

# Penerapan Arsitektur MVC pada Website Pengumpul Tugas Menggunakan PHP

Irsyad Fadhil Makarim<sup>1</sup>, Ayodya Fade Edfira<sup>2</sup>, Aileen Anindya Ramadhani<sup>3</sup>, Michael Christiano<sup>4</sup>,

Wisanggeni Atthoriq Kuswirasatya<sup>5</sup>, Fawwaz Ali Akbar<sup>6\*</sup>

<sup>1, 2, 3, 4, 5, 6</sup> Afiliasi (Informatika, UPN “Veteran” Jawa Timur)

<sup>1</sup> [22081010131@student.upnjatim.ac.id](mailto:22081010131@student.upnjatim.ac.id)

<sup>2</sup> [22081010120@student.upnjatim.ac.id](mailto:22081010120@student.upnjatim.ac.id)

<sup>3</sup> [22081010121@student.upnjatim.ac.id](mailto:22081010121@student.upnjatim.ac.id)

<sup>4</sup> [22081010123@student.upnjatim.ac.id](mailto:22081010123@student.upnjatim.ac.id)

<sup>5</sup> [22081010127@student.upnjatim.ac.id](mailto:22081010127@student.upnjatim.ac.id)

<sup>6</sup> [fawwaz.ali.fik@upnjatim.ac.id](mailto:fawwaz.ali.fik@upnjatim.ac.id)

\*Corresponding author email: [fawwaz.ali.fik@upnjatim.ac.id](mailto:fawwaz.ali.fik@upnjatim.ac.id)

**Abstrak**— Dalam era digital, sistem manajemen tugas yang efektif sangat dibutuhkan, terutama di bidang pendidikan dan profesional. Tujuan dari penelitian ini adalah untuk menerapkan arsitektur Model-View-Controller (MVC) pada website pengumpul tugas yang menggunakan PHP untuk meningkatkan struktur dan efisiensi pengelolaan tugas online. Komponen utama aplikasi MVC adalah Model, View, dan Controller. Model, View, dan Controller masing-masing menangani manajemen data, presentasi, dan logika kontrol. Implementasi dilakukan menggunakan PHP, dengan framework Bootstrap dan jQuery untuk antarmuka, dan MySQL sebagai basis data. Studi ini menggunakan GitHub untuk manajemen kode dan kolaborasi tim, dan menggunakan model pengembangan Waterfall. Hasil menunjukkan bahwa pemisahan kode melalui MVC memungkinkan pengembangan paralel yang lebih efisien dan membuat pengembangan dan pemeliharaan aplikasi lebih mudah. Fitur utama yang berhasil diimplementasikan meliputi pengelolaan daftar tugas, pembuatan dan pengunggahan tugas, serta tampilan status tugas. Pengujian menunjukkan bahwa semua fitur bekerja sesuai dengan spesifikasi yang diharapkan, memberikan pengalaman pengguna yang baik. Kesimpulannya, penerapan MVC dalam pengembangan website pengumpul tugas memberikan keuntungan signifikan dalam hal struktur kode, skalabilitas, dan efisiensi pengembangan, serta memenuhi kebutuhan pengguna dalam mengelola tugas secara efektif dan terorganisir.

**Kata Kunci**— Arsitektur Model-View-Controller (MVC), Website, PHP, Tugas.

## I. PENDAHULUAN

Dalam era digital yang terus berkembang, kebutuhan akan sistem manajemen tugas yang efisien dan terstruktur semakin meningkat, terutama dalam lingkungan pendidikan dan profesional. Website pengumpul tugas menjadi salah satu solusi yang efektif untuk memfasilitasi proses pengumpulan, penilaian, dan pengelolaan tugas-tugas secara online. Untuk membangun aplikasi web yang kuat, terstruktur, dan mudah dikelola, penerapan arsitektur Model-View-Controller (MVC) menjadi sangat relevan.

