

LAMPIRAN

🚩 Data Jumlah Penduduk Muslim Kecamatan Magersari, Kota Mojokerto Tahun 2023

 **PEMERINTAH KOTA MOJOKERTO**
DINAS KEPENDUDUKAN DAN PENCATATAN SIPIL
Jalan Gajah Mada No. 100 Kecamatan Magersari, Kota Mojokerto, 61341
Telepon/Faksimile (0321) 395820
Laman *dispenduk.mojokertokota.go.id* Pos-el *dispendukcapil@mojokertokota.go.id*

Mojokerto, 09 Juli 2024

Nomor : 400.12.41/497/417.509.3/2024
Sifat : Segera
Lampiran : -
Hal : Permintaan Data

Yth. Dekan Fakultas Teknik dan Perencanaan
Institut Teknologi Nasional Malang
di -
MALANG

Menindaklanjuti surat Saudara nomor ITN.05.111 / 25.SKRIPSI / VI.FTSP / 2024 tanggal 29 Mei 2024 perihal Permohonan Data, maka bersama ini kami sampaikan Data Jumlah Penduduk Muslim Kecamatan Magersari Tahun 2023 sebagai berikut :

NO	KELURAHAN	LAKI-LAKI	PEREMPUAN	JUMLAH
1	GUNUNG GEDANGAN	3,919	3,779	7,698
2	MAGERSARI	2,437	2,539	4,976
3	GEDONGAN	875	865	1,740
4	BALONGSARI	3,743	3,745	7,488
5	KEDUNDUNG	7,944	7,687	15,631
6	WATES	8,864	9,145	18,009
TOTAL		27,782	27,760	55,542

Demikian untuk menjadi maklum atas perhatian dan kerjasamanya kami sampaikan terimakasih.

