

Desain Pengembangan Website dengan Arsitektur Model View Controller pada Framework Laravel

Levianita Rahmawati^a, Sumarsono^b

^aInformatika, Sains dan Teknologi, UIN Sunan Kalijaga Yogyakarta, 21106050037@student.uin-suka.ac.id

^bInformatika, Sains dan Teknologi, UIN Sunan Kalijaga Yogyakarta, sumarsono@uin-suka.ac.id

Submitted: 07-07-2024, Reviewed: 19-07-2024, Accepted 07-10-2024

<https://doi.org/10.47233/jteksis.v6i4.1497>

Abstract

The development of technology has changed the face of information media from static to dynamic. The internet is the main catalyst in this change, making access to information faster and wider. One of the dynamic information media that undergoes a transformation process and facilitates user responsibility is a website developed using a framework. However, in the implementation of the framework, it needs to be supported by an architecture that is able to control the developing program code. The purpose of this study is to analyze the use of the Laravel framework with the support of the Model View Controller (MVC) architecture. The data collection method for this study uses the Observation Technique by researchers practicing directly on the Laravel framework system. The results of this study indicate that the use of Laravel with the Model View Controller architecture can help organize program code in a more structured manner to regulate efficiency in dynamic website development and facilitate continuous program code maintenance.

Keywords: *programming code, website, model view controller, laravel*

Abstrak

Perkembangan teknologi telah mengubah wajah media informasi dari statis menjadi dinamis. Internet menjadi katalisator utama dalam perubahan ini sehingga menjadikan akses terhadap informasi lebih cepat dan luas. Salah satu media informasi dinamis yang mengalami proses transformasi dan memfasilitasi responsibilitas terhadap pengguna adalah *website* yang dikembangkan menggunakan *framework*. Namun dalam implementasi *framework* perlu didukung oleh arsitektur yang mampu mengendalikan kode program yang berkembang. Tujuan penelitian ini adalah menganalisis penggunaan *framework laravel* dengan dukungan arsitektur *Model View Controller (MVC)*. Metode pengumpulan data penelitian ini menggunakan Teknik observasi dengan cara peneliti melakukan praktek langsung terhadap sistem *framework Laravel*. Hasil penelitian ini menunjukkan penggunaan *Laravel* dengan arsitektur *Model View Controller* dapat membantu dalam pengorganisasian kode program secara lebih terstruktur untuk mengatur efisiensi dalam pengembangan *website* dinamis serta mempermudah pemeliharaan kode program secara berkesinambungan.

Keywords: *kode pemrograman, website, model view controller, laravel*

This work is licensed under Creative Commons Attribution License 4.0 CC-BY International license



PENDAHULUAN

Pertumbuhan teknologi yang signifikan saat ini mempengaruhi aspek kehidupan manusia. Kemajuan perkembangan teknologi meningkatkan permintaan akses informasi yang cepat dan efisien. Dahulu, informasi dan berita hanya bisa diakses melalui radio, buku, dan televisi yang sudah kuno. Namun saat ini kita bisa mengaksesnya melalui internet. Melalui *website* di perangkat *mobile* atau komputer yang terkoneksi dengan internet, berbagai informasi bisa didapat dengan cepat, akurat, dan relevan [1]. Internet menyediakan berbagai *platform* untuk berbagi informasi, seperti *website*, media sosial, portal berita *online*, dan lain-lain[2].

Website merupakan wadah untuk menyampaikan informasi dengan memanfaatkan konsep *hyperlink* baik secara statis maupun dinamis[3]. Saat ini, *website* bukan hanya digunakan sebagai media informasi yang dibaca atau statis, melainkan juga berperan sebagai media yang lebih

hidup dan interaktif. Data pada *website* disimpan di server yang sama dan berisi berbagai jenis data yang dapat diakses baik individu, kelompok, maupun organisasi [4]. Perancangan *website* berbasis ui/ux telah mengalami banyak perkembangan sebagai salah satu sarana komunikasi yang paling populer di masyarakat. Perkembangan perancangan *website* mencakup penggunaan berbagai *framework* atau kerangka kerja untuk memudahkan pengembangan dan pemeliharaan *website* [5]