Arsitektur MVC adalah sebuah pola desain perangkat lunak yang membagi aplikasi menjadi tiga komponen utama: Model, View, dan Controller. Model menangani manajemen data dan logika bisnis, sementara View menangani presentasi dan antarmuka pengguna. Controller menghubungkan Model dan View, mengatur aliran data, dan menangani input pengguna. Aplikasi web menjadi lebih mudah dibuat, diuji, dan dipertahankan berkat pembagian yang jelas ini.

PHP, sebagai salah satu bahasa pemrograman web yang populer dan fleksibel, mendukung implementasi arsitektur MVC dengan baik. PHP memungkinkan pengembang untuk membangun aplikasi web dinamis dengan kemudahan dalam pengelolaan server-side scripting. Framework PHP seperti Laravel, CodeIgniter, dan Symfony sudah secara default menggunakan arsitektur MVC, tetapi penerapan MVC secara manual juga dapat dilakukan untuk memberikan pemahaman yang lebih mendalam mengenai struktur dan alur kerja aplikasi web.

Tujuan dari penelitian ini adalah untuk menerapkan arsitektur MVC pada sebuah website pengumpul tugas yang menggunakan PHP. Diharapkan dengan menerapkan arsitektur MVC, website ini akan menjadi lebih mudah bagi pengguna untuk mengelola tugas, meningkatkan efisiensi proses pengumpulan tugas, dan memastikan bahwa sistem dapat diperluas dan dirawat dengan mudah di masa mendatang. Selain itu, metode ini dirancang untuk meningkatkan konsistensi dan kualitas kode, memudahkan kerja tim pengembang, dan memberikan pengalaman pengguna yang lebih baik melalui antarmuka yang responsif dan mudah dipahami.

## II. LANDASAN TEORI

**2.1 Model-View-Controller (MVC) Architecture:** Ini adalah konsep desain perangkat lunak yang membagi aplikasi menjadi tiga komponen utama: Model, View, dan Controller.

**Model:** Merepresentasikan struktur data aplikasi dan aturan bisnis.

**View:** Bertanggung jawab untuk menampilkan informasi kepada pengguna.

**Controller:** Mengatur interaksi pengguna dengan model dan view.

**2.2 PHP sebagai Bahasa Pemrograman:** PHP digunakan sebagai bahasa utama dalam pengembangan website. PHP biasanya digunakan untuk mengelola logika backend, mengakses basis data, dan merender halaman web.

**2.3 Keunggulan Arsitektur MVC:**

**Pemisahan Kode:** MVC memungkinkan pemisahan yang jelas antara tampilan, logika aplikasi, dan struktur data. Hal ini memfasilitasi pengembangan, pemeliharaan, dan pengujian aplikasi yang lebih baik.

**Skalabilitas:** Dengan MVC, kode dapat dikelompokkan berdasarkan fungsionalitasnya, memudahkan dalam mengelola aplikasi yang besar dan kompleks.

**Pengembangan Paralel:** Tim pengembang dapat bekerja secara paralel pada berbagai komponen MVC tanpa mengganggu bagian lainnya.

**2.4 Penerapan MVC pada Website Pengumpul Tugas:**

**Model:** Berisi aturan bisnis terkait pengumpulan tugas, termasuk validasi data dan interaksi dengan basis data untuk menyimpan dan mengambil informasi.

**View:** Mengatur tampilan antarmuka pengguna, termasuk halaman untuk melihat daftar tugas, mengunggah file, dan menampilkan status pengumpulan.

**Controller:** Menangani logika aplikasi, seperti menerima input dari pengguna, memproses data, dan mengarahkan aplikasi ke view yang sesuai.