Pr. KEPALA DINAS KEPENDUDUKAN
DAN PENCATATAN SIPIL
KOTA MOJOKERTO

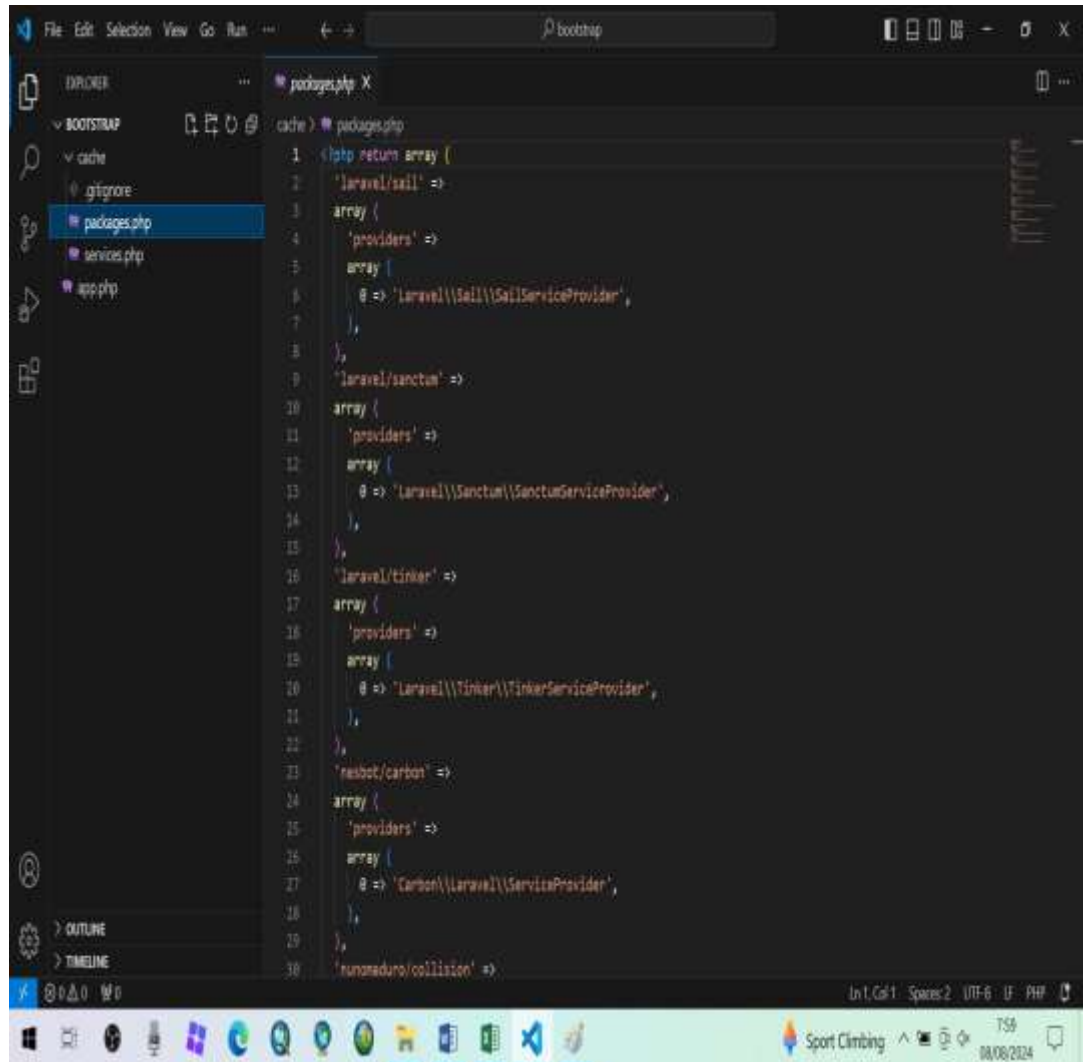

ANANG FAHRUROJI, S.Sos, M.Si
Pembina Utama Muda (IV/c)
NIP 196708071992031005

✚ **Data Dokumentasi Survey Lapangan**

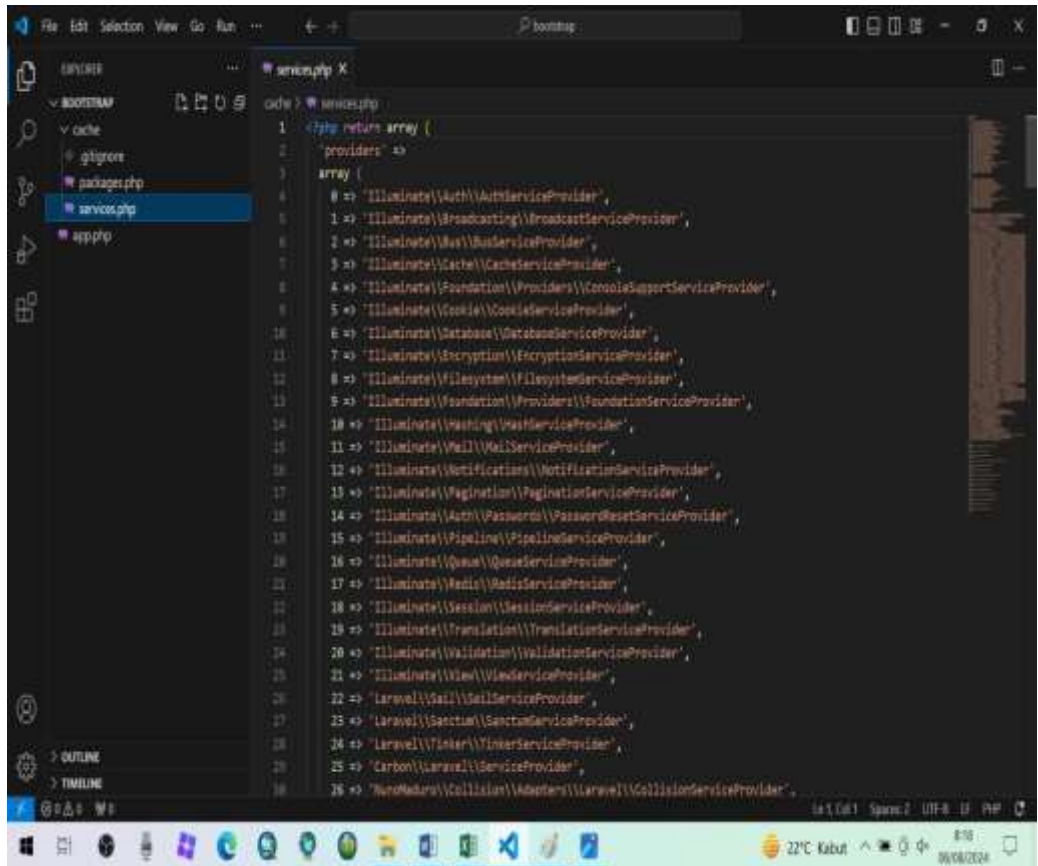


🚩 Data Source Code

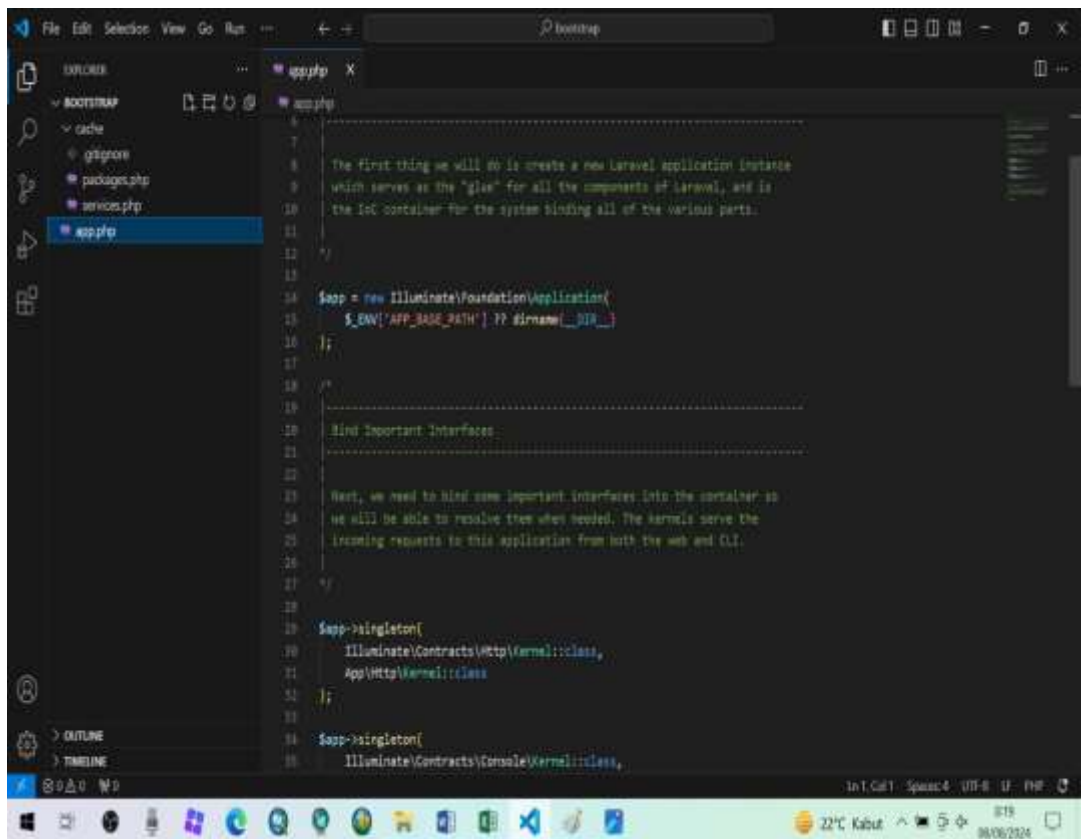
➤ *Bootstrap*



```
1 <?php return array (
2     'laravel/sail' =>
3     array (
4         'providers' =>
5         array (
6             0 => 'Laravel\\Sail\\SailServiceProvider',
7         ),
8     ),
9     'laravel/sanctum' =>
10    array (
11        'providers' =>
12        array (
13            0 => 'Laravel\\Sanctum\\SanctumServiceProvider',
14        ),
15    ),
16    'laravel/tinker' =>
17    array (
18        'providers' =>
19        array (
20            0 => 'Laravel\\Tinker\\TinkerServiceProvider',
21        ),
22    ),
23    'nesbot/carbon' =>
24    array (
25        'providers' =>
26        array (
27            0 => 'Carbon\\Laravel\\ServiceProvider',
28        ),
29    ),
30    'nunomaduro/collision' =>
```



```
1 <?php return array (
2     'providers' =>
3     array (
4         0 => 'Illuminate\Auth\AuthServiceProvider',
5         1 => 'Illuminate\Broadcasting\BroadcastServiceProvider',
6         2 => 'Illuminate\Bus\BusServiceProvider',
7         3 => 'Illuminate\Cache\CacheServiceProvider',
8         4 => 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
9         5 => 'Illuminate\Cookie\CookieServiceProvider',
10        6 => 'Illuminate\Database\DatabaseServiceProvider',
11        7 => 'Illuminate\Encryption\EncryptionServiceProvider',
12        8 => 'Illuminate\Filesystem\FilesystemServiceProvider',
13        9 => 'Illuminate\Foundation\Providers\FoundationServiceProvider',
14        10 => 'Illuminate\Hashing\HashServiceProvider',
