.0	2010 - now	<i>Modern Web</i> <ul style="list-style-type: none">– <i>Single Page Applications</i> (SPA)– <i>Real-time communication</i>– <i>Progressive Web Apps</i> (PWA)– <i>API-driven architecture</i>– <i>Cloud-based deployment</i>– Contoh: Gmail, Facebook, Netflix, Instagram

Karakteristik Web Modern



- a. *Responsiveness*: Tampilan menyesuaikan dengan berbagai ukuran layar (mobile, tablet, desktop)
- b. *Interactivity*: *User experience* yang *smooth* dan *real-time*
- c. *Performance*: *Loading* cepat dan optimasi *resource*
- d. *Security*: Enkripsi data, autentikasi, dan proteksi dari serangan
- e. *Scalability*: Mampu menangani *traffic* yang tinggi
- f. *Maintainability*: Kode yang terstruktur dan mudah dipelihara

- a. Client-Server Architecture
- b. Frontend vs Backend
- c. Apa itu Framework?

2. Mengenal Framework Laravel

- a. Sejarah dan filosofi Laravel
- b. Keunggulan Laravel
- c. Struktur folder Laravel
- d. Konsep MVC dalam Laravel

3. Development Environment

- a. Laragon (PHP 8.2 ke atas)
- b. MySQL/MariaDB
- c. phpMyAdmin
- d. Text Editor VS Code
- e. Composer 2
- f. Node.js & NPM
- g. GIT
- h. Modern Browser

➤ Praktikum



1. Instalasi Laragon
2. Instalasi phpMyAdmin
3. Instalasi Composer
4. Instalasi Node.js & NPM
5. Instalasi GIT
6. Instalasi Laravel 12
 - a. Setup Laravel
 - b. Konfigurasi Env (koneksi DB, APP_KEY, dll)
7. Publish Project Laravel ke Github
8. Update Kode Program dan Sync ke Repo Github

➤ Tugas

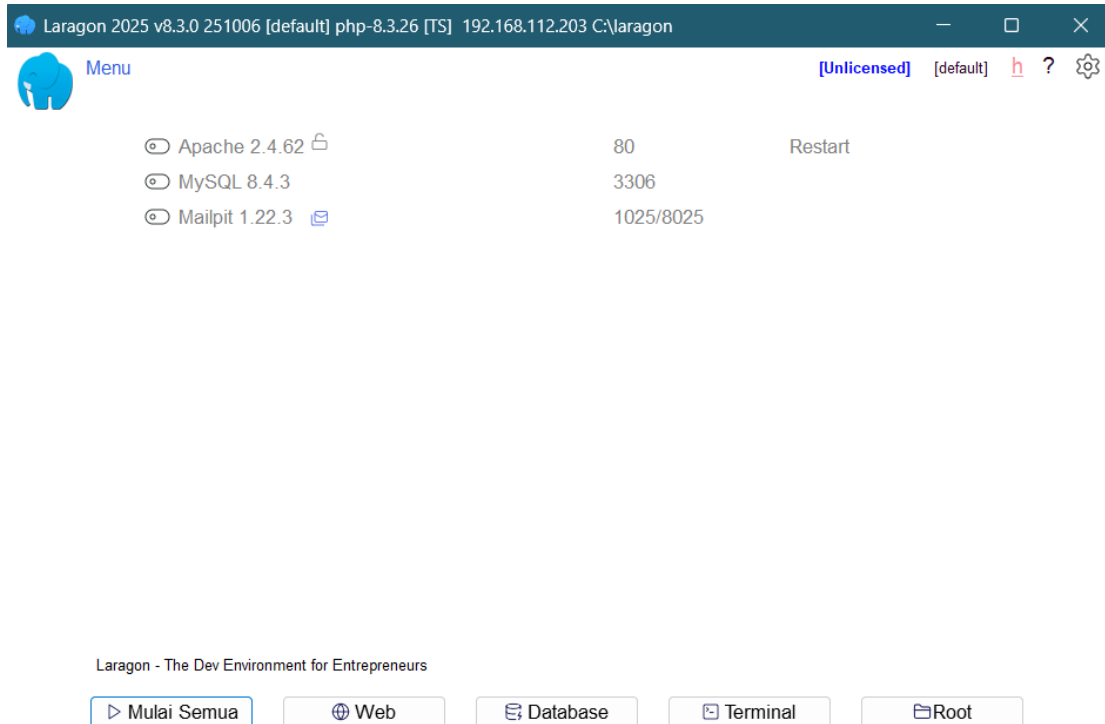
1. Buat dokumentasi langkah-langkah instalasi dengan menyertakan screenshot dan penjelasan tiap langkahnya.
2. Identifikasi folder-folder penting pada project Laravel (minimal 10 folder). Jelaskan fungsinya