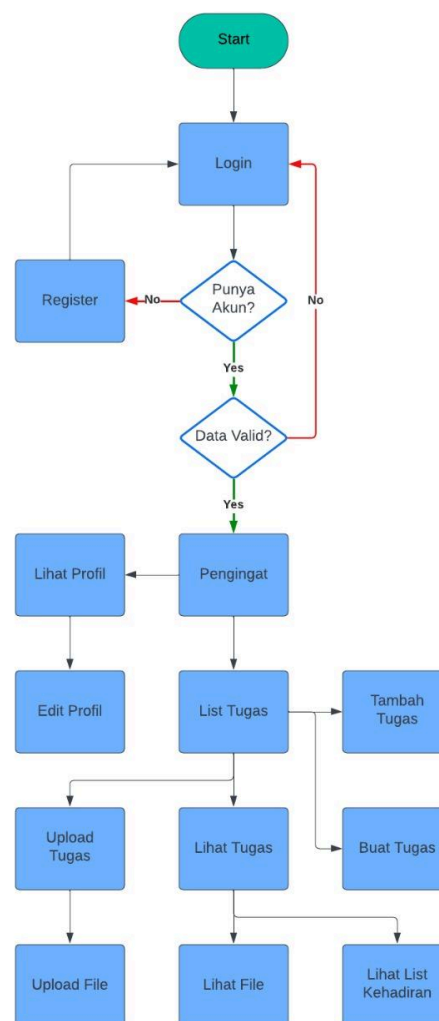
**2.5 Implementasi PHP:** PHP digunakan untuk mengimplementasikan setiap bagian dari MVC. Contoh-contoh kode PHP akan menunjukkan bagaimana setiap komponen MVC diimplementasikan dalam konteks pengumpulan tugas.

**3.1 Model Pengembangan Perangkat Lunak Model pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah model Waterfall, namun hanya sampai tahap pengujian (testing) dan tidak mencakup tahap pemeliharaan (maintenance). Pemilihan model Waterfall didasarkan pada kesesuaiannya untuk proyek website dengan persyaratan yang jelas dan tidak terlalu kompleks, sehingga cocok untuk menggunakan pendekatan linier dan terstruktur.**

**3.2 Alat dan Teknologi Pengembangan website ini menggunakan bahasa pemrograman PHP, HTML, CSS, dan JavaScript. Framework yang digunakan adalah Bootstrap untuk CSS dan jQuery untuk JavaScript. Sistem Manajemen Basis Data yang digunakan adalah MySQL. Alat lain yang digunakan adalah GitHub untuk manajemen kode sumber dan kolaborasi tim.**

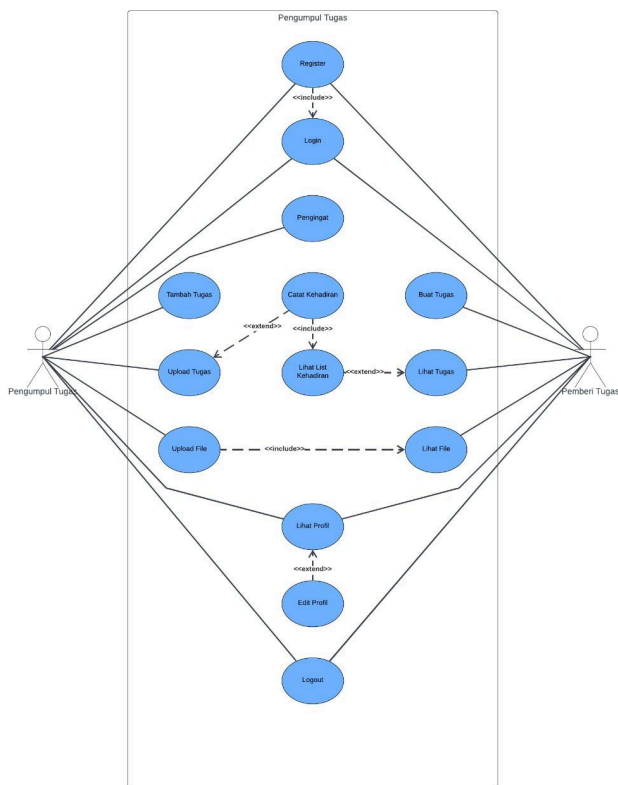
**3.3 Perancangan Sistem**

Perancangan sistem dilakukan dengan membuat wireframe dan mockup untuk menggambarkan antarmuka pengguna dan alur interaksi website. Diagram alur sistem dan use case dapat dilihat pada Gambar 1 dan Gambar 2.