15        11 => 'Illuminate\Mail\MailServiceProvider',
16        12 => 'Illuminate\Notifications\NotificationServiceProvider',
17        13 => 'Illuminate\Pagination\PaginationServiceProvider',
18        14 => 'Illuminate\Auth\Passwords\PasswordResetServiceProvider',
19        15 => 'Illuminate\Pipeline\PipelineServiceProvider',
20        16 => 'Illuminate\Queue\QueueServiceProvider',
21        17 => 'Illuminate\Redis\RedisServiceProvider',
22        18 => 'Illuminate\Session\SessionServiceProvider',
23        19 => 'Illuminate\Translation\TranslationServiceProvider',
24        20 => 'Illuminate\Validation\ValidationServiceProvider',
25        21 => 'Illuminate\View\ViewServiceProvider',
26        22 => 'Laravel\Mail\MailServiceProvider',
27        23 => 'Laravel\Sanctum\SanctumServiceProvider',
28        24 => 'Laravel\Tinker\TinkerServiceProvider',
29        25 => 'Carbon\Laravel\LaravelServiceProvider',
30        26 => 'NunoMaduro\Collision\Adapters\Laravel\CollisionServiceProvider',
```



```
1
2
3
4
5
6
7
8 The first thing we will do is create a new Laravel application instance
9 which serves as the "glue" for all the components of Laravel, and is
10 the IoC container for the system binding all of the various parts.
11
12
13
14 $app = new Illuminate\Foundation\Application(
15     $_ENV['APP_BASE_PATH'] ?? dirname(__DIR__)
16 );
17
18
19
20 Bind Important Interfaces
21
22
23 Next, we need to bind some important interfaces into the container so
24 we will be able to resolve them when needed. The kernels serve the
25 incoming requests to this application from both the web and CLI.
26
27
28
29 $app->singleton(
30     Illuminate\Contracts\Http\Kernel::class,
31     App\Http\Kernel::class
32 );
33
34 $app->singleton(
35     Illuminate\Contracts\Console\Kernel::class,
```

