



**KERJA PRAKTIK - IF184801**

**Perancangan dan Implementasi Visitor Management System  
Pada PT. PLN (Persero) Kantor Pusat Untuk Kendaraan  
Ganjil Genap**

PT. PLN (Persero) Kantor Pusat - Jakarta  
Jl. Trunojoyo 135, Melawai, Kec. Kby. Baru, Kota Jakarta  
Selatan, Daerah Khusus Ibukota Jakarta 12160  
Periode: 18 Desember 2023 - 15 Maret 2024

**Oleh:**

Ravin Pradhitya

5025201068

**Pembimbing Jurusan**  
Victor Hariadi, S.Si., M.Kom.  
**Pembimbing Lapangan**  
Detty Elviany

DEPARTEMEN TEKNIK INFORMATIKA  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya 2024



**KERJA PRAKTIK - IF184801**

**Perancangan dan Implementasi Visitor Management System  
Pada PT. PLN (Persero) Kantor Pusat Untuk Kendaraan  
Ganjil Genap**

PT. PLN (Persero) Kantor Pusat - Jakarta  
Jl. Trunojoyo 135, Melawai, Kec. Kby. Baru, Kota Jakarta Selatan,  
Daerah Khusus Ibukota Jakarta 12160  
Periode: 18 Desember 2023 - 15 Maret 2024

Oleh:

Ravin Pradhitya

5025201068

**Pembimbing Jurusan**

Victor Hariadi, S.Si., M.Kom.

**Pembimbing Lapangan**

Detty Elviany

DEPARTEMEN TEKNIK INFORMATIKA  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya 2024

*[Halaman ini sengaja dikosongkan]*

## DAFTAR ISI

<b>DAFTAR ISI</b>	iv
<b>DAFTAR GAMBAR</b>	ix
<b>DAFTAR TABEL</b>	xi
<b>LEMBAR PENGESAHAN</b>	xiii
<b>KATA PENGANTAR</b>	xvii
<b>BAB I PENDAHULUAN</b>	1
<b>1.1. Latar Belakang</b>	1
<b>1.2. Tujuan</b>	2
<b>1.3. Manfaat</b>	2
<b>1.4. Rumusan Masalah</b>	2
<b>1.5. Lokasi dan Waktu Kerja Praktik</b>	3
<b>1.6. Metodologi Kerja Praktik</b>	3
<b>1.6.1. Perumusan Masalah</b>	3
<b>1.6.2. Studi Literatur</b>	3
<b>1.6.3. Analisis dan Perancangan Sistem</b>	3
<b>1.6.4. Implementasi Sistem</b>	4
<b>1.6.5. Pengujian dan Evaluasi</b>	4
<b>1.6.6. Kesimpulan dan Saran</b>	4
<b>1.7. Sistematika Laporan</b>	4
<b>1.7.1. Bab I Pendahuluan</b>	4
<b>1.7.2. Bab II Profil Perusahaan</b>	4
<b>1.7.3. Bab III Tinjauan Pustaka</b>	4

1.7.4.	<b>Bab IV Analisis dan Perancangan Infrastruktur Sistem</b>	5
1.7.5.	<b>Bab V Implementasi Sistem</b>	5
1.7.6.	<b>Bab VI Pengujian dan Evaluasi</b>	5
1.7.7.	<b>Bab VII Kesimpulan dan Saran</b>	5
<b>BAB II PROFIL PERUSAHAAN</b>		7
2.1.	<b>Profil PT. PLN (Persero)</b>	7
2.2.	<b>Lokasi</b>	7
<b>BAB III TINJAUAN PUSTAKA</b>		9
3.1.	<b>Pemrograman Web</b>	9
3.2.	<b>HTML</b>	9
3.3.	<b>Javascript</b>	9
3.4.	<b>CodeIgniter</b>	10
3.5.	<b>MySQL</b>	10
<b>BAB IV ANALISIS DAN PERANCANGAN INFRASTRUKTUR SISTEM</b>		12
4.1.	<b>Analisis Sistem</b>	12
4.1.1.	<b>Definisi Umum Aplikasi</b>	12
4.2.	<b>Perancangan Infrastruktur Sistem</b>	12
4.2.1.	<b>Desain Sistem</b>	12
<b>BAB V IMPLEMENTASI SISTEM</b>		15
5.1.	<b>Implementasi Frontend</b>	15
5.1.1.	<b>Konfigurasi Codeigniter</b>	15
	Kode Sumber 5.1. Konfigurasi pada .env	15
	Kode Sumber 5.2. Konfigurasi pada App.php	16

<b>5.1.2. Konfigurasi Bootstrap</b>	16
Kode Sumber 5.3. Konfigurasi pada template.php	17
<b>5.2. Implementasi Backend</b>	17
<b>5.2.1. Konfigurasi Routing</b>	17
Kode Sumber 5.4. Konfigurasi pada Routes.php	18
<b>5.2.2. Konfigurasi Controllers</b>	18
Kode Sumber 5.5. Konfigurasi pada DashboardController.php	20
Kode Sumber 5.6. Konfigurasi pada MemberController.php	22
Kode Sumber 5.7. Konfigurasi pada SettingsController.php	25
Kode Sumber 5.8. Konfigurasi pada TransaksiController.php	28
<b>5.3. Implementasi Database</b>	29
Kode Sumber 5.9. Konfigurasi .env untuk database	29
Kode Sumber 5.10. Konfigurasi migration untuk PintuMasuk	30
Kode Sumber 5.11. Konfigurasi migration untuk PintuKeluar	31
Kode Sumber 5.12. Konfigurasi migration untuk Member	33
Kode Sumber 5.13. Konfigurasi migration untuk Transaksi	34
<b>BAB VI PENGUJIAN DAN EVALUASI</b>	36
<b>6.1. Tujuan Pengujian</b>	36
<b>6.2. Kriteria Pengujian</b>	36
<b>6.3. Skenario Pengujian</b>	36
<b>6.4. Evaluasi Pengujian</b>	37
<b>BAB VII KESIMPULAN DAN SARAN</b>	39
<b>7.1. Kesimpulan</b>	39
<b>7.2. Saran</b>	39
<b>DAFTAR PUSTAKA</b>	41



