

1.1 Tujuan

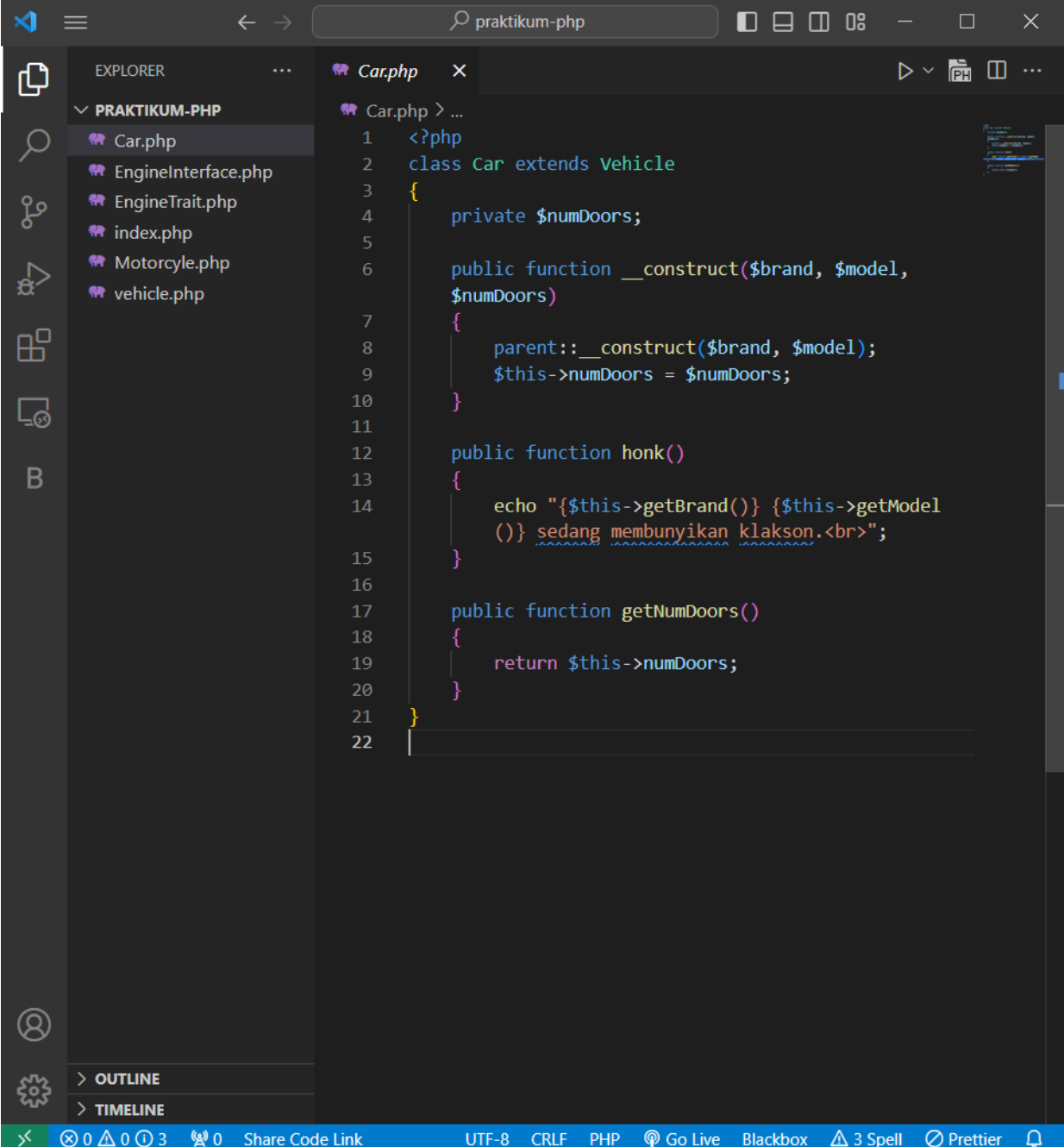
- Memahami konsep dasar pemrograman berorientasi objek (OOP) dalam PHP 8.
- Menerapkan konsep class, object, atribut, method, visibility, constructor, setter, dan getter dalam PHP.
- Memahami konsep inheritance, polymorphism, abstract, interface, dan traits dalam OOP.
- Menggunakan autoloading untuk memuat kelas dan trait secara otomatis dalam aplikasi PHP.
- Menggunakan namespace untuk mengorganisasi kelas dan trait dalam proyek.