Framework merupakan sebuah kerangka kerja yang dibuat untuk mempermudah proses perancangan sebuah *website*. Dalam konteks *website*, *framework* menyediakan bermacam fungsi, sintaksis, *library*, *extension*, dan *template* siap pakai untuk mempercepat pembangunan sistem [6]. Komponen dan variabel *framework* berguna untuk mempermudah pembacaan kode, perencanaan, pengujian, dan pemeliharaan yang dibutuhkan saat proses perancangan *website* [7]. Perkembangan

framework dimulai pada tahun 2004 yaitu dengan dipublikasikan secara umum *framework Prado 1*, lalu satu tahun kemudian, tahun 2005, muncul tiga *framework* terkenal di dunia, yaitu *CakePHP 1*, *Symfony 1*, dan *Prado 2*, seiring berkembangnya teknologi saat ini, tercatat sudah ada lebih dari 30 *framework* terkenal di dunia dengan berbagai variasi fungsi khusus[8]. Demikian juga pada perkembangan *framework* bahasa pemrograman PHP (*Hypertext Preprocessor*), ada berbagai jenis *framework* PHP yang populer, seperti *CodeIgniter*, *Symfony*, *Yii*, dan *Laravel*, yang semuanya menggunakan arsitektur *Model View Controller* (MVC)[9]. Setiap *framework* memiliki keunggulan dan fitur spesifik yang dapat disesuaikan dengan kebutuhan proyek.

Kerangka kerja atau *framework* PHP yang paling populer dan banyak digunakan dalam pengembangan *website* saat ini adalah *Laravel* yang dirancang dengan kerangka sistematis menggunakan arsitektur *Model View Controller* (MVC)[10]. *Laravel* menawarkan berbagai fitur yang mempermudah pengembangan aplikasi *website*, seperti *routing*, autentikasi, manajemen *database*, dan dukungan untuk migrasi *database*. Selain itu, *Laravel* dikenal karena dokumentasinya yang komprehensif dan komunitas pengembang yang aktif. Setelah mengalami banyak perbaikan dan pengembangan, *Laravel* mencapai kestabilan dengan dirilisnya versi *Laravel 5.2.36* pada 6 Juni 2016 yang mendukung sistem operasi lintas platform [11]. Dengan berbagai keunggulan tersebut, *Laravel* telah menjadi salah satu *framework* PHP yang paling terkenal dan sering digunakan oleh pengembang web di seluruh dunia.

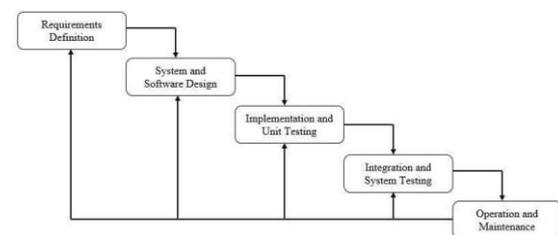
Keunggulan *Laravel* tidak lepas dari penggunaan arsitektur *Model-View-Controller* atau MVC yang menjadi inti dari *framework* ini. MVC adalah pola desain arsitektur yang memisahkan aplikasi menjadi tiga bagian utama yaitu *Model*, *View*, dan *Controller*. *Model* bertanggung jawab untuk logika data dan pengelolaan *database*[12], [13], *View* mengurus tampilan antarmuka pengguna, dan *Controller* mengatur alur data antara *Model* dan *View* serta menangani interaksi pengguna [14]. Penggunaan arsitektur MVC pada *framework Laravel* mempermudah pengembangan, perawatan, dan pengujian aplikasi web, sehingga memungkinkan pengembang untuk fokus pada masing-masing komponen tanpa mengganggu yang lain [15].

Pengembangan *website* dengan menggunakan arsitektur *Model View Controller* (MVC) pada *framework Laravel* memberikan pendekatan yang terstruktur dan efisien dalam mengorganisir kode. MVC membagi aplikasi menjadi tiga komponen utama: *Model* untuk mengelola data dan logika

bisnis, *View* untuk menangani tampilan dan antarmuka pengguna, serta *Controller* sebagai pengatur interaksi antara *Model* dan *View*. Dengan menggunakan *Laravel*, pengembang dapat dengan mudah mengimplementasikan logika aplikasi dalam *Model*, mengatur tampilan dengan *View*, dan mengatur alur aplikasi melalui *Controller*. Hal ini tidak hanya meningkatkan kualitas pengembangan, tetapi juga memudahkan pemeliharaan dan skalabilitas *website* dalam jangka panjang.