*[Halaman ini sengaja dikosongkan]*



## **DAFTAR GAMBAR**

Gambar 4.1. Cara Kerja MVC.....	13
---------------------------------	----

*[Halaman ini sengaja dikosongkan]*

## **DAFTAR TABEL**

*[Halaman ini sengaja dikosongkan]*

**LEMBAR PENGESAHAN  
KERJA PRAKTIK**

**Perancangan dan Implementasi Visitor Management  
System Pada PT. PLN (Persero) Kantor Pusat Untuk  
Ganjil Genap**

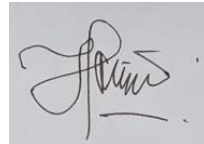
Oleh:

Ravin Pradhitya

5025201068

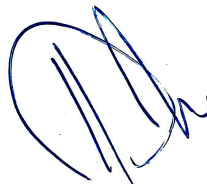
Disetujui oleh Pembimbing Kerja Praktik:

1. Victor Hariadi, S.Si.,  
M.Kom.  
NIP. 196912281994121001



(Pembimbing Departemen)

2. Detty Elviany



(Pembimbing Lapangan)

*[Halaman ini sengaja dikosongkan]*

# **Perancangan dan Implementasi Visitor Management System Pada PT. PLN (Persero) Kantor Pusat Untuk Kendaraan Ganjil Genap**

Nama Mahasiswa : Ravin Pradhitya  
NRP : 5025201068  
Departemen : Teknik Informatika FTEIC-ITS  
Pembimbing Departemen : Victor Hariadi, S.Si., M.Kom.  
Pembimbing Lapangan : Detty Elviany

## **ABSTRAK**

*PT. PLN (Persero) merupakan badan usaha milik negara (BUMN) yang bergerak di bidang ketenagalistrikan. Produk yang kami kerjakan saat melakukan Kerja Praktik adalah VMS (Visitor Management System), yaitu untuk meningkatkan keamanan dan efisiensi operasional dalam mengelola kunjungan, khususnya terkait dengan aturan ganjil genap untuk kendaraan. Pengguna utama aplikasi yang kami kembangkan adalah staf administrasi dan petugas keamanan.*

*Aplikasi dibuat dengan menggunakan framework CodeIgniter dan memiliki beberapa fitur, beberapa diantaranya adalah registrasi kunjungan, verifikasi ganjil genap, dan rekam jejak kunjungan. Kami diminta untuk menyiapkan arsitektur yang tepat agar sistem dapat diakses oleh staf administrasi dan petugas keamanan dengan kendala seminimal mungkin. Aplikasi ini menggunakan Bootstrap sebagai frontend serta menggunakan MySQL sebagai database.*

***Kata Kunci : Website, VMS, CodeIgniter, Bootstrap, MySQL***

*[Halaman ini sengaja dikosongkan]*



## **KATA PENGANTAR**

Puji syukur penulis panjatkan kepada Allah SWT atas penyertaan dan karunia-Nya sehingga penulis dapat menyelesaikan salah satu kewajiban penulis sebagai mahasiswa Departemen Teknik Informatika ITS yaitu Kerja Praktik yang berjudul: Perancangan dan Implementasi Visitor Management System Pada PT. PLN (Persero) Kantor Pusat Untuk Kendaraan Ganjil Genap.

Penulis menyadari bahwa masih banyak kekurangan baik dalam melaksanakan kerja praktik maupun penyusunan buku laporan kerja praktik ini. Namun penulis berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi.

Melalui buku laporan ini penulis juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu menyusun laporan kerja praktik baik secara langsung maupun tidak langsung antara lain:

1. Kedua orang tua penulis.
2. Bapak Victor Hariadi, S.Si., M.Kom., selaku dosen pembimbing kerja praktik.
3. Ibu Detty Elviany selaku pembimbing lapangan selama kerja praktik berlangsung.
4. Teman-teman penulis yang senantiasa memberikan semangat ketika penulis melaksanakan KP.

Surabaya, 15 April 2024  
Ravin Pradhitya

*[Halaman ini sengaja dikosongkan]*

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Visitor Management System (VMS) adalah sistem yang digunakan untuk mengelola dan memantau kunjungan tamu di suatu tempat. PT. PLN (Persero) Kantor Pusat membutuhkan VMS untuk kendaraan ganjil genap. Kendaraan ganjil genap adalah kebijakan yang diterapkan di beberapa kota di Indonesia untuk mengurangi kemacetan dengan membatasi kendaraan yang boleh melintas di jalan berdasarkan nomor plat kendaraan. Tujuan VMS pada PT. PLN (Persero) Kantor Pusat adalah untuk memudahkan pengelolaan kunjungan tamu dan kendaraan yang masuk ke kantor pusat, serta memastikan bahwa kendaraan yang masuk sesuai dengan aturan ganjil genap yang berlaku. VMS dapat membantu meningkatkan keamanan dan efisiensi dalam pengelolaan kunjungan tamu dan kendaraan. Selain itu, VMS juga dapat memberikan data dan laporan yang akurat mengenai kunjungan tamu dan kendaraan yang masuk ke kantor pusat.

Dengan memanfaatkan sistem VMS, PT. PLN (Persero) Kantor Pusat dapat secara efisien memantau dan mengelola setiap kunjungan dengan lebih terstruktur. Sistem ini dirancang untuk mencatat data identitas setiap pengunjung, termasuk nomor kendaraan yang digunakan. Dalam konteks ganjil genap, VMS dapat dioptimalkan untuk memeriksa kelengkapan surat izin kendaraan yang diperlukan sesuai dengan aturan lalu lintas yang berlaku. Hal ini dapat membantu perusahaan menjaga keamanan serta memastikan bahwa kendaraan yang masuk ke area kantor sesuai dengan ketentuan lalu lintas yang berlaku.

Melalui perancangan dan implementasi VMS ini, PT. PLN (Persero) Kantor Pusat dapat meningkatkan efisiensi operasional, meningkatkan tingkat keamanan, dan mendukung ketaatan terhadap regulasi lalu lintas, khususnya terkait dengan aturan ganjil genap. Sebagai bagian dari upaya perusahaan dalam menerapkan teknologi modern, VMS juga mencerminkan komitmen PT. PLN (Persero) dalam mengadopsi inovasi untuk meningkatkan pengelolaan dan pengawasan kegiatan operasional sehari-hari.

## **1.2. Tujuan**

Tujuan kerja praktik ini adalah menyelesaikan kewajiban nilai kerja praktik sebesar 4 sks dan membantu PT. PLN (Persero) Kantor Pusat untuk memastikan bahwa kendaraan yang masuk telah memenuhi aturan ganjil genap.

