

Nama : Anak Agung Gede Bagus Wiswaprayadnya Natha

NIM : 2305551113

Kelas : Pemrograman Internet B

Implementasi Konsep OOP dasar dalam PHP beserta Instalasi Composer dan Laravel

1. Kode Program

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Control Panel Pesawat</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f0f8ff;
      text-align: center;
      padding-top: 50px;
    }
    .container {
      background-color: #fff;
      border-radius: 10px;
      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
      display: inline-block;
      padding: 30px;
    }
    h1 {
      font-size: 24px;
      color: #333;
    }
    button {
      background-color: #4CAF50;
      color: white;
      padding: 10px 20px;
      border: none;
      border-radius: 5px;
      cursor: pointer;
      margin: 10px;
      font-size: 16px;
    }
    button:hover {
      background-color: #45a049;
    }
    input {
      padding: 8px;
      font-size: 16px;
      border-radius: 5px;
      border: 1px solid #ccc;
      margin-bottom: 10px;
    }
    .output {
      margin-top: 20px;
      font-size: 18px;
    }
```

```

        color: #333;
    }
</style>
</head>
<body>

<div class="container">
    <h1>Control Panel Pesawat</h1>

    <form method="post">
        <button type="submit" name="action" value="fly">Lepas
Landas</button>
        <button type="submit" name="action"
value="land">Mendarat</button><br>

        <input type="text" name="terminal" placeholder="Enter terminal">
        <button type="submit" name="action"
value="taxi">Taxi</button><br>

        <button type="submit" name="action" value="checkstate">Cek
Status</button>
    </form>

    <div class="output">
    <?php
    session_start();

    class Pesawat {
        public $nama;
        public $model;
        public $manufaktur;
        private $state;

        public function __construct() {
            if (isset($_SESSION['state'])) {
                $this->state = $_SESSION['state'];
            } else {
                $this->state = 'Mendarat';
                $_SESSION['state'] = $this->state;
            }
        }

        public function fly() {
            if ($this->state == 'Mendarat' || $this->state == 'Taxi') {
                $this->state = 'Mengudara';
                $_SESSION['state'] = $this->state;
                echo("Pesawat Lepas Landas<br>");
            } else {
                echo("Pesawat tidak dalam posisi untuk lepas
landas!<br>");
            }
        }

        public function land() {
            if ($this->state == 'Mengudara') {
                $this->state = 'Mendarat';
                $_SESSION['state'] = $this->state;
                echo("Pesawat Mendarat<br>");
            } else {
                echo("Pesawat tidak dalam posisi untuk mendarat!<br>");
            }
        }
    }
}

```

```

    }

    public function taxi($terminal) {
        if ($this->state == 'Mendarat' || $this->state == 'Taxi' ||
$this->state == 'Mengudara') {
            $this->state = 'Taxi';
            $_SESSION['state'] = $this->state;
            echo("Pesawat telah taxi ke $terminal!<br>");
        } else {
            echo("Pesawat tidak dalam posisi untuk taxi<br>");
        }
    }

    public function checkstate() {
        echo("[STATE CHECK] : State pesawat adalah $this-
>state!<br>");
    }
}

$pesawat1 = new Pesawat();

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $action = $_POST['action'];

    switch ($action) {
        case 'fly':
            $pesawat1->fly();
            break;
        case 'land':
            $pesawat1->land();
            break;
        case 'taxi':
            $terminal = isset($_POST['terminal']) &&
!empty($_POST['terminal']) ? $_POST['terminal'] : 'Terminal 01';
            $pesawat1->taxi($terminal);
            break;
        case 'checkstate':
            $pesawat1->checkstate();
            break;
    }
}
?>
</div>
</div>