➤ Config

```
1 <?php
2
3 use Illuminate\Support\Facades\Facade;
4 use Illuminate\Support\ServiceProvider;
5
6 return [
7
8     /*
9     |> Application Name
10    |>
11    |> This value is the name of your application. This value is used when the
12    |> framework needs to place the application's name in a notification or
13    |> any other location as required by the application or its packages.
14    |>
15    |> @example 'name' => env('APP_NAME', 'Laravel');
16    |>
17    |> */
18     'name' => env('APP_NAME', 'Laravel'),
19
20     /*
21     |> Application Environment
22     |>
23     |> This value determines the "environment" your application is currently
24     |> running in. This may determine how you prefer to configure various
25     |> services the application utilizes. Set this in your ".env" file.
26     |>
27     |> @example 'env' => env('APP_ENV', 'production');
28     |>
29     |> */
30     'env' => env('APP_ENV', 'production'),
```

```
1 <?php
2
3 return [
4
5     /*
6     |> Authentication Defaults
7     |>
8     |> This option controls the default authentication "guard" and password
9     |> reset options for your application. You may change these defaults
10    |> as needed, but they're a perfect start for most applications.
11    |>
12    |> @example 'defaults' => [
13    |>     'guard' => 'web',
14    |>     'passwords' => 'users',
15    |> ];
16    |>
17    |> */
18     'defaults' => [
19         'guard' => 'web',
20         'passwords' => 'users',
21     ],
22
23     /*
24     |> Authentication Guards
25     |>
26     |> Next, you may define every authentication guard for your application.
27     |> Of course, a great default configuration has been defined for you
28     |> here which uses session storage and the Eloquent user provider.
29     |>
30     |> All authentication drivers have a user-provider. This defines how the
```

```
File Edit Selection View Go Run ... broadcasting.php  
DRIVER  
▼ CMS  
  app.php  
  auth.php  
  broadcasting.php  
  cache.php  
  cms.php  
  database.php  
  filesystems.php  
  hashing.php  
  logging.php  
  mail.php  
  queue.php  
  sanctum.php  
  services.php  
  session.php  
  view.php  
  OUTLINE  
  TIMELINE  
  In: Git | Spas4 | UTF-8 | LF | PHP  
  22°C Kabut | 8:25 | 08/08/2024
```

1 <?php
2
3 return [
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

Default Broadcaster

This option controls the default broadcaster that will be used by the framework when an event needs to be broadcast. You may set this to any of the connections defined in the "connections" array below.

Supported: "pusher", "ably", "redis", "log", "null"

'default' => env('BROADCAST_DRIVER', 'null'),

Broadcast Connections

Here you may define all of the broadcast connections that will be used to broadcast events to other systems or over websockets. Samples of each available type of connection are provided inside this array.

```
File Edit Selection View Go Run ... cache.php X  
DRIVER  
▼ CMS  
  app.php  
  auth.php  
  broadcasting.php  
  cache.php  
  cms.php  
  database.php  
  filesystems.php  
  hashing.php  
  logging.php  
  mail.php  
  queue.php  
  sanctum.php  
  services.php  
  session.php  
  view.php  
  OUTLINE  
  TIMELINE  
  In: Git | Spas4 | UTF-8 | LF | PHP  
  22°C Kabut | 8:25 | 08/08/2024
```

1 <?php
2
3 use Illuminate\Support\Str;
4
5 return [
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

Default Cache Store

This option controls the default cache connection that gets used while using this caching library. This connection is used when another is not explicitly specified when executing a given caching function.

'default' => env('CACHE_DRIVER', 'file'),

Cache Stores

Here you may define all of the cache "stores" for your application as well as their drivers. You may even define multiple stores for the same cache driver to group types of items stored in your caches.

Supported drivers: "apc", "array", "database", "file", "memcached", "redis", "dynamodb", "octane", "null"

```
1 <?php
2
3 use Illuminate\Support\Str;
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18 'default' => env('CACHE_DRIVER', 'file'),
19
20
21
22
23
24
25
26
27
28
29
30
```

Default Cache Store

This option controls the default cache connection that gets used while using this caching library. This connection is used when another is not explicitly specified when executing a given caching function.