## **1.3. Manfaat**

Manfaat yang diperoleh dengan adanya Visitor Management System antara lain adalah meningkatkan efisiensi, keamanan, dan akurasi dalam pengelolaan kunjungan tamu dan kendaraan, khususnya terkait aturan ganjil genap.

## **1.4. Rumusan Masalah**

Rumusan masalah dari kerja praktik ini adalah sebagai berikut:

1. Bagaimana meningkatkan keamanan lingkungan kantor PT. PLN (Persero) Kantor Pusat dengan implementasi Visitor Management System (VMS) yang terintegrasi?

2. Bagaimana VMS dapat mendukung penerapan aturan ganjil genap dalam memverifikasi dan memantau kendaraan yang memasuki area kantor PT. PLN (Persero) Kantor Pusat?

## **1.5. Lokasi dan Waktu Kerja Praktik**

Kerja praktik ini dilaksanakan pada waktu dan tempat sebagai berikut:

Lokasi : PT. PLN (Persero) Kantor Pusat

Waktu : 18 Desember 2023 - 15 Maret 2024

## **1.6. Metodologi Kerja Praktik**

### **1.6.1. Perumusan Masalah**

Dalam tahap ini kami perlu mengetahui permasalahan apa saja yang terjadi dan dapat diselesaikan atau dioptimasi. Lalu kami juga perlu mengetahui semua kebutuhan dalam permasalahan tersebut.

### **1.6.2. Studi Literatur**

Setelah ditentukan rumusan masalah mengenai sistem yang akan dibuat, dilakukan studi literatur mengenai implementasinya. Pada tahap ini dilakukan proses pencarian, pembelajaran, dan pengumpulan informasi yang berkaitan dengan implementasi sistem yang akan dibuat. Informasi dapat diperoleh dari internet ataupun dari proyek sebelumnya yang serupa dan memungkinkan untuk diimplementasikan.

### **1.6.3. Analisis dan Perancangan Sistem**

Tahap ini meliputi penjelasan mengenai hasil dari studi literatur yang dilakukan. Dari beberapa metode yang ditemukan saat literasi dianalisa metode mana

yang paling tepat dan efektif untuk digunakan untuk menyelesaikan permasalahan. Ditentukan bahasa pemrograman yang akan digunakan, serta batasan data yang akan digunakan, sehingga dapat memunculkan hasil yang diharapkan.

#### **1.6.4. Implementasi Sistem**

Implementasi merupakan realisasi dari tahap perancangan. Pada tahap ini kami Melakukan *deployment* pada aplikasi yang telah dibuat oleh tim developer.

#### **1.6.5. Pengujian dan Evaluasi**

Setelah website yang telah direncanakan telah jadi, perlu adanya evaluasi untuk menguji apakah website sesuai dengan harapan client. Jika masih belum sesuai atau perlu menambah fitur, rapat akan dilakukan lagi untuk mem-*floor*-kan fitur-fitur apa saja yang perlu diperbaiki atau ditambah.

#### **1.6.6. Kesimpulan dan Saran**

Pengujian yang dilakukan ini telah memenuhi syarat yang diinginkan, dan berjalan dengan baik dan lancar.

### **1.7. Sistematika Laporan**

#### **1.7.1. Bab I Pendahuluan**

Bab ini berisi latar belakang, tujuan, manfaat, rumusan masalah, lokasi dan waktu kerja praktik, metodologi, dan sistematika laporan.

#### **1.7.2. Bab II Profil Perusahaan**

Bab ini berisi gambaran umum PT. PLN (Persero), lokasi perusahaan.

#### **1.7.3. Bab III Tinjauan Pustaka**

Bab ini berisi dasar teori dari teknologi yang digunakan dalam menyelesaikan proyek kerja praktik.

**1.7.4. Bab IV Analisis dan Perancangan Infrastruktur Sistem**

Bab ini berisi mengenai tahap analisis sistem aplikasi dalam menyelesaikan proyek kerja praktik.

**1.7.5. Bab V Implementasi Sistem**

Bab ini berisi uraian tahap - tahap yang dilakukan untuk proses implementasi aplikasi.

**1.7.6. Bab VI Pengujian dan Evaluasi**

Bab ini berisi hasil uji coba dan evaluasi dari aplikasi yang telah dikembangkan selama pelaksanaan kerja praktik.

**1.7.7. Bab VII Kesimpulan dan Saran**

Bab ini berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik.

*[Halaman ini sengaja dikosongkan]*



## **BAB II**

### **PROFIL PERUSAHAAN**

#### **2.1. Profil PT. PLN (Persero)**

PT PLN (Persero) atau Perusahaan Listrik Negara, merupakan Badan Usaha Milik Negara (BUMN) yang berperan penting di sektor kelistrikan Indonesia. Terlibat dalam seluruh rantai nilai energi, PLN hadir mulai dari pembangkit listrik, pembangunan, transmisi, distribusi, hingga penjualan eceran listrik, menjadikannya pemain utama dalam penyediaan energi listrik di Indonesia.

#### **2.2. Lokasi**

Jl. Trunojoyo 135, Melawai, Kec. Kby. Baru, Kota Jakarta Selatan, DKI Jakarta 12160

*[Halaman ini sengaja dikosongkan]*

## **BAB III**

### **TINJAUAN PUSTAKA**

#### **3.1. Pemrograman Web**

Pemrograman Web adalah istilah yang terkait erat dengan internet dan pembuatan website. Ini merupakan proses pembuatan situs web untuk digunakan di internet. Istilah lain untuk web yang dikenal oleh banyak orang adalah WWW atau World Wide Web.

World Wide Web adalah kumpulan halaman website yang saling terhubung melalui hyperlink, membentuk jaringan informasi yang besar dan menggunakan protokol HTTP (HyperText Transfer Protocol).

#### **3.2. HTML**

HTML merupakan sebuah markup language atau bahasa penandaan yang terdiri dari serangkaian tag. Fungsinya adalah untuk memberikan instruksi kepada browser tentang bagaimana cara menampilkan isi halaman web. HTML adalah standar dalam internet yang ditetapkan dan diawasi oleh World Wide Web Consortium (W3C).

#### **3.3. Javascript**

JavaScript adalah bahasa skrip yang dijalankan di sisi klien, digunakan untuk membuat efek animasi dan interaktif pada halaman web. Banyaknya library JavaScript memungkinkan pengembang meningkatkan interaktivitas halaman web. Diperlukan browser yang mendukung JavaScript untuk mengeksekusi skrip JavaScript di sisi klien.