</body>
</html>

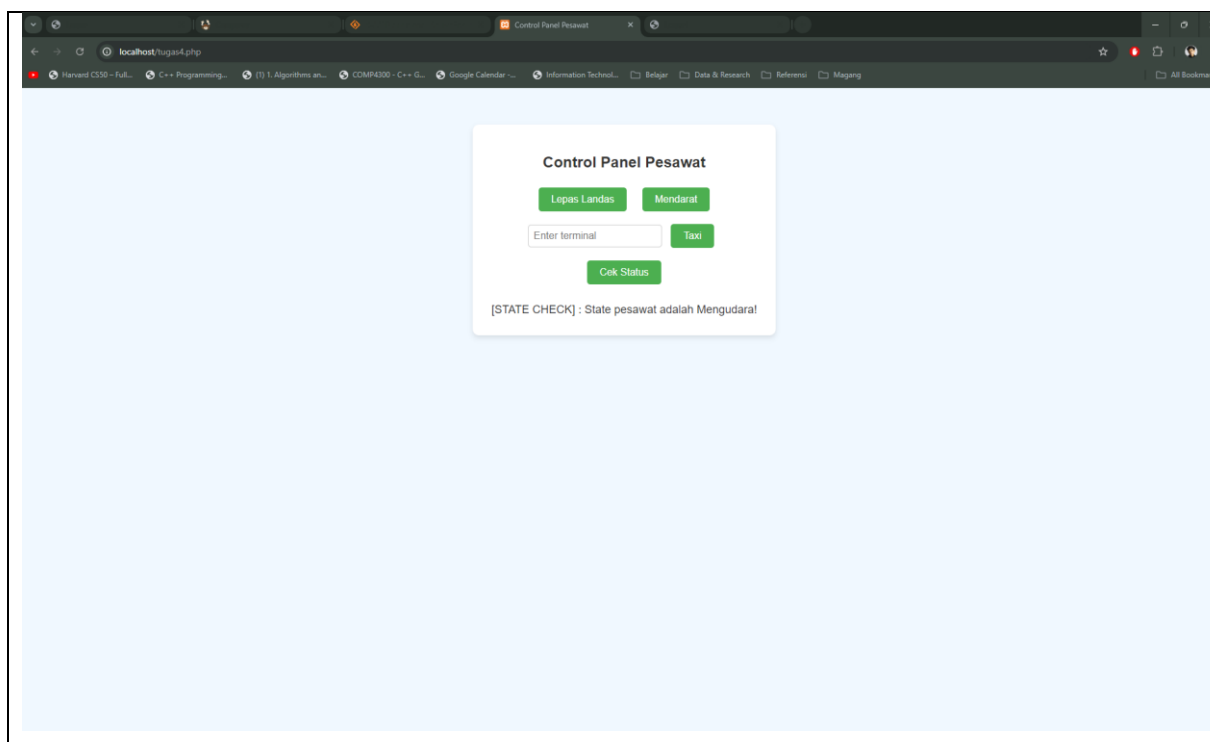
```

Kode program ini digunakan untuk mengontrol sebuah pesawat secara virtual berdasarkan tindakan yang dipilih oleh pengguna. Pengguna dapat melakukan berbagai aksi seperti lepas landas, mendarat, dan melakukan taxi melalui form yang berisi tombol-tombol pada halaman web. Tindakan pengguna, seperti lepas landas atau mendarat, diproses oleh server menggunakan metode POST. Pengguna juga dapat memasukkan nama terminal ketika memilih aksi taxi, dan pesawat akan bergerak ke terminal tersebut jika dalam kondisi yang

memungkinkan. Semua kondisi pesawat disimpan menggunakan sesi (session) sehingga statusnya dapat dipertahankan selama pengguna berinteraksi dengan halaman.

Fungsi `fly()` mengizinkan pesawat lepas landas hanya jika status pesawat dalam kondisi mendarat atau taxi. Sedangkan, fungsi `land()` memungkinkan pesawat mendarat jika sedang mengudara. Fungsi `taxi()` akan menggerakkan pesawat menuju terminal tertentu jika pesawat dalam status mendarat, taxi, atau mengudara. Selain itu, pengguna dapat mengecek status pesawat melalui tombol "Cek Status," yang akan menampilkan kondisi terbaru dari pesawat (misalnya, apakah pesawat sedang mengudara, mendarat, atau dalam proses taxi). Sistem ini mempertahankan status pesawat melalui sesi, sehingga status tidak hilang saat halaman di-refresh.

2. Tampilan Implementasi Konsep OOP dasar dalam PHP

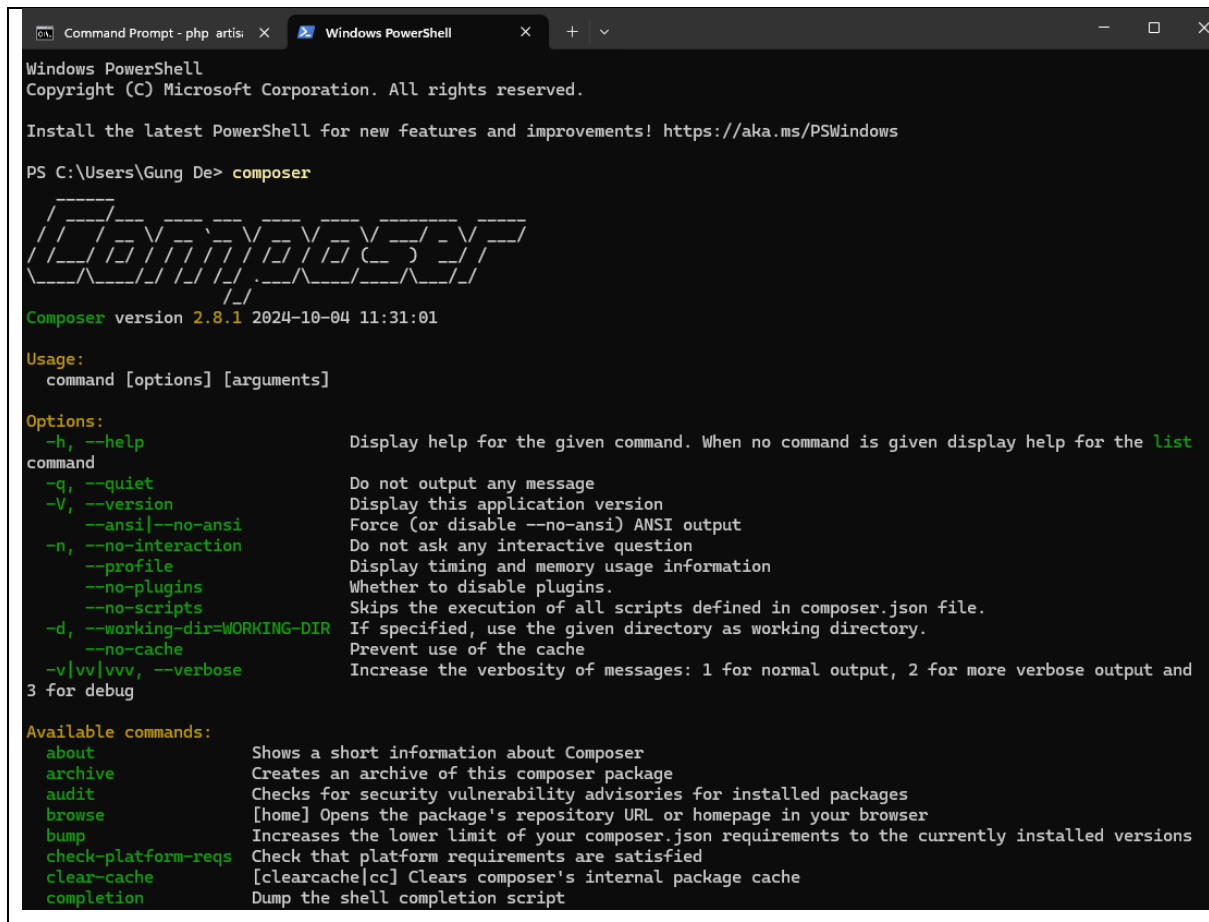