Cache Stores

Here you may define all of the cache "stores" for your application as well as their drivers. You may even define multiple stores for the same cache driver to group types of items stored in your caches.

Supported drivers: 'apc', 'array', 'database', 'file', 'memcached', 'redis', 'dynamodb', 'octane', 'null'

➤ Database

```
1 <?php
2
3 namespace Database\Factories;
4
5 use Illuminate\Database\Eloquent\Factories\Factory;
6
7
8
9
10
11
12
13
14
15
16
17
18 class MajidmushallaFactory extends Factory
19 {
20     /**
21      * Define the model's default state.
22      *
23      * @return array<string, mixed>
24      */
25     public function definition(): array
26     {
27         return [
28             //
29         ];
30     }
31 }
32
```



```
File Edit Selection View Go Run -- database
... usefactory.php X
factories > usefactory.php
1 <?php
2
3 namespace Database\Factories;
4
5 use Illuminate\Database\Eloquent\Factories\Factory;
6 use Illuminate\Support\Facades\Hash;
7 use Illuminate\Support\Str;
8
9 /**
10  * @extends \Illuminate\Database\Eloquent\Factories\Factory<App\Models\User>
11  */
12 class UserFactory extends Factory
13 {
14     /**
15      * The current password being used by the factory.
16      */
17     protected static string $password;
18
19     /**
20      * Define the model's default state.
21      */
22     /**
23      * @return array/string, mixed
24      */
25     public function definition(): array
26     {
27         return [
28             'name' => fake()->name(),
29             'email' => fake()->unique()->safeEmail(),
30             'email_verified_at' => now(),
31             'password' => static::$password ??= Hash::make('password'),
32         ];
33     }
34 }
35
```

```
File Edit Selection View Go Run -- database
... 2014_10_12_00000_create_users_table.php X
migrations > 2014_10_12_00000_create_users_table.php
1 <?php
2 use Illuminate\Database\Migrations\Migration;
3 use Illuminate\Database\Schema\Blueprint;
4 use Illuminate\Support\Facades\Schema;
5
6 return new class extends Migration
7 {
8     /**
9      * Run the migrations.
10      */
11     public function up(): void
12     {
13         Schema::create('users', function (Blueprint $table) {
14             $table->id();
15             $table->string('name');
16             $table->string('email')->unique();
17             $table->timestamp('email_verified_at')->nullable();
18             $table->string('password');
19             $table->rememberToken();
20             $table->timestamps();
21         });
22     }
23
24     /**
25      * Reverse the migrations.
26      */
27     public function down(): void
28     {
29         Schema::dropIfExists('users');
30     }
31 }
32
```

The screenshot shows the Visual Studio Code editor with the file `DatabaseSeeder.php` open. The code is as follows:

```
1 <?php
2
3 namespace Database\Seeders;
4
5 // use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6 use Illuminate\Database\Seeder;
7
8 class DatabaseSeeder extends Seeder
9 {
10     /**
11      * Seed the application's database.
12      */
13     public function run(): void
14     {
15         // App\Models\User::factory(10)->create();
16
17         // App\Models\User::factory()->create([
18             // 'name' => 'Test User',
19             // 'email' => 'test@example.com',
20             // ]);
21     }
22 }
```

The screenshot shows the Visual Studio Code editor with the file `MasjidsMushollaSeeder.php` open. The code is as follows:

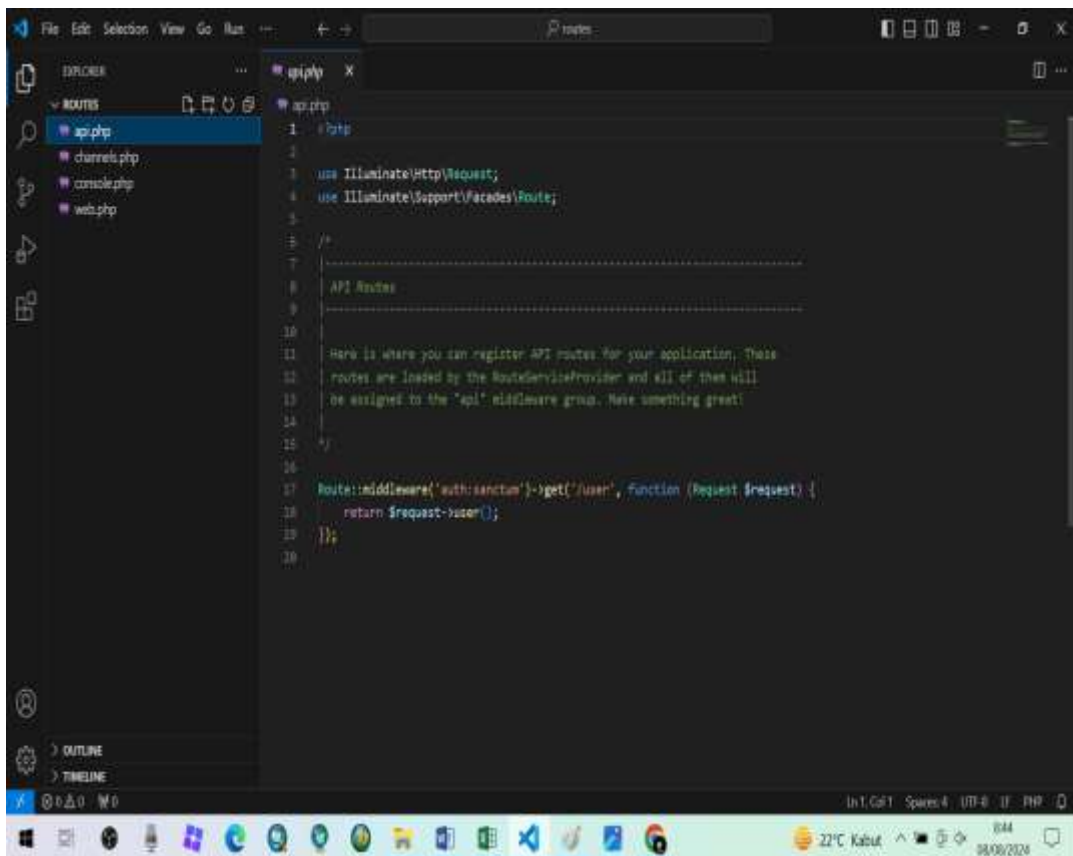
```
1 <?php
2
3 namespace Database\Seeders;
4
5 use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6 use Illuminate\Database\Seeder;
7
8 class MasjidsMushollaSeeder extends Seeder
9 {
10     /**
11      * Run the database seeds.
12      */
13     public function run(): void
14     {
15         //
16     }
17 }
```

➤ Geojson

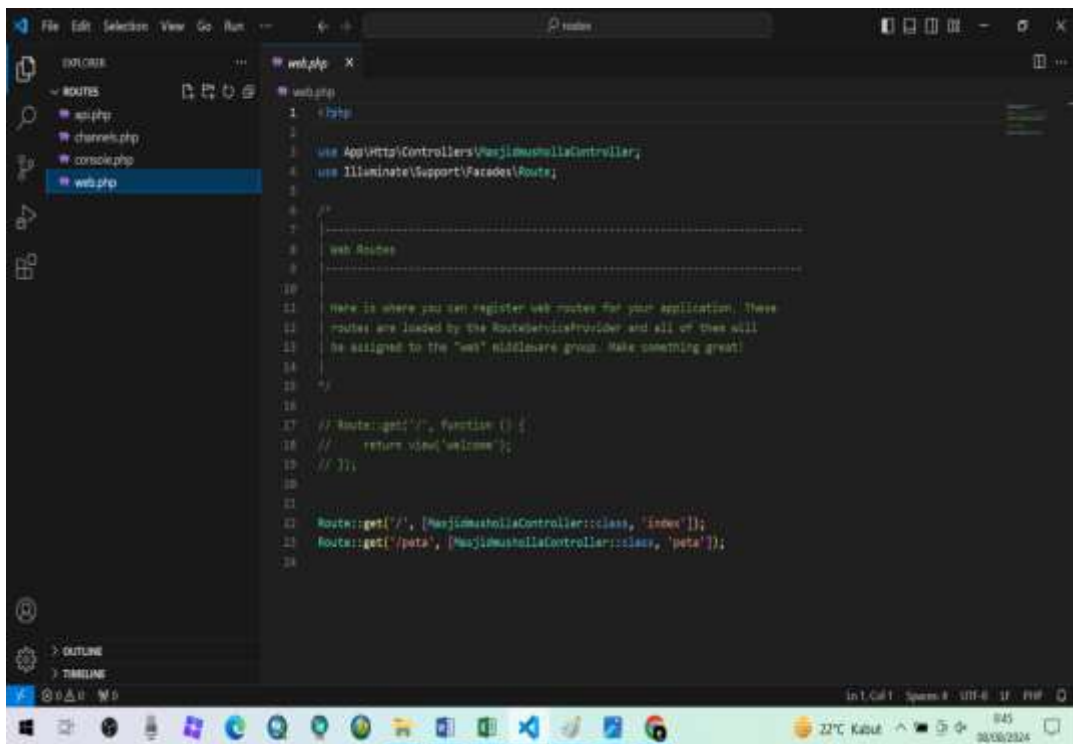
```
1 {
2   "type": "FeatureCollection",
3   "features": [
4     {
5       "type": "Feature",
6       "id": 0,
7       "geometry": {
8         "type": "Polygon",
9         "coordinates": [
10          [
11            [112.44826, -7.465189999999999453],
12            [112.44827000000000, -7.465189999999999477],
13            [112.44827000000000, -7.465199999999999414],
14            [112.44825000000000, -7.46519999999999944],
15            [112.44826, -7.465189999999999453],
16            [112.44828000000000, -7.465409999999999577],
17            [112.44828000000000, -7.465319999999999573],
18            [112.44828000000000, -7.465219999999999565],
19            [112.44828000000000, -7.465139999999999579],
20            [112.44828000000000, -7.464219999999999553],
21            [112.44828000000000, -7.463189999999999581],
22            [112.44829000000000, -7.463099999999999521],
23            [112.44829000000000, -7.462939999999999593],
24            [112.44829000000000, -7.462779999999999545],
25            [112.44796000000000, -7.462719999999999525],
26            [112.44737000000000, -7.462589999999999576],
27            [112.44723000000000, -7.462589999999999516],
28            [112.44701000000000, -7.462619999999999574],
29            [112.44627000000000, -7.462739999999999544],
30            [112.44602000000000, -7.462899999999999589]
```

```
1 {
2   "type": "FeatureCollection",
3   "features": [
4     {
5       "type": "Feature",
6       "id": 0,
7       "geometry": {
8         "type": "Polygon",
9         "coordinates": [
10          [
11            [112.44826, -7.465189999999999453],
12            [112.44827000000000, -7.465189999999999477],
13            [112.44827000000000, -7.465199999999999414],
14            [112.44825000000000, -7.46519999999999944],
15            [112.44826, -7.465189999999999453],
16            [112.44828000000000, -7.465409999999999577],
17            [112.44828000000000, -7.465319999999999573],
18            [112.44828000000000, -7.465219999999999565],
19            [112.44828000000000, -7.465139999999999579],
20            [112.44828000000000, -7.464219999999999553],
21            [112.44828000000000, -7.463189999999999581],
22            [112.44829000000000, -7.463099999999999521],
23            [112.44829000000000, -7.462939999999999593],
24            [112.44829000000000, -7.462779999999999545],
25            [112.44796000000000, -7.462719999999999525],
26            [112.44737000000000, -7.462589999999999576],
27            [112.44723000000000, -7.462589999999999516],
28            [112.44701000000000, -7.462619999999999574],
29            [112.44627000000000, -7.462739999999999544],
30            [112.44602000000000, -7.462899999999999589]
```