### **3.4. CodeIgniter**

CodeIgniter merupakan sebuah framework pengembangan web yang terkenal, menggunakan Model View Controller (MVC), dan bersifat open-source. CodeIgniter menyediakan beragam library yang luas dan memiliki bobot yang ringan, sehingga tidak memberatkan sumber daya website. Dengan menerapkan pola MVC, framework ini memungkinkan pengembang untuk membangun aplikasi dengan struktur yang lebih terorganisir.

### **3.5. MySQL**

MySQL adalah sistem manajemen database relasional open-source (RDBMS) yang menggunakan bahasa SQL dan operasi client-server. Cocok untuk berbagai keperluan seperti data warehousing, e-commerce, dan logging aplikasi. Dikembangkan awalnya oleh MySQL AB, kini dimiliki oleh Oracle. MySQL terkenal karena kecanggihannya, stabilitas, dan kecepatan tinggi dalam mengelola dan mengakses data, serta kompatibilitasnya dengan berbagai platform seperti Linux, macOS, Windows, dan Ubuntu.

*[Halaman ini sengaja dikosongkan]*

## **BAB IV**

# **ANALISIS DAN PERANCANGAN INFRASTRUKTUR SISTEM**

### **4.1. Analisis Sistem**

Pada bab ini akan dijelaskan mengenai tahapan dalam membangun infrastruktur aplikasi sistem penerimaan peserta didik baru yaitu analisis dari infrastruktur sistem yang akan dibangun. Hal tersebut dijelaskan ke dalam dua bagian, definisi umum aplikasi dan analisis kebutuhan.

#### **4.1.1. Definisi Umum Aplikasi**

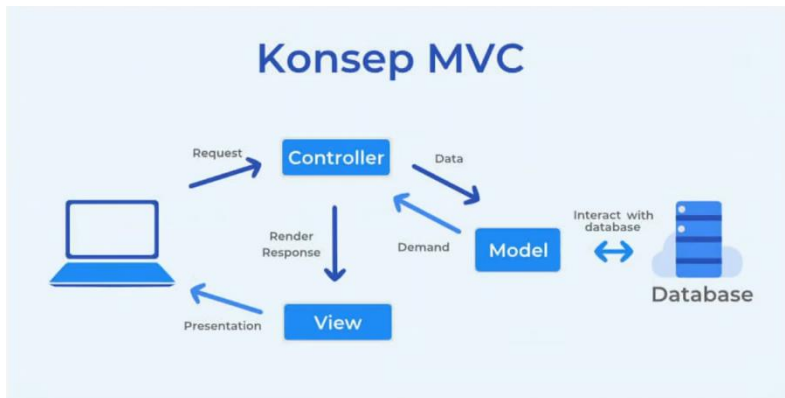
Secara umum, aplikasi Visitor Management System (VMS) merupakan sistem untuk mengelola kunjungan ke suatu lokasi dengan otomatisasi pendaftaran dan pemantauan, bertujuan meningkatkan keamanan dan efisiensi operasional fasilitas tersebut. Dengan menggunakan VMS, pengguna dapat mencatat informasi identitas pengunjung, mengelola akses ke fasilitas, dan memantau aktivitas kunjungan secara efisien.

### **4.2. Perancangan Infrastruktur Sistem**

#### **4.2.1. Desain Sistem**

Desain arsitektur pada Visitor Management System PT. PLN (Persero) ini menggunakan CodeIgniter sebagai framework, Bootstrap sebagai frontend, dan MySQL sebagai database memiliki struktur yang terorganisir dengan baik. Model berfungsi sebagai pengelola data di bagian back-end, sementara view bertanggung jawab atas tampilan visual yang dihasilkan untuk pengguna dengan dukungan Bootstrap. Controller bertindak sebagai penghubung antara model dan view, mengatur logika aplikasi dan

respons terhadap aksi pengguna. MySQL digunakan sebagai sistem manajemen basis data untuk menyimpan dan mengelola data aplikasi. Frontend menggunakan Bootstrap untuk tampilan yang menarik dan responsif, sementara backend memanfaatkan CodeIgniter untuk mengelola server, database, dan logika aplikasi secara efisien.



Gambar 1. Cara Kerja MVC

Dengan desain ini, aplikasi dapat beroperasi dengan efisiensi tinggi dan terstruktur dengan baik. Setiap komponen berinteraksi satu sama lain secara koheren, memungkinkan pengelolaan data yang mudah dan efektif. Dengan dukungan CodeIgniter dan Bootstrap, pengembangan aplikasi dapat dilakukan dengan lebih cepat dan tersusun, meningkatkan produktivitas dan kualitas aplikasi secara keseluruhan.

*[Halaman ini sengaja dikosongkan]*



## **BAB V**

### **IMPLEMENTASI SISTEM**

Bab ini membahas tentang implementasi dari sistem yang kami buat. Implementasi ini akan dibagi ke dalam beberapa bagian, yaitu bagian frontend, backend, dan database.

#### **5.1. Implementasi Frontend**

Implementasi frontend ini menggunakan Bootstrap. Adapun implementasinya sebagai berikut:

##### **5.1.1. Konfigurasi Codeigniter**

Konfigurasi CodeIgniter melibatkan beberapa langkah untuk mengatur `base_url`. Berikut adalah Kode Sumber:

```
#-----  
-----  
# ENVIRONMENT  
#-----  
-----  
  
CI_ENVIRONMENT = production  
  
#-----  
-----  
# APP  
#-----  
-----  
  
app.baseURL = 'https://dashboarparkirpln.com/'  
# If you have trouble with `.` , you could also use `\_` .  
# app_baseURL = ''  
# app.forceGlobalSecureRequests = false  
# app.CSPEnabled = false
```

Kode Sumber 5.1. Konfigurasi pada `.env`

```

<?php

namespace Config;

use CodeIgniter\Config\BaseConfig;

class App extends BaseConfig
{
    /**
     * -----
     * Base Site URL
     * -----
     *
     * URL to your CodeIgniter root. Typically, this will be
     your base URL,
     * WITH a trailing slash:
     *
     * E.g., http://example.com/
     */
    public string $baseURL = 'https://dashboarparkirp1n.com/';
}

```

Kode Sumber 5.2. Konfigurasi pada App.php

### 5.1.2. Konfigurasi Bootstrap

Untuk menggunakan Bootstrap, perlu mengimpor CSS dan JavaScript Bootstrap ke dalam file HTML. Konfigurasi tersebut akan ditampilkan pada Kode Sumber berikut:

```

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootst
rap.bundle.min.js" crossorigin="anonymous"></script>
<script src="<?= base_url('asset-
admin'); ?>/js/scripts.js"></script>