METODE PENELITIAN

Dalam artikel ini, peneliti menggunakan metode *System Development Life Cycle* (SDLC) menggunakan model *Waterfall* untuk mengimplementasikan arsitektur *Model View Controller* (MVC) pada *framework Laravel*. Metode SDLC *Waterfall* adalah pendekatan linier dan proses yang terstruktur dalam pengembangan perangkat lunak, yang meliputi tahapan-tahapan yang dilakukan secara berurutan [16].



Gambar 1. Tahap Metode Waterfall

Tahapan-tahapan tersebut meliputi analisis kebutuhan (*requirements definition*), perancangan desain sistem (*system and software design*), implementasi (*implementation and unit testing*), pengujian (*integration and system testing*), dan pemeliharaan (*operation and maintenance*).

2.1. Analisis kebutuhan

Tahap ini melibatkan pengumpulan informasi mengenai kebutuhan pengguna dan spesifikasi sistem yang akan dibangun. Metode yang digunakan yaitu metode kualitatif deskriptif dengan menggunakan cara kajian literatur terkait pengembangan *website* menggunakan metode MVC. Tujuannya adalah untuk memahami fitur-fitur utama yang harus ada dalam *website* dan bagaimana pengguna berinteraksi dengan sistem tersebut [17].

2.2. Perancangan desain sistem

Pada tahap ini, sistem *website* dirancang menggunakan bahasa pemodelan seperti *Unified Modeling Language* (UML), *Business Process Model and Notation* (BPMN), *flowchart*, dan lainnya. Bahasa pemodelan ini telah menjadi standar

dalam industri untuk membantu dalam memahami, merancang, dan mendokumentasikan semua aspek sistem secara menyeluruh [18].

2.3. Implementasi

Tahap implementasi melibatkan penerapan desain ke dalam kode program menggunakan bahasa pemrograman. Peneliti menggunakan bahasa pemrograman PHP dengan *framework Laravel* dengan jenis Pemrograman Berbasis Objek atau *Object Oriented Programming* (OOP).

Framework Laravel mengimplementasi komponen *Model, View, dan Controller* dalam struktur direktori *Laravel*. Dengan struktur direktori yang sistematis, pengembangan *website* lebih terorganisir sehingga memudahkan pemeliharaan dan pengembangan fitur baru di masa depan [19]. Implementasi MVC menggunakan *Laravel* memungkinkan pembagian tugas yang jelas antara data, tampilan, dan logika aplikasi, yang pada akhirnya meningkatkan modularitas dan skalabilitas sistem.

2.4. Pengujian sistem

Pengujian merupakan tahapan penting dalam pengembangan suatu sistem untuk memastikan bahwa sistem berfungsi dengan baik sesuai dengan spesifikasi yang telah ditentukan. Pengujian perangkat lunak dibedakan menjadi dua jenis, yaitu *white box testing* dan *black box testing*. *White box testing* adalah metode pengujian yang melibatkan pemeriksaan dan analisis kode program dalam modul untuk mendeteksi kesalahan. Sebaliknya, *black box testing* adalah metode pengujian yang berfokus pada aspek eksternal aplikasi, seperti tampilan, fungsi-fungsi yang tersedia, dan kesesuaian alur fungsi dengan sistem kerja [20]. Pengembangan *website* ini menggunakan *black box testing* dengan beberapa pengujian, yaitu:

- Pengujian unit (*unit testing*), digunakan untuk menguji setiap unit atau komponen dari aplikasi secara terpisah untuk memastikan bahwa setiap bagian berfungsi dengan benar dengan *PHPUnit*.
- Pengujian fungsional (*functional testing*), digunakan untuk menguji berbagai fungsi atau fitur dengan menggunakan berbagai skenario atau *test case*.
- Usability test*, digunakan untuk mengevaluasi seberapa mudah pengguna berinteraksi dengan antarmuka pengguna (UI) dan pengalaman pengguna (UX) sistem.