Gambar di atas menunjukkan hasil eksekusi script `.php` yang mengandung elemen HTML dan CSS. Dapat diperhatikan bahwa data hasil eksekusi ditampilkan dalam bentuk kontainer, yang berfungsi sebagai panel kontrol untuk pesawat. Pada gambar, pengguna dapat melihat tombol-tombol yang memungkinkan mereka mengontrol berbagai fungsi pesawat, seperti lepas landas, mendarat, dan melakukan taxi ke terminal tertentu. Setiap aksi yang

dilakukan oleh pengguna akan memengaruhi status pesawat, yang disimpan dan dipertahankan menggunakan sesi (session) sehingga status pesawat tetap konsisten di setiap interaksi.

Ketika pengguna memilih tombol untuk lepas landas, pesawat hanya akan mengudara jika dalam kondisi mendarat atau taxi, sedangkan tombol mendarat hanya berfungsi jika pesawat sedang mengudara. Pada aksi taxi, pengguna dapat memasukkan nama terminal untuk mengarahkan pesawat ke tujuan yang diinginkan, dan status pesawat diperbarui menjadi taxi. Status pesawat, termasuk kondisi saat ini, ditampilkan kepada pengguna dengan menggunakan tombol "Cek Status," sehingga pengguna selalu dapat mengetahui kondisi terbaru pesawat, apakah sedang mengudara, mendarat, atau dalam proses taxi.

3. Hasil Instalasi Composer



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

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Gung De> composer

Composer version 2.8.1 2024-10-04 11:31:01

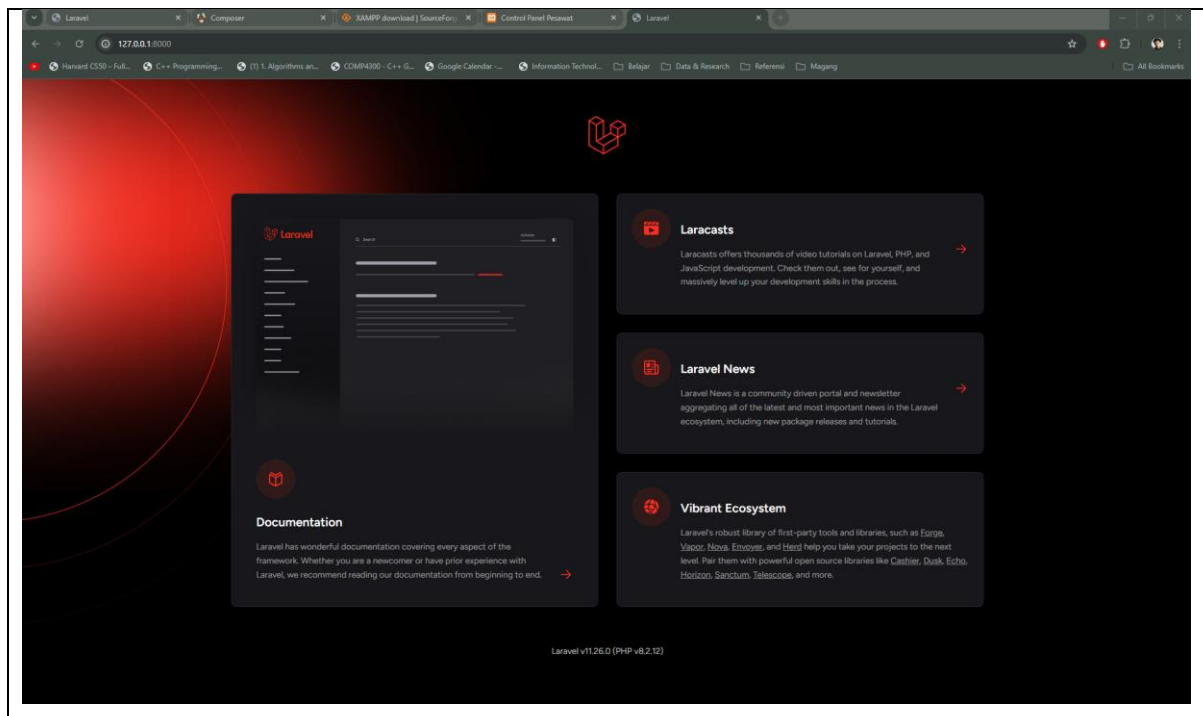
Usage:
  command [options] [arguments]

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
```