➤ Jawaban

1. Instalasi Laragon

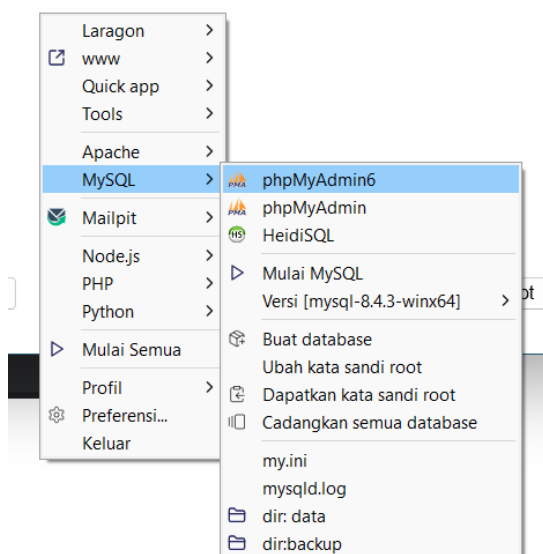
Laragon merupakan aplikasi yang dapat merubah komputer menjadi sebuah server maupun lokal atau dapat disebut sistem web stack untuk pengembangan web.

Langkah: Unduh Laragon Full version dari situs resminya. Jalankan installer (Next > Next > Install). Setelah selesai, buka aplikasi Laragon dan klik tombol "Start All". Pastikan Apache dan MySQL berjalan (berwarna hijau).



2. Instalasi phpMyAdmin

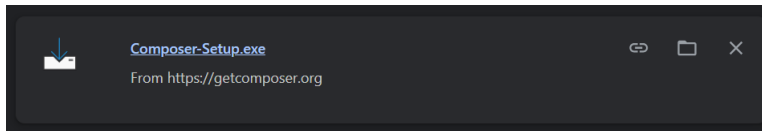
Langkah : Klik kanan pada tombol Database di Laragon atau buka menu MySQL > phpMyAdmin. Jika belum ada, Laragon akan menawarkan untuk mengunduhnya secara otomatis.