1.2 Alat dan Bahan

1. Laptop
2. Web Server (XAMPP/Laragon atau lainnya)
3. Text Editor - Visual Studio Code
4. Browser - Chrome

1.3 Lampiran

car

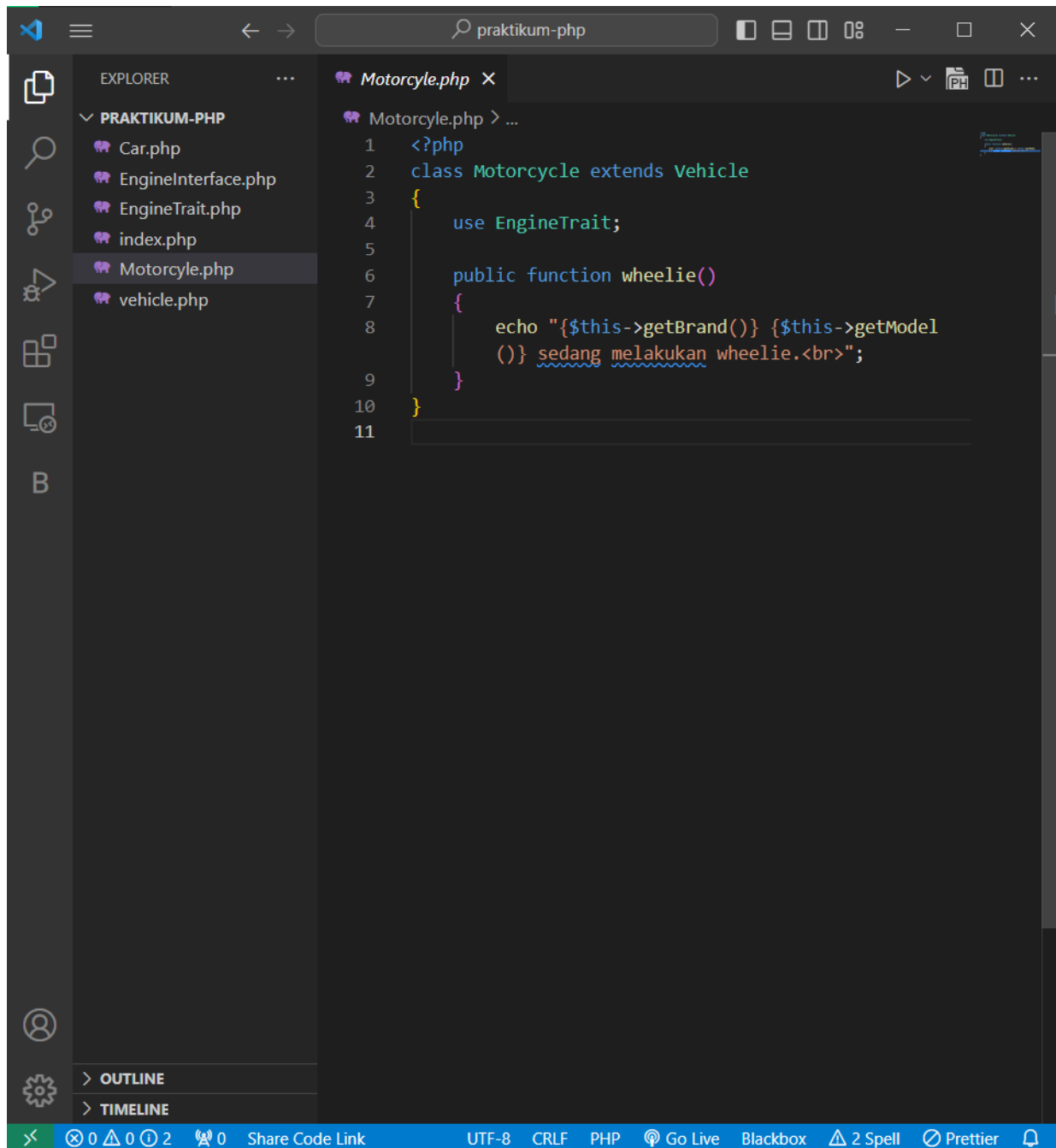


The screenshot shows a code editor with a dark theme. On the left, the 'EXPLORER' sidebar displays a file tree for a project named 'PRAKTIKUM-PHP'. The files listed are 'Car.php', 'EngineInterface.php', 'EngineTrait.php', 'index.php', 'Motorcycle.php', and 'vehicle.php'. The 'Car.php' file is selected and open in the main editor area. The code in 'Car.php' is as follows:

```
1 <?php
2 class Car extends Vehicle
3 {
4     private $numDoors;