```

```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.8.0/Chart
.min.js" crossorigin="anonymous"></script>
  <script src="<?= base_url('asset-admin'); ?>/demo/chart-
area-demo.js"></script>
  <script src="<?= base_url('asset-admin'); ?>/demo/chart-bar-
demo.js"></script>
  <script src="https://cdn.jsdelivrivr.net/npm/simple-
datatables@7.1.2/dist/umd/simple-datatables.min.js"
crossorigin="anonymous"></script>
  <script src="<?= base_url('asset-admin'); ?>/js/datatables-
simple-demo.js"></script>

```

Kode Sumber 5.3. Konfigurasi pada template.php

## 5.2. Implementasi Backend

Implementasi backend ini menggunakan CodeIgniter. Adapun implementasinya sebagai berikut:

### 5.2.1. Konfigurasi Routing

Konfigurasi CodeIgniter melibatkan beberapa langkah untuk mengatur berbagai aspek aplikasi, termasuk pengaturan routing. Berikut adalah Kode Sumber:

```

<?php

use CodeIgniter\Router\RouteCollection;

/**
 * @var RouteCollection $routes
 */
$routes->get('/', 'Admin\DashboardController::index');

// route admin dashboard
$routes->get('dashboard', 'Admin\DashboardController::index');

// route admin pintu masuk
$routes->get('pintu-masuk', 'Admin\SettingsController::masuk');

```

```

$routes->post('pintu-masuk/tambah',
'Admin\SettingsController::storeMasuk');
$routes->put('pintu-masuk/ubah/(:num)',
'Admin\SettingsController::updateMasuk/$1');
$routes->delete('pintu-masuk/hapus/(:num)',
'Admin\SettingsController::destroyMasuk/$1');

// route admin pintu keluar
$routes->get('pintu-keluar',
'Admin\SettingsController::keluar');
$routes->post('pintu-keluar/tambah',
'Admin\SettingsController::storeKeluar');
$routes->put('pintu-keluar/ubah/(:num)',
'Admin\SettingsController::updateKeluar/$1');
$routes->delete('pintu-keluar/hapus/(:num)',
'Admin\SettingsController::destroyKeluar/$1');

// route admin member
$routes->get('member', 'Admin\MemberController::index');
$routes->post('member/tambah', 'Admin\MemberController::store');
$routes->put('member/ubah/(:num)',
'Admin\MemberController::update/$1');
$routes->delete('member/hapus/(:num)',
'Admin\MemberController::destroy/$1');

// route admin transaksi
$routes->get('transaksi', 'Admin\TransaksiController::index');
$routes->get('transaksi/tambah',
'Admin\TransaksiController::form_create');
$routes->post('transaksi/create-transaksi',
'Admin\TransaksiController::create_transaksi');
$routes->put('transaksi/selesai-transaksi/(:num)',
'Admin\TransaksiController::update_transaksi/$1');
$routes->delete('transaksi/delete-transaksi/(:num)',
'Admin\TransaksiController::delete_transaksi/$1');

```

Kode Sumber 5.4. Konfigurasi pada Routes.php

### 5.2.2. Konfigurasi Controllers

Controllers adalah bagian dari CodeIgniter yang menangani permintaan HTTP dari pengguna dan

menjalankan tindakan yang sesuai. Detail konfigurasinya akan ditampilkan pada Kode Sumber berikut:

```
<?php
namespace App\Controllers\Admin;

use App\Controllers\BaseController;

class DashboardController extends BaseController
{
    public function index(): string
    {
        $data = [
            'title' => 'Dashboard',
            'daftar_pintuMasuk' => $this->MasukModel->limit(6)-
>find(),
            'daftar_pintuKeluar' => $this->KeluarModel-
>limit(5)->find(),
            'daftar_member' => $this->MemberModel->limit(3)-
>find(),
            'daftar_transaksi' => $this->TransaksiModel-
>limit(3)->find(),
            'kendaraan_ganjil' => $this->TransaksiModel-
>where('keterangan_plat', 'ganjil')->where('waktu_masuk >=',
date('Y-m-d 00:00:00'))->where('waktu_masuk <', date('Y-m-d
23:59:59'))->countAllResults(),
            'kendaraan_genap' => $this->TransaksiModel-
>where('keterangan_plat', 'genap')->where('waktu_masuk >=',
date('Y-m-d 00:00:00'))->where('waktu_masuk <', date('Y-m-d
23:59:59'))->countAllResults(),
            'kendaraan_masuk' => $this->TransaksiModel-
>select('nama_pintuMasuk')->where('waktu_masuk >=', date('Y-m-d
00:00:00'))->where('waktu_masuk <', date('Y-m-d 23:59:59'))-
>countAllResults(),
            'kendaraan_keluar' => $this->TransaksiModel-
```

```

>select('nama_pintuKeluar')->where('waktu_keluar >=', date('Y-m-
d 00:00:00'))->where('waktu_keluar <', date('Y-m-d 23:59:59'))-
>countAllResults(),

    ];

    return view('admin/dashboard/index', $data);
}
}
}

```

Kode Sumber 5.5. Konfigurasi pada DashboardController.php

```

<?php

namespace App\Controllers\Admin;

use App\Controllers\BaseController;
use CodeIgniter\HTTP\ResponseInterface;

class MemberController extends BaseController
{
    // member
    public function index()
    {
        $data = [
            'title' => 'Member',
            'daftar_member' => $this->MemberModel-
>orderBy('no_kartu', 'ASCE')->findAll()
        ];

        return view('admin/member/index', $data);
    }

    // tambah member
    public function store()
    {
        $data = [

            'nama_member' => esc($this->request-
>getPost('nama_member')),

```

```

        'no_kartu' => esc($this->request-
>getPost('no_kartu')),
        'kendaraan_member' => esc($this->request-
>getPost('kendaraan_member')),
        'plat_member' => esc($this->request-
>getPost('plat_member')),
        'keterangan_plat' => esc($this->request-
>getPost('keterangan_plat')),
    ];

    $this->MemberModel->insert($data);

    return redirect()->back()->with('success', 'Data Member
Berhasil Ditambahkan');
}

// ubah pintu masuk
public function update($id_member)
{
    $data = [

        'nama_member' => esc($this->request-
>getPost('nama_member')),
        'no_kartu' => esc($this->request-
>getPost('no_kartu')),
        'kendaraan_member' => esc($this->request-
>getPost('kendaraan_member')),
        'plat_member' => esc($this->request-
>getPost('plat_member')),
        'keterangan_plat' => esc($this->request-
>getPost('keterangan_plat')),
    ];

    $this->MemberModel->update($id_member, $data);

    return redirect()->back()->with('success', 'Data Member
Berhasil Diubah');
}

// hapus member
public function destroy($id_member)