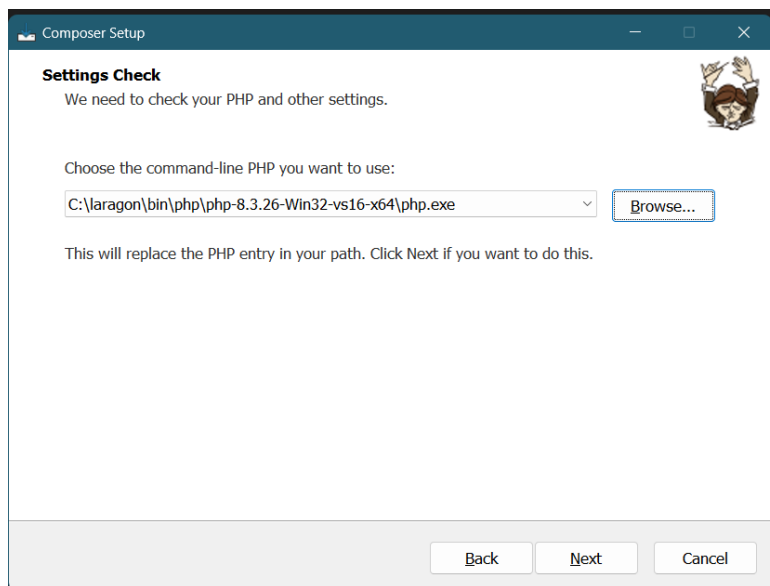


3. Instalasi Composer

1. Penginstallan di <https://getcomposer.org/>



2. Pemindahan file, jika pakai Laragon, biasanya ada di: C:\laragon\bin\php\php-x.x.x\php.exe.



3. Pengecekan compesor, apabila sudah terinstall di CMD

```
C:\laragon\www
λ composer

Composer version 2.9.5 2026-01-29 11:40:53

Usage:
  command [options] [arguments]

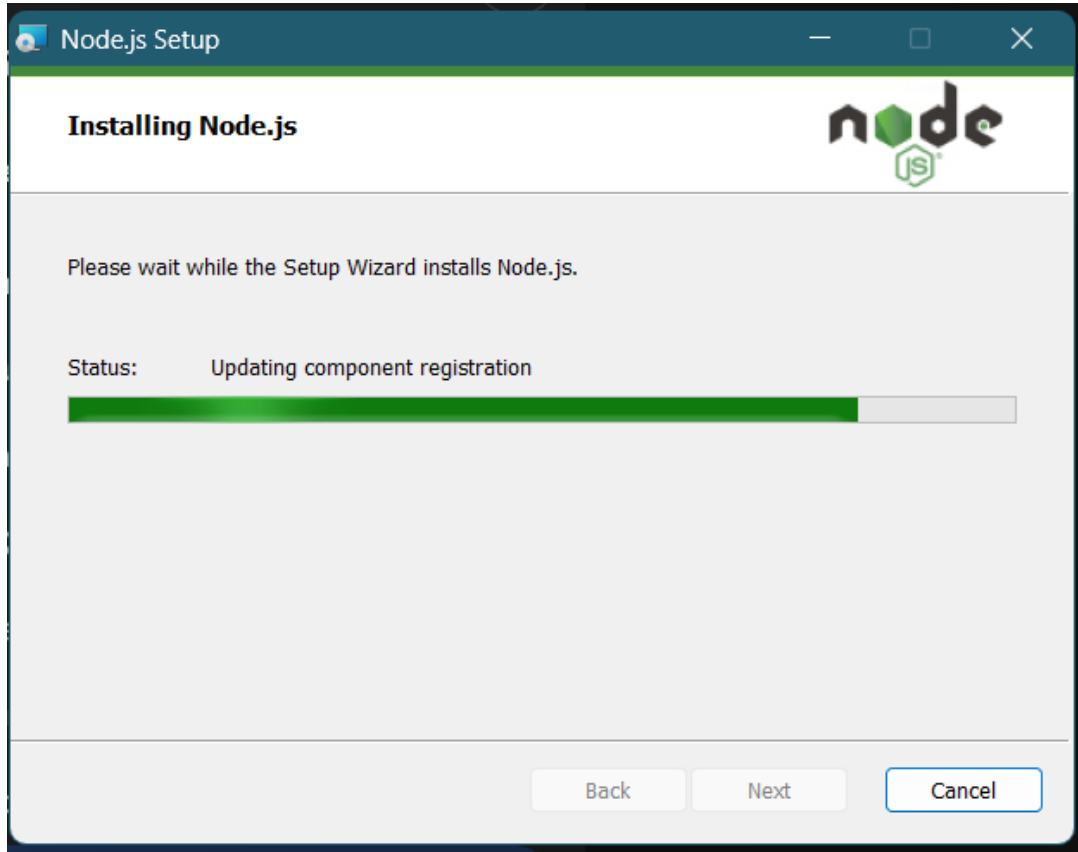
Options:
  -h, --help                Display help for the given command. When no command is given display help for the list command
  -q, --quiet               Do not output any message
  -V, --version             Display this application version
  --ansi|--no-ansi          Force (or disable --no-ansi)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

Available commands:
  about                Shows a short information about Composer
  archive              Creates an archive of this composer package
  audit                Checks for security vulnerability advisories for installed packages
  browse               [home] Opens the package's repository URL or homepage in your browser
  bump                Increases the lower limit of your composer.json requirements to the currently installed versions
  check-platform-reqs Check that platform requirements are satisfied
  clear-cache           [clearcache|cc] Clears composer's internal package cache
  completion           Dump the shell completion script
  config               Sets config options
  create-project        Creates new project from a package into given directory
  depends              [why] Shows which packages cause the given package to be installed
  diagnose             Diagnoses the system to identify common errors
  dump-autoload        [dumpautoload] Dumps the autoloader
  exec                 Executes a vendored binary/script
  fund                 Discover how to help fund the maintenance of your dependencies
  global               Allows running commands in the global composer dir ($COMPOSER_HOME)
  help                 Display help for a command
  init                 Creates a basic composer.json file in current directory
  install              [i] Installs the project dependencies from the composer.lock file if present, or falls back on the composer.json
  licenses             Shows information about licenses of dependencies
```