5
6     public function __construct($brand, $model,
7         $numDoors)
8     {
9         parent::__construct($brand, $model);
10        $this->numDoors = $numDoors;
11    }
12
13    public function honk()
14    {
15        echo "{$this->getBrand()} {$this->getModel()}"
16            . " sedang membunyikan klakson.<br>";
17    }
18
19    public function getNumDoors()
20    {
21        return $this->numDoors;
22    }
23 }
```

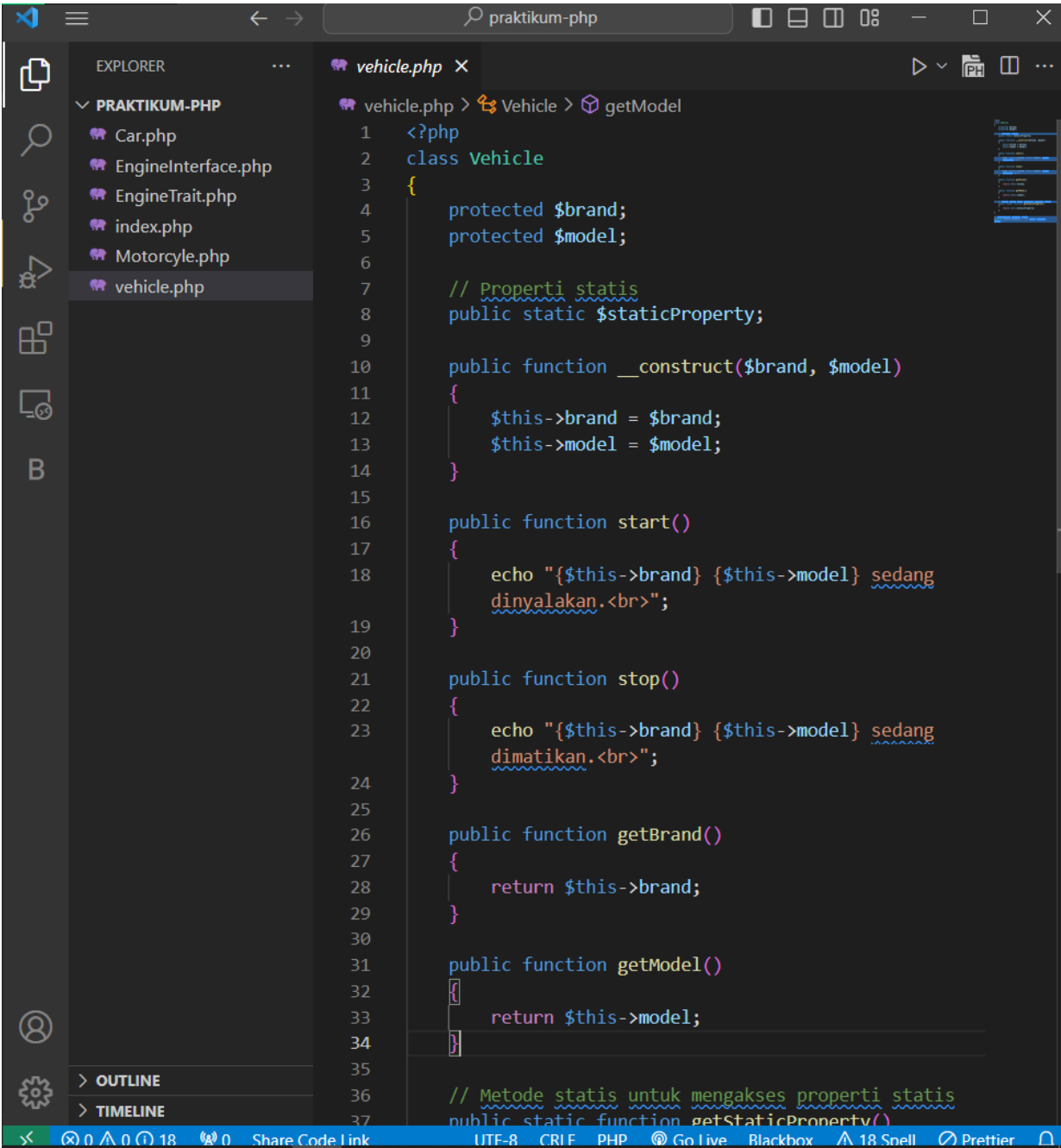
The status bar at the bottom of the editor shows various icons and settings, including 'UTF-8', 'CRLF', 'PHP', 'Go Live', 'Blackbox', '3 Spell', and 'Prettier'.