➤ Web Routes

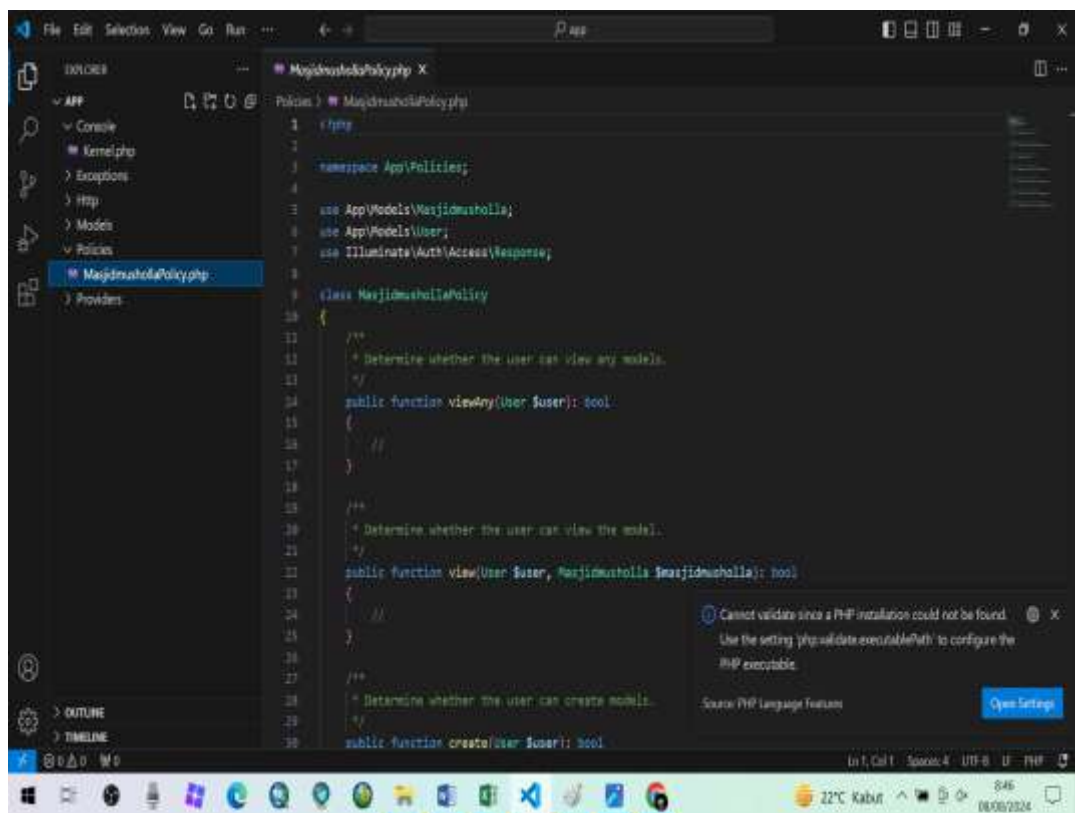


```
1 <?php
2
3 use Illuminate\Http\Request;
4 use Illuminate\Support\Facades\Route;
5
6
7
8
9
10
11
12 Here is where you can register API routes for your application. These
13 routes are loaded by the RouteServiceProvider and all of them will
14 be assigned to the "api" middleware group. Make something great!
15
16
17
18
19
20
21
22
23
24
25
26
27 Route::middleware('auth:sanctum')->get('/user', function (Request $request) {
28     return $request->user();
29 });
30
```