Pengujian yang komprehensif ini memastikan bahwa sistem berfungsi dengan baik dan siap memberikan pengalaman terbaik kepada pengguna.

2.5. Pemeliharaan sistem

Proses pemeliharaan berfokus untuk menjaga perangkat lunak agar tetap berfungsi dengan baik setelah diimplementasikan. Tahap ini memastikan bahwa semua bagian perangkat lunak telah diuji dengan benar dan berfungsi sesuai dengan yang diharapkan. Tujuan dari tahap ini adalah untuk meminimalkan kesalahan (*error*) yang mungkin terdapat pada program dan memastikan bahwa hasil keluaran sesuai dengan yang diinginkan [21]. Proses pemeliharaan meliputi perbaikan *bug*, penyesuaian untuk menghadapi perubahan lingkungan, dan peningkatan sistem berdasarkan umpan balik pengguna.

HASIL DAN PEMBAHASAN

Prinsip Kerja Framework Laravel

Pengembangan sistem *website* bisa saja hanya menggunakan bahasa pemrograman PHP murni tanpa *framework* tambahan. Namun, penggunaan *framework* dalam pengembangan *website* memiliki banyak keuntungan, yaitu memudahkan pengelolaan kode, meningkatkan efisiensi, dan mempercepat proses pengembangan. *Laravel*, sebagai salah satu *framework* PHP yang paling populer, menawarkan berbagai fitur bawaan yang sangat membantu dalam pengembangan *website*. Fitur-fitur ini meliputi migrasi *database*, sistem *routing* yang fleksibel, serta alat bantu untuk pengujian dan *debugging*[22].

Fitur migrasi *database* memungkinkan pengembang untuk membuat perubahan pada struktur *database* dengan mudah dan terorganisir. Migrasi ini mencakup penambahan atau penghapusan tabel dan kolom, yang semuanya dikelola melalui perintah *artisan Laravel*. Contoh penggunaan migrasi *database* di *Laravel* dapat dilihat pada direktori *Database/migrations*, di mana semua file migrasi disimpan.

Selain itu, *Laravel* juga memiliki sistem *routing* yang fleksibel, yang mengatur bagaimana *URL* aplikasi merespons permintaan pengguna. Sistem *routing* didefinisikan dalam file *routes/web*. PHP dan memungkinkan pengembang untuk menetapkan rute yang mengarah ke *controller* yang sesuai.

```
Route::get('/about', function () {  
    return view('about');  
});
```

Gambar 2. Route Laravel

Secara umum, *route* menentukan *URI* (*Uniform Resource Identifier*) dan menentukan

fungsi yang harus dijalankan ketika URI tersebut diakses. *Route* dapat digunakan untuk menangani permintaan *GET*, *POST*, *PUT*, *PATCH*, *DELETE*, dan metode *HTTP* lainnya.

Pengujian dan *debugging* di *Laravel* juga difasilitasi dengan baik. *Laravel* menyediakan alat bantu seperti *PHPUnit* untuk pengujian otomatis dan perintah *artisan* untuk *debugging*. Direktori *Test* menyimpan semua file pengujian yang dijalankan melalui *PHPUnit*, memungkinkan pengembang untuk memastikan bahwa aplikasi berfungsi dengan baik sebelum di-*deploy* ke lingkungan produksi[23].

Struktur direktori yang ada pada *framework Laravel* adalah sebagai berikut:

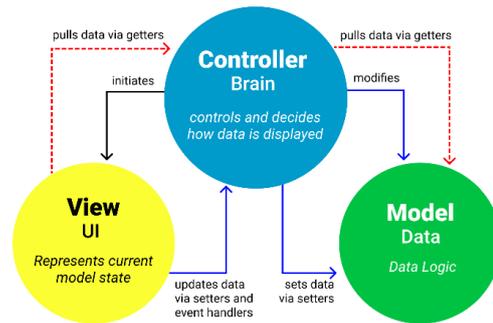
- 1) *App/Http*: Direktori ini menyimpan semua file request dan response *HTTP*,
- 2) *Database/migrations*: Direktori ini berisi file untuk membuat perubahan pada *database*, seperti menambah atau menghapus tabel dan kolom,
- 3) *Database/seeds*: Direktori ini berisi file *database seeds* yang dihasilkan oleh *Laravel* dengan perintah "*PHP artisan make:seeder*" dan digunakan untuk menginisialisasi data dalam tabel,
- 4) *Public*: Direktori ini menyimpan resource aplikasi yang dapat diakses melalui browser, seperti gambar, *JavaScript*, dan *CSS*,
- 5) *Resources*: Direktori ini memiliki sub-direktori *assets*, *lang*, dan *views*, yang berfungsi untuk mengkompilasi file, mendefinisikan aplikasi multi bahasa, dan menyimpan semua template file *HTML*,
- 6) *Test*: Direktori ini menyimpan file test yang dijalankan melalui *PHPUnit*.

Dengan kerangka dan fitur-fitur yang sistematis ini, *Laravel* memudahkan pengembangan *website* yang lebih terstruktur dan efisien, memungkinkan pengembang untuk fokus pada pembuatan fitur dan fungsionalitas yang kompleks tanpa harus khawatir tentang pengelolaan kode yang rumit.

Arsitektur Model View Controller pada Laravel

Arsitektur *Model-View-Controller (MVC)* pada *Laravel* memberikan kerangka yang terstruktur dan terorganisir untuk pengembangan *website*.

MVC Architecture Pattern



Gambar 3. Struktur Arsitektur MVC

Dengan memisahkan logika aplikasi menjadi tiga komponen utama: *Model*, *View*, dan *Controller*, *MVC* membantu dalam pemeliharaan dan pengembangan aplikasi yang lebih efisien. Berikut adalah implementasi *MVC* pada struktur folder *Laravel*:

- 1) *Model*: *Model* dalam *Laravel* disimpan di dalam direktori *App/Models*, bertanggung jawab untuk berinteraksi dengan *database*, mengelola data, dan mengimplementasikan logika bisnis yang terkait dengan data tersebut.

```

1 <?php
2 namespace App\Models;
3 use Illuminate\Database\Eloquent\Factories\HasFactory;
4 use Illuminate\Foundation\Auth\User as Authenticatable;
5 use Illuminate\Notifications\Notifiable;
6 use Laravel\Sanctum\HasApiTokens;
7
8 class User extends Authenticatable
9 {
10     use HasApiTokens, HasFactory, Notifiable;
11     protected $guarded = ['id'];
12
13     protected $hidden = [
14         'password',
15         'remember_token',
16     ];
17
18     protected $casts = [
19         'email_verified_at' => 'datetime',
20     ];
    
```

Gambar 4. Models pada Laravel

Gambar di atas merupakan file *app/Models/User.php* untuk mendefinisikan *model 'user'*. *Model* ini digunakan untuk mewakili data pengguna dan berinteraksi dengan tabel *'users'* dalam *database*. *Model* ini mendefinisikan kelas *'user'* yang mewarisi kelas *Authenticatable*. Memiliki atribut *'id'*, *'password'*, *'remember_token'*, *'email_verified_at'* serta menggunakan beberapa sifat untuk menambahkan fungsionalitas, yaitu:

- a. *HasApiTokens*, untuk mendukung penggunaan *token API*.
- b. *HasFactory*, untuk mendukung pembuatan instance *Model* menggunakan *factory*.

- c. *Notifiable*, untuk mendukung pengiriman notifikasi.
- 2) *View*: *View* disimpan di dalam direktori *resources/views*, berfungsi untuk menampilkan antarmuka pengguna dan menggunakan *Blade templating engine* untuk mempermudah pembuatan dan pengelolaan tampilan *HTML*.

```

1 <html>
2 <head>
3   <title>Welcome</title>
4 </head>
5 <body>
6   <h1>Welcome to My Application</h1>
7 </body>
8 </html>

```

Gambar 5. Views pada Laravel

Blade templating engine memungkinkan penggunaan fitur seperti pengulangan dan kondisi dalam template, yang memudahkan dalam pembuatan tampilan yang dinamis dan interaktif.