## III. METODE PENELITIAN

Gambar 1.0

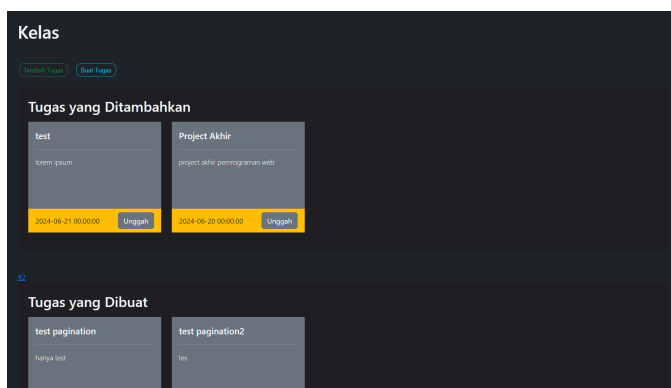


Gambar 2.0

Dengan mengimplementasikan metode penelitian maka didapatkan hasil program yang dikembangkan beserta hasil testing yang dilakukan.

## A. Hasil Program

### 1. Daftar Tugas



List Tugas menampilkan semua tugas yang ditambahkan dan dibuat. Selain itu juga terdapat tombol untuk menambahkan dan membuat tugas.

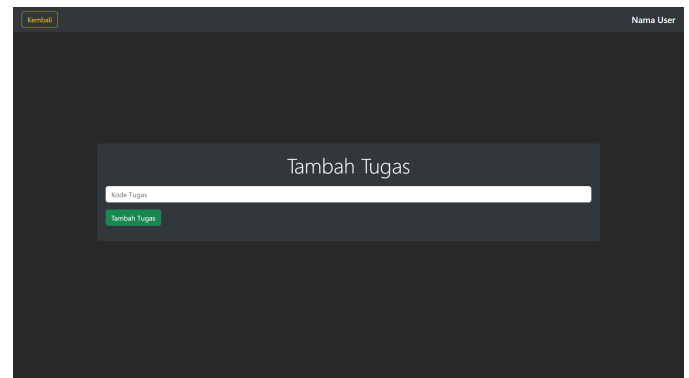
### 2. Tambah Tugas

3.4 Implementasi Dalam tahap implementasi atau pengkodean, terdapat tantangan berupa konflik kode dengan programmer lain dalam tim. Hal ini diatasi dengan melakukan komunikasi yang baik, membagi tugas secara jelas, dan menggunakan manajemen kode sumber seperti GitHub untuk menghindari konflik saat menggabungkan kode.

3.5 Pengujian Metode pengujian yang dilakukan adalah pengujian unit (unit testing) untuk memastikan setiap komponen atau fungsi pada website bekerja dengan baik secara terpisah. Hasil pengujian unit disajikan dalam bentuk deskripsi naratif. Selain itu, dilakukan evaluasi website yang melibatkan programmer website sendiri untuk menilai kualitas dan kinerja website secara keseluruhan. Proses evaluasi ini dilakukan setelah pengujian unit selesai dan sebelum website diimplementasikan secara penuh.

Dalam setiap tahapan pengembangan, tim menggunakan GitHub sebagai alat manajemen kode sumber dan kolaborasi antar anggota tim. Hal ini membantu menjaga keterbaruan kode, melacak perubahan, dan memudahkan penggabungan kontribusi dari setiap anggota tim.

## IV. HASIL & PEMBAHASAN



Tambah tugas untuk menambahkan tugas dengan memasukkan kode tugas yang sudah dibuat.

### 3. Buat Tugas

Buat Tugas untuk membuat tugas dengan memasukkan informasi berupa judul, deskripsi, deadline, dan kode tugas.