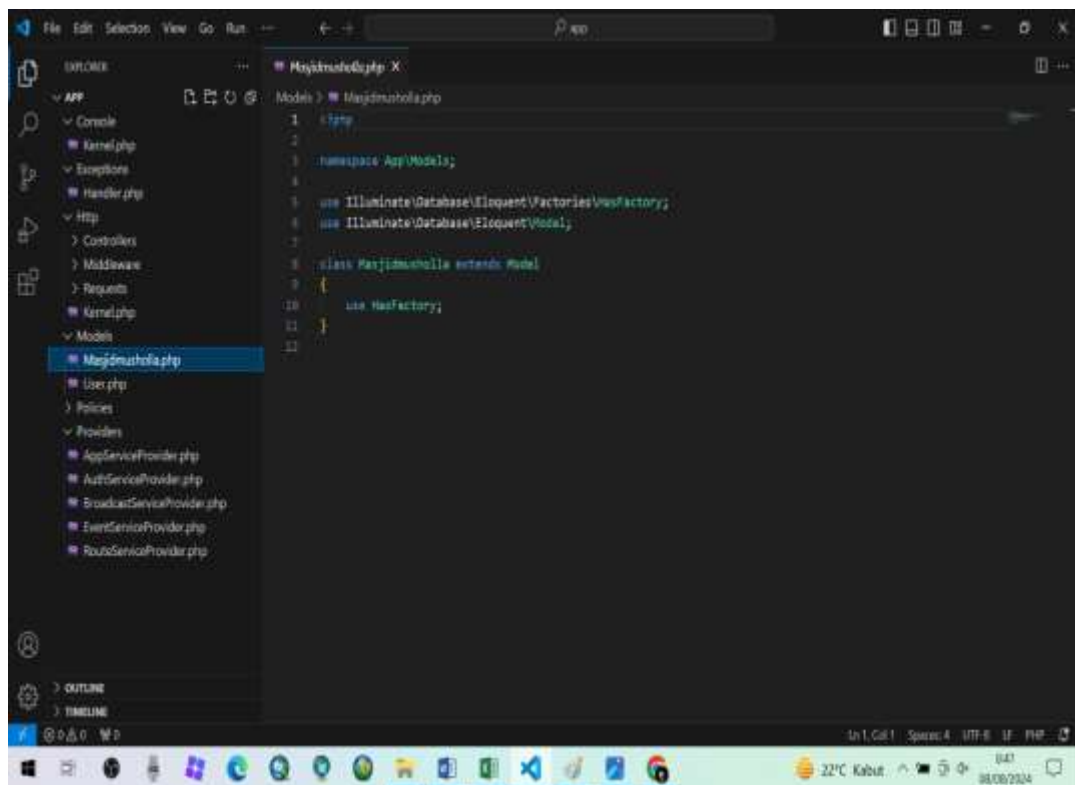


```
1 <?php
2
3 use App\Http\Controllers\MasjidMushollaController;
4 use Illuminate\Support\Facades\Route;
5
6
7
8
9
10
11
12 Here is where you can register web routes for your application. These
13 routes are loaded by the RouteServiceProvider and all of them will
14 be assigned to the "web" middleware group. Make something great!
15
16
17
18
19
20
21
22
23
24
25
26
27 // Route::get('/', function () {
28 //     return view('welcome');
29 // });
30
31
32
33 Route::get('/', [MasjidMushollaController::class, 'index']);
34 Route::get('/pets', [MasjidMushollaController::class, 'pets']);
35
```

➤ Masjid dan Musholla *PHP*



```
1 <?php
2
3 namespace App\Policies;
4
5 use App\Models\MasjidMusholla;
6 use App\Models\User;
7 use Illuminate\Auth\Access\Response;
8
9 class MasjidMushollaPolicy
10 {
11     /**
12      * Determine whether the user can view any models.
13      */
14     public function viewAny(User $user): bool
15     {
16         //
17     }
18
19     /**
20      * Determine whether the user can view the model.
21      */
22     public function view(User $user, MasjidMusholla $masjidMusholla): bool
23     {
24         //
25     }
26
27     /**
28      * Determine whether the user can create models.
29      */
30     public function create(User $user): bool
```



```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\UuidFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class MasjidMusholla extends Model
9 {
10     use HasFactory;
11 }
12
```