- 3) *Controller*: *Controller* disimpan di dalam direktori *App/Http/Controllers*, bertanggung jawab untuk menerima input dari pengguna, mengolah data melalui *Model*, dan mengembalikan hasilnya ke *View*. Secara umum, *Controller* berisi serangkaian method yang disebut "*actions*" atau "*handlers*", yang menangani berbagai jenis permintaan *HTTP* seperti *GET*, *POST*, *PUT*, *DELETE*, dan lainnya. Ketika sebuah permintaan diterima, *router* akan menentukan *Controller* mana yang harus menangani permintaan tersebut, dan kemudian method yang sesuai dalam *Controller* tersebut akan dieksekusi.

```

3 namespace App\Http\Controllers;
4
5 use App\Models\User;
6 use Illuminate\Http\Request;
7
8 class UserController extends Controller
9 {
10     public function index()
11     {
12         $users = User::all();
13         return view('users.index', ['users' => $users]);
14     }
15
16     public function show($id)
17     {
18         $user = User::find($id);
19         return view('users.show', ['user' => $user]);
20     }
21 }

```

Gambar 6. Controllers pada Laravel

Gambar di atas merupakan bagian dari *Controller UserController*. *Controller* ini mengimpor *Model 'user'* agar dapat digunakan dan berinteraksi dengan data pengguna dalam *database*.

UserController memiliki method '*index*' dan '*show*'.

SIMPULAN

Berdasarkan analisis pengembangan *website* dengan menggunakan arsitektur *Model View Controller (MVC)* pada *framework Laravel*, dapat disimpulkan bahwa penggunaan *Laravel* dalam memisahkan logika aplikasi menjadi *Model*, tampilan, dan pengontrol memberikan keuntungan signifikan dalam pengembangan *website*. *Model* bertanggung jawab untuk interaksi dengan *database* dan logika bisnis, tampilan menampilkan antarmuka pengguna dengan bantuan *Blade templating engine*, sementara pengontrol menerima input pengguna, mengolah data melalui *Model*, dan mengirimkan hasilnya ke tampilan. Dengan struktur direktori yang terorganisir dan fitur bawaan yang mendukung manajemen *database*, *routing*, serta pengujian, *Laravel* memberikan dukungan lengkap untuk proses pengembangan *website*. Penelitian masa depan dari arsitektur *MVC* membangun *website* yang efisien, terstruktur, dan mudah dipelihara adalah terkait optimasi kinerja, pengelolaan dependensi, *security*, dan desain *User Experience*.

UCAPAN TERIMAKASIH

Penulis memberikan ucapan terima kasih kepada program Studi Informatika UIN Sunan Kalijaga Yogyakarta yang telah membantu dan memfasilitasi dalam pelaksanaan penelitian ini.

DAFTAR PUSTAKA

- [1] M. Santoso, "Implementation Of UI/UX Concepts And Techniques In Web Layout Design With Figma," *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 6, no. 2 SE-Articles, Apr. 2024.
- [2] J. Ren, H. Dong, A. Popovic, G. Sabnis, and J. Nickerson, "Digital platforms in the news industry: how social media platforms impact traditional media news viewership," *Eur. J. Inf. Syst.*, vol. 33, no. 1, pp. 1–18, 2024.
- [3] O. W. Purbo, "A systematic analysis: Website development using Codeigniter and Laravel framework," *Enrich. J. Manag.*, vol. 12, no. 1, pp. 1008–1014, 2021.
- [4] E. A. Nugroho, S. Sumarsono, and E. H. Gunawan, "Framework of the employee attendance system with QR code in the pandemic Covid-19," in *2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM)*, 2021, pp. 503–506.
- [5] V. Dikcius, S. Urbonavicius, K. Adomaviciute, M. Degutis, and I. Zimaitis, "Learning Marketing Online: The Role of Social Interactions and Gamification Rewards," *J. Mark. Educ.*, 2020.
- [6] Sapardi, W. Hadikristanto, and N. T. Kurniadi, "Implementasi Pengembangan Aplikasi Sistem Manajemen Aset Berbasis Web Menggunakan Metode Waterfall Untuk Mengoptimalkan Penggunaan Aset Pada PT. Hutama Karya (Persero)," *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 5, no. 4 SE-Articles, Oct. 2023.
- [7] R. R. Prasena, "Studi Komparasi Pengembangan