Motorcycle



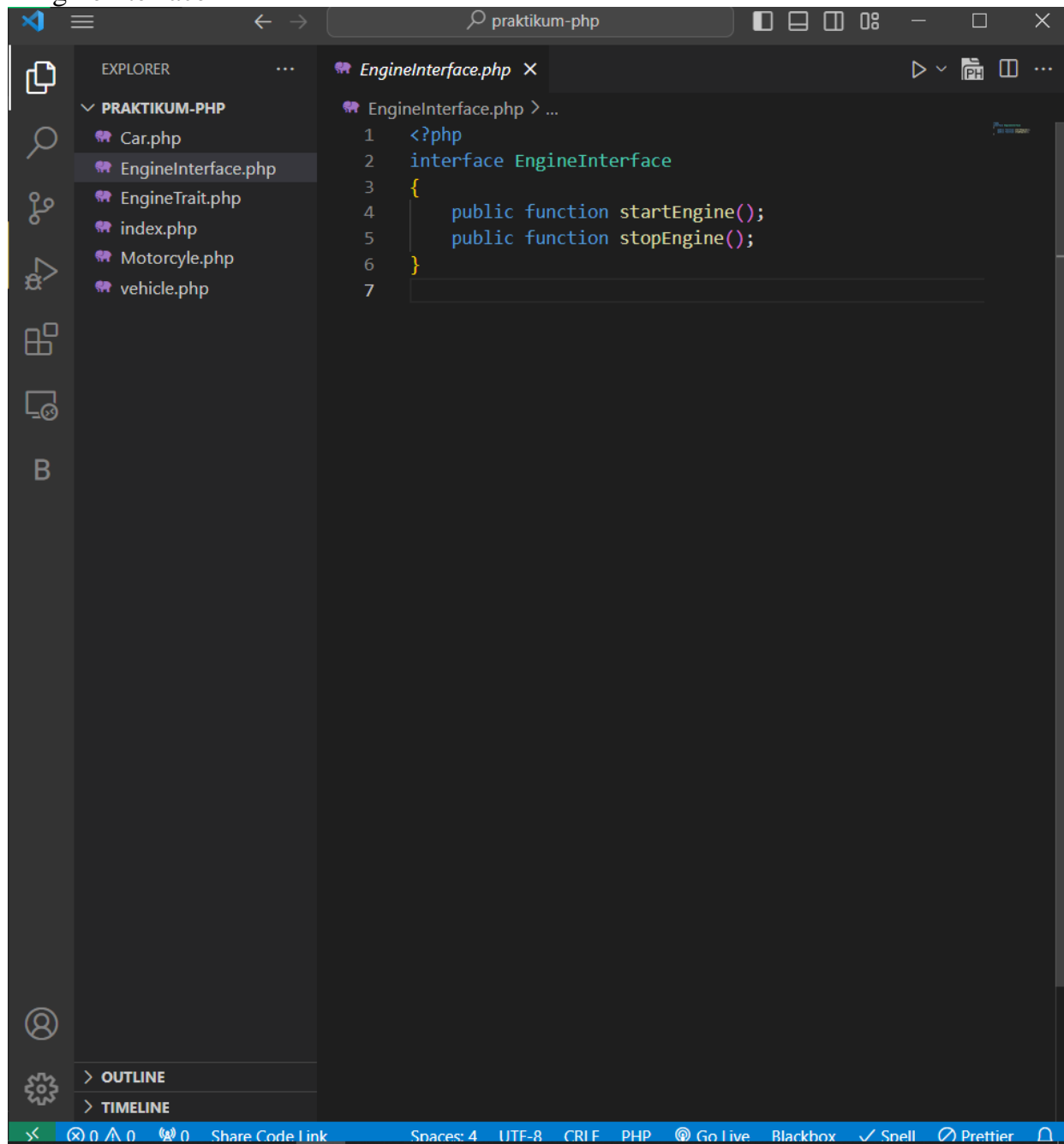
```
1 <?php
2 class Motorcycle extends Vehicle
3 {
4     use EngineTrait;
5
6     public function wheelie()
7     {
8         echo "{$this->getBrand()} {$this->getModel()} sedang melakukan wheelie.<br>";
9     }
10 }
11
```