4. Instalasi Node.js & NPM

1. Penginstallan di <https://nodejs.org/en>



2. Pengecekan, apabila sudah terinstall di CMD

```
C:\Users\Lenovo>node -v
v24.13.0

C:\Users\Lenovo>npm -v
11.6.2
```

5. Instalasi GIT

1. Penginstallan di nodejs.org.
2. Pilih versi LTS (Long Term Support). Versi ini paling stabil





3. Pengecekan, apabila sudah terinstall di CMD

```
C:\Users\Lenovo>git --version  
git version 2.46.0.windows.1
```

6. Instalasi Laravel 12

1. Setup Laravel

```
- Installing phpunit/phpunit (11.5.52): Extracting archive  
59 package suggestions were added by new dependencies, use `composer suggest` to see details.  
Generating optimized autoload files  
> Illuminate\Foundation\ComposerScripts::postAutoloadDump  
> @php artisan package:discover --ansi  
  
INFO Discovering packages.  
  
laravel/pail ..... DONE  
laravel/sail ..... DONE  
laravel/tinker ..... DONE  
nesbot/carbon ..... DONE  
nunomaduro/collision ..... DONE  
nunomaduro/termwind ..... DONE  
  
81 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].  
  
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 ..... 22.18ms DONE  
  
INFO Running migrations.  
  
0001_01_01_000000_create_users_table ..... 45.40ms DONE  
0001_01_01_000001_create_cache_table ..... 13.79ms DONE  
0001_01_01_000002_create_jobs_table ..... 28.87ms DONE
```

2. Konfigurasi Env (koneksi DB, APP_KEY, dll)

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
  
PS C:\laragon\www\projek_pw1> php artisan key:generate  
  
INFO Application key set successfully.
```



```
DB_CONNECTION=mysql
# DB_HOST=127.0.0.1
# DB_PORT=3306
# DB_DATABASE=projek_pwl
# DB_USERNAME=root
# DB_PASSWORD=
```

```
PROBLEMS  OUTPUT  TERMINAL  ...
php + - [ ] [ ] ... [X]

PS C:\laragon\www\projek_pwl> php artisan key:generate

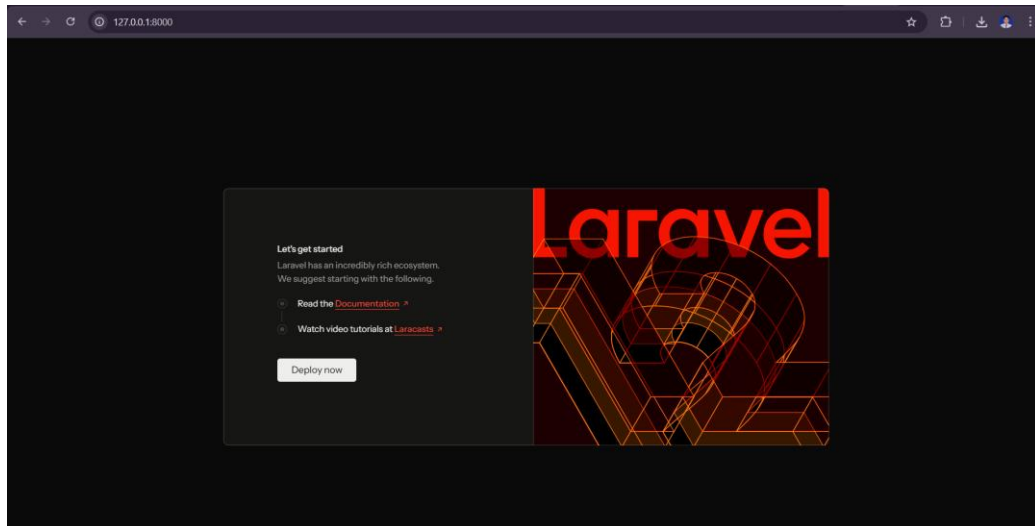
[INFO] Application key set successfully.