```

```

    {
        $this->MemberModel->where('id_member', $id_member)-
>delete();

        return redirect()->back()->with('success', 'Data Member
Berhasil Dihapus');
    }
}

```

Kode Sumber 5.6. Konfigurasi pada MemberController.php

```

<?php

namespace App\Controllers\Admin;

use App\Controllers\BaseController;

class SettingsController extends BaseController
{
    // pintu masuk
    public function masuk()
    {
        $data = [
            'title' => 'Pintu Masuk',
            'daftar_pintuMasuk' => $this->MasukModel-
>orderBy('kode_pintuMasuk', 'ASCE')->findAll()
        ];

        return view('admin/settings/masuk', $data);
    }

    // tambah pintu masuk
    public function storeMasuk()
    {
        $data = [

            'nama_pintuMasuk' => esc($this->request-
>getPost('nama_pintuMasuk')),
            'kode_pintuMasuk' => esc($this->request-
>getPost('kode_pintuMasuk')),

```



```

        'ip_addressMasuk' => esc($this->request-
>getPost('ip_addressMasuk')),
        'ip_camMasuk' => esc($this->request-
>getPost('ip_camMasuk')),
        'status_pintuMasuk' => esc($this->request-
>getPost('status_pintuMasuk'))
    ];

    $this->MasukModel->insert($data);

    return redirect()->back()->with('success', 'Data Pintu
Masuk Berhasil Ditambahkan');
}

// ubah pintu masuk
public function updateMasuk($id_pintuMasuk)
{
    $data = [

        'nama_pintuMasuk' => esc($this->request-
>getPost('nama_pintuMasuk')),
        'kode_pintuMasuk' => esc($this->request-
>getPost('kode_pintuMasuk')),
        'ip_addressMasuk' => esc($this->request-
>getPost('ip_addressMasuk')),
        'ip_camMasuk' => esc($this->request-
>getPost('ip_camMasuk')),
        'status_pintuMasuk' => esc($this->request-
>getPost('status_pintuMasuk'))
    ];

    $this->MasukModel->update($id_pintuMasuk, $data);

    return redirect()->back()->with('success', 'Data Pintu
Masuk Berhasil Diubah');
}

// hapus pintu masuk
public function destroyMasuk($id_pintuMasuk)
{
    $this->MasukModel->where('id_pintuMasuk',

```

```

$id_pintuMasuk)->delete());

        return redirect()->back()->with('success', 'Data Pintu
Masuk Berhasil Dihapus');
    }

    // pintu keluar
    public function keluar()
    {
        $data = [
            'title' => 'Pintu Keluar',
            'daftar_pintuKeluar' => $this->KeluarModel-
>orderBy('kode_pintuKeluar', 'ASCE')->findAll()
        ];

        return view('admin/settings/keluar', $data);
    }

    // tambah pintu keluar
    public function storeKeluar()
    {
        $data = [

            'nama_pintuKeluar' => esc($this->request-
>getPost('nama_pintuKeluar')),
            'kode_pintuKeluar' => esc($this->request-
>getPost('kode_pintuKeluar')),
            'ip_addressKeluar' => esc($this->request-
>getPost('ip_addressKeluar')),
            'ip_camKeluar' => esc($this->request-
>getPost('ip_camKeluar')),
            'status_pintuKeluar' => esc($this->request-
>getPost('status_pintuKeluar'))
        ];

        $this->KeluarModel->insert($data);

        return redirect()->back()->with('success', 'Data Pintu
Keluar Berhasil Ditambahkan');
    }

```

```

// ubah pintu keluar
public function updateKeluar($id_pintuKeluar)
{
    $data = [

        'nama_pintuKeluar' => esc($this->request-
>getPost('nama_pintuKeluar')),
        'kode_pintuKeluar' => esc($this->request-
>getPost('kode_pintuKeluar')),
        'ip_addressKeluar' => esc($this->request-
>getPost('ip_addressKeluar')),
        'ip_camKeluar' => esc($this->request-
>getPost('ip_camKeluar')),
        'status_pintuKeluar' => esc($this->request-
>getPost('status_pintuKeluar'))
    ];

    $this->KeluarModel->update($id_pintuKeluar, $data);

    return redirect()->back()->with('success', 'Data Pintu
Keluar Berhasil Diubah');
}

// hapus pintu keluar
public function destroyKeluar($id_pintuKeluar)
{
    $this->KeluarModel->where('id_pintuKeluar',
$id_pintuKeluar)->delete();

    return redirect()->back()->with('success', 'Data Pintu
Keluar Berhasil Dihapus');
}
}

```

Kode Sumber 5.7. Konfigurasi pada SettingsController.php

```

<?php

namespace App\Controllers\Admin;

```

```

use App\Controllers\BaseController;
use CodeIgniter\HTTP\ResponseInterface;

class TransaksiController extends BaseController
{
    // daftar transaksi
    public function index()
    {
        $data = [
            'title' => 'Transaksi',
            'daftar_transaksi' => $this->TransaksiModel-
>orderBy('id_transaksi', 'DESC')->findAll(),
            'daftar_pintuKeluar' => $this->KeluarModel-
>findAll()
        ];

        return view('admin/transaksi/index', $data);
    }

    // form tambah transaksi
    public function form_create()
    {
        $data = [
            'title' => 'Tambah Transaksi',
            'daftar_member' => $this->MemberModel->findAll(),
            'daftar_pintuMasuk' => $this->MasukModel->findAll(),
            'daftar_pintuKeluar' => $this->KeluarModel-
>findAll(),
            'validation' => \Config\Services::validation()
        ];

        return view('admin/transaksi/create', $data);
    }

    // form selesai transaksi
    public function form_update($id_transaksi)
    {
        $data = [
            'title' => 'Selesai Transaksi',
            'daftar_transaksi' => $this->TransaksiModel-
>find($id_transaksi),