vehicle



```
1 <?php
2 class Vehicle
3 {
4     protected $brand;
5     protected $model;
6
7     // Properti statis
8     public static $staticProperty;
9
10    public function __construct($brand, $model)
11    {
12        $this->brand = $brand;
13        $this->model = $model;
14    }
15
16    public function start()
17    {
18        echo "{$this->brand} {$this->model} sedang
19        dinyalakan.<br>";
20    }
21
22    public function stop()
23    {
24        echo "{$this->brand} {$this->model} sedang
25        dimatikan.<br>";
26    }
27
28    public function getBrand()
29    {
30        return $this->brand;
31    }
32
33    public function getModel()
34    {
35        return $this->model;
36    }
37
38    // Metode statis untuk mengakses properti statis
39    public static function getStaticProperty()
```

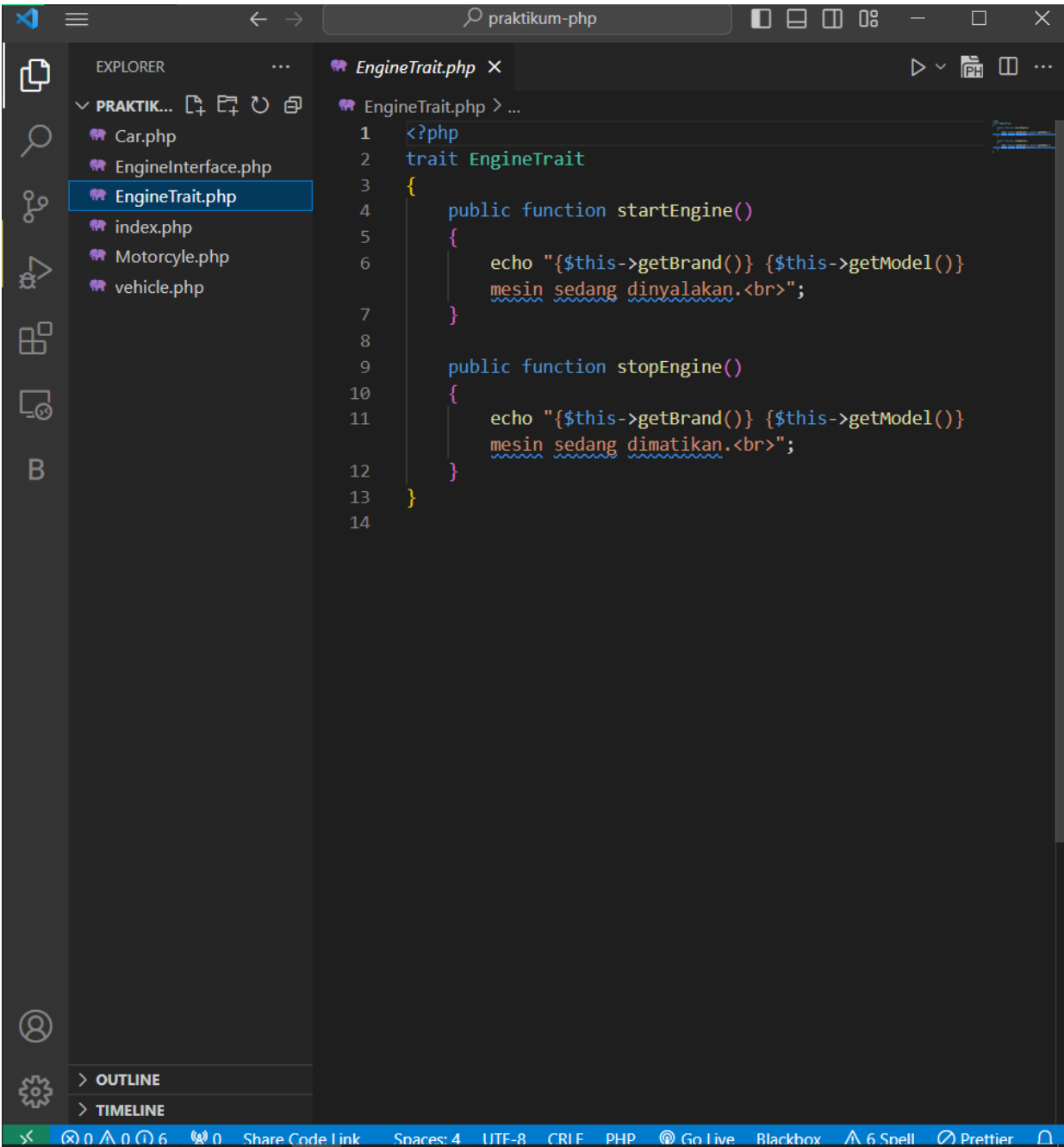
Engineinterface



The screenshot shows a Visual Studio Code editor window with a dark theme. The Explorer sidebar on the left displays a project named 'PRAKTIKUM-PHP' containing several files: 'Car.php', 'EngineInterface.php' (selected), 'EngineTrait.php', 'index.php', 'Motorcycle.php', and 'vehicle.php'. The main editor area shows the content of 'EngineInterface.php', which defines a PHP interface with two methods. The status bar at the bottom indicates the file is encoded in UTF-8, uses CRLF line endings, and is a PHP file. Various extensions like Go Live, Blackbox, Snell, and Prettier are also visible.