PS C:\laragon\www\projek_pwl> php artisan serve

[WARN] Unable to respect the `PHP_CLI_SERVER_WORKERS` environment variable
without the `--no-reload` flag. Only creating a single server.

[INFO] Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server
```





7. Publish Project Laravel ke Github

```
Lenovo@LAPTOP-1TKQ0FS5 MINGW64 ~/Documents/SMT 4/PWL/PemogramanWebLanjut (main)
$ git push -u origin main
Enumerating objects: 76, done.
Counting objects: 100% (76/76), done.
Delta compression using up to 12 threads
Compressing objects: 100% (59/59), done.
Writing objects: 100% (76/76), 71.96 KiB | 1.12 MiB/s, done.
Total 76 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/nzm-el/PemogramanWebLanjut
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

8. Identifikasi folder-folder penting pada project Laravel (minimal 10 folder) dan jelaskan fungsinya.

No	Nama Folder	Penjelasan Fungsi
1	app/	Ini adalah folder inti aplikasi. Hampir semua logika bisnis aplikasi (seperti Model, Controller, Middleware) disimpan di sini.
2	app/Http/Controllers/	Sub-folder tempat menyimpan file <i>Controller</i> . Controller berfungsi mengatur alur data antara Model dan View (menerima request user dan mengembalikan response).
3	app/Models/	Tempat menyimpan file <i>Model</i> Eloquent. Model merepresentasikan tabel di database dan digunakan untuk berinteraksi dengan data (CRUD).
4	bootstrap/	Berisi file app.php yang menginisialisasi framework. Folder ini juga menyimpan folder cache untuk optimasi performa aplikasi.



No	Nama Folder	Penjelasan Fungsi
5	config/	Berisi seluruh file konfigurasi aplikasi, seperti konfigurasi database (database.php), aplikasi (app.php), mail, dan services lainnya.
6	database/	Berisi file-file yang berhubungan dengan database, seperti <i>Migrations</i> (struktur tabel), <i>Seeders</i> (data dummy), dan <i>Factories</i> .
7	public/	Folder ini adalah pintu masuk (<i>entry point</i>) bagi request web. Berisi file index.php dan aset yang bisa diakses publik seperti gambar, CSS hasil kompilasi, dan JavaScript.
8	resources/	Berisi aset yang belum dikompilasi (raw), seperti file view (Blade templates), file CSS mentah (SASS/LESS), dan file JavaScript mentah.
9	routes/	Tempat mendefinisikan seluruh URL/rute aplikasi. File terpenting di sini adalah web.php (untuk rute web browser) dan api.php (untuk rute API).
10	storage/	Folder tempat Laravel menyimpan file yang dihasilkan aplikasi, seperti log error (logs/), file cache, sesi user, dan file yang diunggah pengguna (<i>user uploads</i>).
11	tests/	Berisi file-file untuk pengujian otomatis (<i>automated testing</i>), baik <i>Feature</i>



No	Nama Folder	Penjelasan Fungsi
		<i>Test</i> maupun <i>Unit Test</i> untuk memastikan aplikasi berjalan sesuai harapan.
12	vendor/	Berisi <i>library</i> pihak ketiga (dependencies) yang diunduh oleh Composer . Folder ini tidak boleh diedit secara manual.