```

```

        'daftar_member' => $this->MemberModel->findAll(),
        'daftar_pintuMasuk' => $this->MasukModel->findAll(),
        'daftar_pintuKeluar' => $this->KeluarModel-
>findAll(),
        'validation' => \Config\Services::validation()
    ];

    return view('admin/transaksi/update', $data);
}

// tambah transaksi
public function create_transaksi()
{
    // aturan validasi input
    $rules = $this->validate([
        'nama_member' => 'required',
        'no_kartu' => 'required',
        'waktu_masuk' => 'required',
        'plat_member' => 'required',
        'keterangan_plat' => 'required',
        'nama_pintuMasuk' => 'required',
    ]);

    // jika validasi gagal
    if (!$rules) {
        session()->setFlashdata('failed', 'Data Transaksi
Gagal Ditambahkan');
        return redirect()->back()->withInput();
    }

    // jika data valid
    $this->TransaksiModel->insert([
        'nama_member' => $this->request-
>getPost('nama_member'),
        'no_kartu' => $this->request->getPost('no_kartu'),
        'waktu_masuk' => $this->request-
>getPost('waktu_masuk'),
        'plat_member' => $this->request-
>getPost('plat_member'),
        'keterangan_plat' => $this->request-
>getPost('keterangan_plat'),
    ]);
}

```

```

        'nama_pintuMasuk' => $this->request-
>getPost('nama_pintuMasuk'),
        ]);

        return redirect()->to(base_url('transaksi'))-
>with('success', 'Data Transaksi Berhasil Ditambahkan');
    }

    // modal selesai transaksi
    public function update_transaksi($id_transaksi)
    {
        $data = [

            'waktu_keluar' => $this->request-
>getPost('waktu_keluar'),
            'nama_pintuKeluar' => $this->request-
>getPost('nama_pintuKeluar'),

        ];

        $this->TransaksiModel->update($id_transaksi, $data);

        return redirect()->back()->with('success', 'Data
Transaksi Berhasil Diselesaikan');
    }

    // hapus transaksi
    public function delete_transaksi($id_transaksi)
    {
        $this->TransaksiModel->delete($id_transaksi);

        return redirect()->back()->with('success', 'Data
Transaksi Berhasil Dihapus');
    }
}

```

Kode Sumber 5.8. Konfigurasi pada TransaksiController.php

### 5.3. Implementasi Database

Implementasi database ini berfokus pada konfigurasi satu database agar mampu menerima banyak request read dan write dalam satu waktu. Berikut adalah Kode Sumber:

```
database.default.hostname = localhost
database.default.database = palang_parkir
database.default.username = root
database.default.password =
database.default.DBDriver = MySQLi
database.default.DBPrefix =
database.default.port = 3306
```

Kode Sumber 5.9. Konfigurasi .env untuk database

```
<?php

namespace App\Database\Migrations;

use CodeIgniter\Database\Migration;

class PintuMasuk extends Migration
{
    public function up()
    {
        $this->forge->addField([
            'id_pintuMasuk' => [
                'type'           => 'INT',
                'constraint'     => 5,
                'unsigned'       => true,
                'auto_increment' => true,
            ],
            'nama_pintuMasuk' => [
                'type'           => 'VARCHAR',
                'constraint'     => '100',
            ],
            'kode_pintuMasuk' => [
                'type'           => 'VARCHAR',
                'constraint'     => '100',
            ],
        ],
```

```

        ],
        'ip_addressMasuk' => [
            'type'         => 'VARCHAR',
            'constraint'   => '100',
        ],
        'ip_camMasuk' => [
            'type'         => 'VARCHAR',
            'constraint'   => '100',
        ],
        'status_pintuMasuk' => [
            'type'         => 'VARCHAR',
            'constraint'   => '100',
        ],
    ],
]);
$this->forge->addKey('id_pintuMasuk', true);
$this->forge->createTable('pintu_masuk');
}

public function down()
{
    $this->forge->dropTable('pintu_masuk');
}
}

```

Kode Sumber 5.10. Konfigurasi migration untuk PintuMasuk

```

<?php

namespace App\Database\Migrations;

use CodeIgniter\Database\Migration;

class PintuKeluar extends Migration
{
    public function up()
    {
        $this->forge->addField([
            'id_pintuKeluar' => [
                'type'         => 'INT',
                'constraint'   => 5,
            ]
        ]);
    }
}

```



```

        'unsigned'      => true,
        'auto_increment' => true,
    ],
    'nama_pintuKeluar' => [
        'type'          => 'VARCHAR',
        'constraint'    => '100',
    ],
    'kode_pintuKeluar' => [
        'type'          => 'VARCHAR',
        'constraint'    => '100',
    ],
    'ip_addressKeluar' => [
        'type'          => 'VARCHAR',
        'constraint'    => '100',
    ],
    'ip_camKeluar' => [
        'type'          => 'VARCHAR',
        'constraint'    => '100',
    ],
    'status_pintuKeluar' => [
        'type'          => 'VARCHAR',
        'constraint'    => '100',
    ],
    ],
    ]);
    $this->forge->addKey('id_pintuKeluar', true);
    $this->forge->createTable('pintu_keluar');
}

public function down()
{
    $this->forge->dropTable('pintu_keluar');
}
}

```

Kode Sumber 5.11. Konfigurasi migration untuk PintuKeluar

```

<?php
namespace App\Database\Migrations;

```

```

use CodeIgniter\Database\Migration;

class Member extends Migration
{
    public function up()
    {
        $this->forge->addField([
            'id_member' => [
                'type'           => 'INT',
                'constraint'     => 5,
                'unsigned'       => true,
                'auto_increment' => true,
            ],
            'nama_member' => [
                'type'           => 'VARCHAR',
                'constraint'     => '100',
            ],
            'no_kartu' => [
                'type'           => 'VARCHAR',
                'constraint'     => '100',
            ],
            'kendaraan_member' => [
                'type'           => 'VARCHAR',
                'constraint'     => '100',
            ],
            'plat_member' => [
                'type'           => 'VARCHAR',
                'constraint'     => '100',
            ],
            'keterangan_plat' => [
                'type'           => 'VARCHAR',
                'constraint'     => '100',
            ],
        ],
        []);
        $this->forge->addKey('id_member', true);
        $this->forge->createTable('member');
    }

    public function down()
    {
        $this->forge->dropTable('member');
    }
}