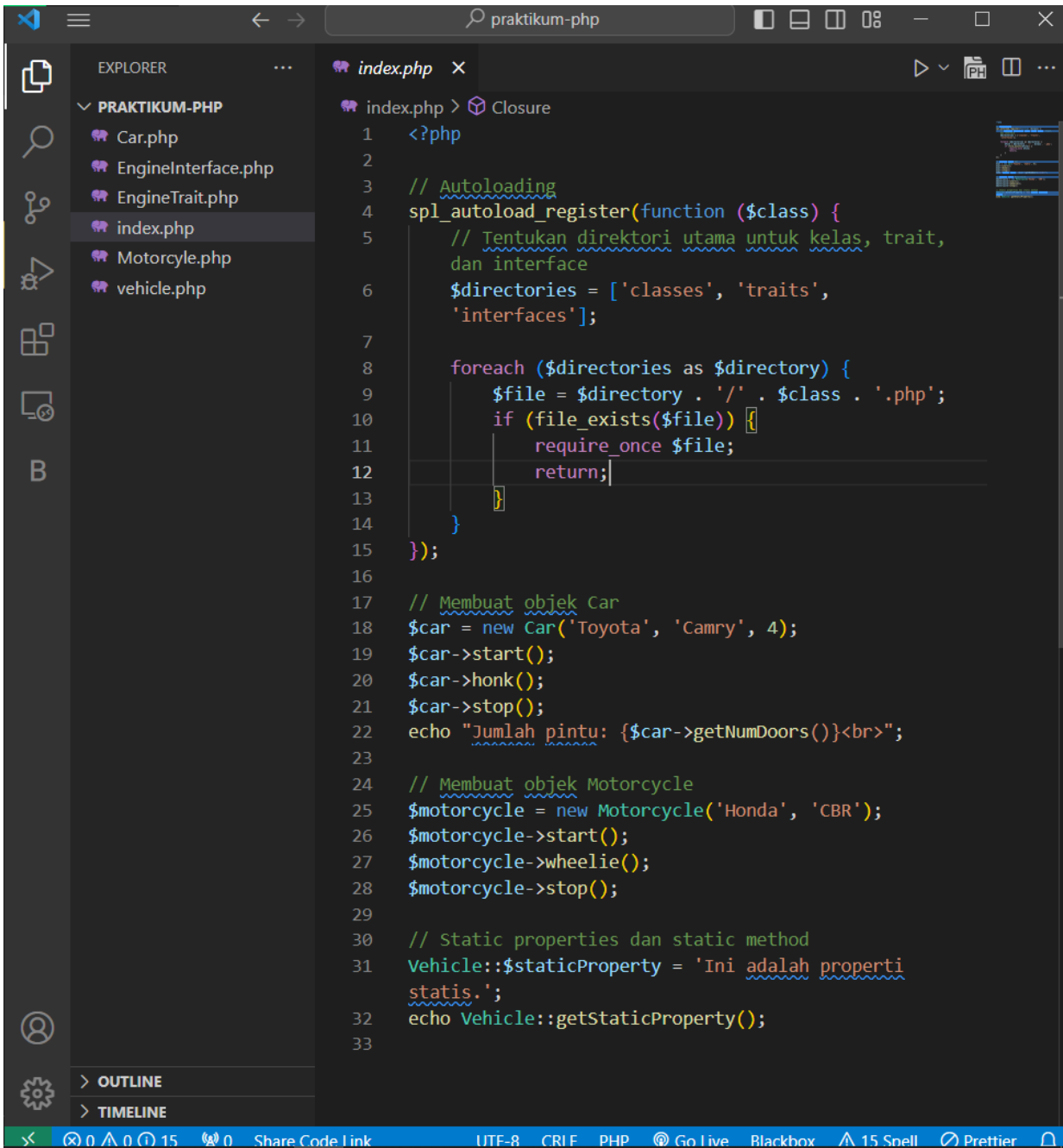
```
1 <?php
2 interface EngineInterface
3 {
4     public function startEngine();
5     public function stopEngine();
6 }
7
```

enginetrait



```
1 <?php
2 trait EngineTrait
3 {
4     public function startEngine()
5     {
6         echo "{$this->getBrand()} {$this->getModel()}"
7         mesin sedang dinyalakan.<br>;
8     }
9     public function stopEngine()
10    {
11        echo "{$this->getBrand()} {$this->getModel()}"
12        mesin sedang dimatikan.<br>;
13    }
14 }
```

index



```
1 <?php
2
3 // Autoloading
4 spl_autoload_register(function ($class) {
5     // Tentukan direktori utama untuk kelas, trait,
    dan interface
6     $directories = ['classes', 'traits',
    'interfaces'];
7
8     foreach ($directories as $directory) {
9         $file = $directory . '/' . $class . '.php';
10        if (file_exists($file)) {
11            require_once $file;
12            return;
13        }
14    }
15 });
16
17 // Membuat objek Car
18 $car = new Car('Toyota', 'Camry', 4);
19 $car->start();
20 $car->honk();
21 $car->stop();
22 echo "Jumlah pintu: {" . $car->getNumDoors() . "<br>";
23
24 // Membuat objek Motorcycle
25 $motorcycle = new Motorcycle('Honda', 'CBR');
26 $motorcycle->start();
27 $motorcycle->wheelie();
28 $motorcycle->stop();
29
30 // Static properties dan static method
31 Vehicle::$staticProperty = 'Ini adalah properti
    statis.';
32 echo Vehicle::getStaticProperty();
33
```

Hasil