Gambar di atas merupakan hasil instalasi dari Composer. Composer dapat diinstall dengan mengakses laman download dari website penyedia Composer sendiri. Pengguna selanjutnya dapat mengikuti instruksi, terutama menspesifikasikan folder yang digunakan dalam instalasi php. Apabila sudah terinstall, pengguna dapat membuka kembali terminal, dan memanggil perintah 'composer' untuk menampilkan list perintah composer apabila telah terinstall.

4. Hasil Instalasi Laravel



Gambar di atas memperlihatkan hasil instalasi laravel pada sistem. Laravel dapat diinstall melalui Composer yang telah diinstall sebelumnya. Untuk mencapai hal tersebut, pengguna dapat memanggil perintah `composer global require laravel/installer` pada terminal. Selanjutnya Composer akan mencari dan mengunduh library laravel secara otomatis ke instalasi php yang dispesifikasikan pada instalasi composer sebelumnya. Apabila telah berhasil dihimpun oleh composer, selanjutnya pengguna bisa memanggil perintah `laravel new example-app` untuk mengarahkan php membangun example project dari library laravel, sebagai penanda bahwa laravel sudah terinstall dengan benar. Untuk dapat melihat hasil dari web-app yang telah disusun, pengguna dapat memanggil perintah `php artisan serve` agar dapat mengakses laman web app melalui browser pengguna.