```

```
}  
}
```

### Kode Sumber 5.12. Konfigurasi migration untuk Member

```
<?php  
  
namespace App\Database\Migrations;  
  
use CodeIgniter\Database\Migration;  
  
class Transaksi extends Migration  
{  
    public function up()  
    {  
        $this->forge->addField([  
            'id_transaksi' => [  
                'type'          => 'INT',  
                'constraint'    => 5,  
                'unsigned'      => true,  
                'auto_increment' => true,  
            ],  
            'nama_member' => [  
                'type'          => 'VARCHAR',  
                'constraint'    => '100',  
            ],  
            'no_kartu' => [  
                'type'          => 'VARCHAR',  
                'constraint'    => '100',  
            ],  
            'waktu_masuk' => [  
                'type' => 'DATETIME',  
                'null' => true,  
            ],  
            'waktu_keluar' => [  
                'type' => 'DATETIME',  
                'null' => true,  
            ],  
            'plat_member' => [  
                'type'          => 'VARCHAR',
```

```

        'constraint' => '100',
    ],
    'keterangan_plat' => [
        'type'      => 'VARCHAR',
        'constraint' => '100',
    ],
    'nama_pintuMasuk' => [
        'type'      => 'VARCHAR',
        'constraint' => '100',
    ],
    'nama_pintuKeluar' => [
        'type'      => 'VARCHAR',
        'constraint' => '100',
    ],
    ]);
$this->forge->addKey('id_transaksi', true);
$this->forge->createTable('transaksi');
}

public function down()
{
    $this->forge->dropTable('transaksi');
}
}

```

Kode Sumber 5.13. Konfigurasi migration untuk Transaksi

*[Halaman ini sengaja dikosongkan]*

## **BAB VI**

### **PENGUJIAN DAN EVALUASI**

Bab ini menjelaskan tahap uji coba terhadap Aplikasi Visitor Management System PT. PLN (Persero). Pengujian dilakukan untuk memastikan fungsionalitas dan kesesuaian hasil implementasi arsitektur dengan analisis dan perancangan arsitektur.

#### **6.1. Tujuan Pengujian**

Pengujian dilakukan terhadap Aplikasi Visitor Management System PT. PLN (Persero) guna menguji kemampuan arsitektur dalam melayani permintaan sistem aplikasi.

#### **6.2. Kriteria Pengujian**

Penilaian atas pencapaian tujuan pengujian didapatkan dengan memperhatikan beberapa hasil yang diharapkan berikut:

- a. Kemampuan arsitektur untuk melayani tampilan aplikasi.
- b. Kemampuan arsitektur untuk melayani query data dari aplikasi ke database.
- c. Kemampuan arsitektur untuk menyimpan data yang diinput melalui aplikasi.
- d. Kemampuan arsitektur untuk mengkategorikan kendaraan ganjil genap.

#### **6.3. Skenario Pengujian**

Skenario pengujian dilakukan dengan melakukan peran sebagai user yang akan menjalankan fitur-fitur. Langkah-langkah untuk setiap kebutuhan fungsionalitas yaitu sebagai berikut:

1. Staf administrasi dan petugas keamanan dapat membuka aplikasi VMS.
2. Staf administrasi dan petugas keamanan dapat daftar dan login.
3. Staf administrasi dan petugas keamanan dapat melakukan pengelolaan data.
4. Staf administrasi dan petugas keamanan dapat mengetahui kendaraan ganjil atau genap.

#### 6.4. Evaluasi Pengujian

Hasil pengujian dilakukan terhadap pengamatan mengenai perilaku sistem aplikasi VMS terhadap kasus skenario uji coba. Tabel 6.1 di bawah ini menjelaskan hasil uji coba terhadap aplikasi yang telah dibuat.

*Tabel 6.1. Hasil Evaluasi Pengujian*

<b>Kriteria Pengujian</b>	<b>Hasil Pengujian</b>
Server dapat melayani tampilan aplikasi	Terpenuhi
Server dapat menyediakan data yang diminta aplikasi dari database	Terpenuhi
Server dapat menyimpan data yang di-input user ke database	Terpenuhi

*[Halaman ini sengaja dikosongkan]*



## **BAB VII**

### **KESIMPULAN DAN SARAN**

#### **7.1. Kesimpulan**

Kesimpulan yang didapat setelah melakukan perancangan arsitektur sistem aplikasi VMS pada kegiatan kerja praktek di PT. PLN (Persero) Kantor Pusat adalah sebagai berikut :

- a. Arsitektur sistem yang dibangun telah sesuai dengan permintaan.
- b. Dengan adanya aplikasi VMS, PT. PLN (Persero) Kantor Pusat dapat dengan mudah memonitoring kendaraan ganjil genap yang masuk ke area kantor.

#### **7.2. Saran**

Saran untuk perancangan arsitektur sistem aplikasi VMS adalah sebagai berikut :

- a. Implementasikan sistem pemantauan dan pemeliharaan dalam arsitektur aplikasi. Berikan fasilitas untuk memonitor kinerja, mencatat aktivitas, dan melakukan pemeliharaan rutin guna memastikan kinerja yang optimal dari sistem VMS.

*[Halaman ini sengaja dikosongkan]*

## DAFTAR PUSTAKA

- [1] Dicoding Blog. 2019. Berkenalan Dengan Pemrograman Web - Dicoding Blog. [ONLINE] Available at: <https://www.dicoding.com/blog/kenal-pemrograman-web/>. [Diakses April 2024].
- [2] Coding Studio. 2023. Apa itu CodeIgniter? Pengertian dan Cara Kerjanya - Coding Studio. [ONLINE] Available at: <https://codingstudio.id/blog/codeigniter-adalah/>. [Diakses April 2024].
- [3] Hostinger Tutorial. 2023. Apa Itu MySQL? Pengertian MySQL, Cara Kerja, dan Kelebihannya – Hostinger Tutorial. [ONLINE] Available at: <https://www.hostinger.co.id/tutorial/apa-itu-mysql>. [Diakses April 2024].
- [4] Niagahoster Blog. 2021. MVC: Pengertian Dasar, Manfaat, dan Contohnya dalam Pengembangan Website - Niagahoster Blog. [ONLINE] Available at: <https://www.niagahoster.co.id/blog/mvc-adalah/>. [Diakses April 2024].

*[Halaman ini sengaja dikosongkan]*

## **BIODATA PENULIS I**

Nama : Ravin Pradhitya  
Tempat, Tanggal Lahir : Jakarta, 23 September 2002  
Jenis Kelamin : Laki-laki  
Telepon : +628129090876  
Email : ravinpradhitya@gmail.com

### **AKADEMIS**

Kuliah : Departemen Teknik Informatika –  
FTEIC , ITS  
Angkatan : 2020  
Semester : 8 (Delapan)