- Website Dengan Framework Codeigniter Dan Laravel,” in *Conference on Business, Social Sciences and Innovation Technology*, 2020, vol. 1, no. 1, pp. 613–621.
- [8] F. Shi *et al.*, “A new technology perspective of the Metaverse: Its essence, framework and challenges,” *Digit. Commun. Networks*, 2023.
- [9] R. R. Putri, “Comparison Study of PHP Frameworks in Web Application Development: Laravel, CodeIgniter, and Symfony,” *J. Ilmu Komput.*, vol. 2, no. 4, 2023.
- [10] R. N. Thakur and U. S. Pandey, “The role of model-view controller in object oriented software development,” *Nepal J. Multidiscip. Res.*, vol. 2, no. 2, pp. 1–6, 2019.
- [11] I. G. S. Rahayuda, “Implementasi Teknologi Informasi Untuk Mengembangkan E-Government Menggunakan Framework Laravel,” *Semnasteknomedia Online*, vol. 5, no. 1, pp. 2–4, 2017.
- [12] R. F. Al Fikri and E. Hernawati, “Penerapan prinsip model view controller pada sistem informasi dosen berbasis website,” *Rabit J. Teknol. dan Sist. Inf. Univrab*, vol. 9, no. 1, pp. 69–78, 2024.
- [13] P. T. I. Permana and A. N. Sihanato, “Implementasi Arsitektur MVC Dalam Pengembangan Aplikasi Customer Relationship Portal,” *J. Teknol. Inf.*, vol. 10, no. 1, pp. 50–57, 2024.
- [14] O. Barzilai and R. Gafni, “Using Web Frameworks in Server Side Programming Courses,” *J. Comput. Inf. Syst.*, vol. 63, no. 4, pp. 866–876, 2023.
- [15] M. Stauffer, *Laravel: Up & Running*. “O’Reilly Media, Inc.,” 2023.
- [16] E. Alfonsius and W. W. Kalengkongan, “Development of an Alumni Data Processing Information System Using the SDLC Modeling System Development Method,” *J. Ilm. Sist. Inf. Akunt.*, vol. 3, no. 1, pp. 53–59, 2023.
- [17] A. R. Basar, N. H. Adi, and S. Zebua, “Sistem Informasi Pemesanan dan Pelayanan Katering Menggunakan Metode Model View Controller Berbasis Web,” *JR J. Responsive Tek. Inform.*, vol. 3, no. 02, pp. 43–57, 2019.
- [18] S. Sumarsono, D. Saputro, and A. F. Rifai, “Pemodelan Proses Bisnis Kuliah Online MOOCs menggunakan BPMN (Studi Kasus alison. com),” *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 8, no. 3, pp. 199–209, 2023.
- [19] S. Anwar, “Rancangan Aplikasi Penjualan Perlengkapan Dan Mainan Anak Menggunakan Konsep Model View Controller (Mvc),” *J. Ris. Sist. Inf. dan Teknol. Inf.*, vol. 2, no. 1, pp. 33–45, 2020.
- [20] M. N. Ichsanudin, M. Yusuf, and S. Suraya, “Penguujian Fungsional Perangkat Lunak Sistem Informasi Perpustakaan Dengan Metode Black Box Testing Bagi Pemula,” *STORAGE J. Ilm. Tek. dan Ilmu Komput.*, vol. 1, no. 2, pp. 1–8, 2022.
- [21] A. Almogahed *et al.*, “A refactoring classification framework for efficient software maintenance,” *IEEE Access*, 2023.
- [22] susanto susanto, basworo pramono, A. Widyandayani, and P. Patmawati, “Implementasi Metode RAD pada Sistem Pengaduan Masyarakat (SIPMAS) di Desa Logung Menggunakan Framework Laravel,” *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 6, no. 3 SE-Articles, Jul. 2024.
- [23] N. T. Marli’aini and D. A. Anggoro, “Sistem Informasi Persediaan Barang Pada PT. TGA Berbasis Website Menggunakan Framework Laravel,” *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 6, no. 3 SE-Articles, Jul. 2024.