#### 4. Upload Tugas

Upload Tugas menampilkan informasi tugas yang ditambahkan secara lengkap serta terdapat fitur untuk mencatat kehadiran, upload file, dan forum diskusi.

#### B. Hasil Testing

Tabel 1.  
Black Box hasil uji coba

Skenario dan Hasil Uji				
Fitur	Test Case	Pengujian	Output Harapan	Status
Autentikasi	Validasi input kosong	Username: (kosong) Password: (kosong)	Menampilkan pesan untuk mengisi Input	Berhasil
	Validasi	Username:	Pesan	Berhasil

	input salah	admin Password: Pass	error: Username /Password Salah	
	Validasi input benar	Username: kalpinkenjeran Password: kalpin123	Menuju ke halaman Peningkat	Berhasil
Menampilkan Data	Tampilkan List Tugas	Masuk ke halaman List Tugas	Data semua tugas muncul	Berhasil
	Tampilkan Peningkat	Masuk ke halaman Peningkat	Menampilkan Peningkat jika ada	Berhasil
Pengelolaan Data	Tambah tugas baru	Judul: tes Deskripsi: tes123 Deadline: 30-06-2024 Kode Tugas: tes1234	Tugas berhasil ditambahkan	Berhasil
	Edit Profil	Nama: Kslfin Syah Username: kalpinkenjeran Password Lama: kalpin123	Profil berhasil diubah	Berhasil

#### V. KESIMPULAN

Berdasarkan implementasi metode penelitian yang telah dilakukan dalam pengembangan website pengumpul tugas menggunakan arsitektur MVC (Model-View-Controller) dan PHP, berikut adalah kesimpulan utama yang didapat:

##### 1. Pemisahan Kode:

- Implementasi arsitektur MVC berhasil memisahkan logika bisnis, tampilan, dan kontrol. Hal ini memudahkan pengembangan dan pemeliharaan kode,

serta memungkinkan pengembangan aplikasi yang lebih terstruktur dan terorganisir.

2. **Efisiensi Pengembangan:**

- Dengan MVC, pengembangan paralel oleh anggota tim menjadi lebih efisien. Setiap anggota dapat fokus pada komponen yang berbeda (Model, View, Controller) tanpa saling mengganggu, sehingga meningkatkan produktivitas tim secara keseluruhan.

3. **Fitur Utama:**

- Beberapa fitur utama yang berhasil diimplementasikan meliputi daftar tugas, penambahan tugas baru, pembuatan tugas, pengunggahan tugas, dan tampilan tugas. Fitur-fitur ini memungkinkan pengguna untuk mengelola dan mengumpulkan tugas dengan mudah dan efisien.

4. **Hasil Pengujian:**

- Pengujian dilakukan melalui berbagai skenario, termasuk validasi input kosong, input salah, dan input benar, serta pengujian fitur seperti penambahan tugas baru, pengelolaan profil, dan tampilan data tugas. Hasil pengujian menunjukkan bahwa semua fitur bekerja sesuai dengan spesifikasi yang diharapkan.

5. **Keuntungan MVC:**

- Penerapan MVC terbukti memberikan keuntungan dalam hal pemisahan kode, skalabilitas, dan efisiensi pengembangan. Pendekatan ini juga mempermudah proses debugging dan penambahan fitur baru di masa depan.

6. **Manfaat untuk Pengguna:**

- Website yang dikembangkan mampu memenuhi kebutuhan pengguna dalam mengelola dan mengumpulkan tugas secara efisien. Antarmuka pengguna yang jelas dan terstruktur memudahkan pengguna dalam berinteraksi dengan sistem.

Kesimpulan ini menegaskan bahwa penerapan arsitektur MVC dalam pengembangan website pengumpul tugas memberikan dampak positif pada struktur kode, efisiensi pengembangan, dan pengalaman pengguna.

REFERENSI

[1]