

Penerapan Arsitektur MVC pada Website Pengumpul Tugas Menggunakan PHP

Irsyad Fadhil Makarim¹, Ayodya Fade Edfira², Aileen Anindya Ramadhani³, Michael Christiano⁴, Wisanggeni Athorih Kuswirasatya⁵, Fawwaz Ali Akbar^{6*}

1. 2. 3. 4. 5. 6 Informatika, UPN “Veteran” Jawa Timur

¹22081010131@student.upnjatim.ac.id

²22081010120@student.upnjatim.ac.id

³22081010121@student.upnjatim.ac.id

⁴22081010123@student.upnjatim.ac.id

⁵22081010127@student.upnjatim.ac.id

*Corresponding author email: 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, jQuery, MySQL.

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. *Novelty* dari *website* ini dibandingkan dengan *website* pengumpul tugas sejenis adalah penggunaan arsitektur MVC yang memungkinkan pemisahan yang jelas antara logika bisnis, tampilan, dan kontrol alur, sehingga mempermudah dalam pemeliharaan dan pengembangan fitur baru. Efisiensi yang diakibatkan oleh penggunaan arsitektur MVC dapat diukur melalui beberapa indikator, seperti waktu yang dibutuhkan untuk menambahkan atau memperbarui fitur, kecepatan respon sistem terhadap input pengguna, dan tingkat kepuasan

pengguna dalam mengelola tugas melalui antarmuka yang lebih terstruktur dan mudah dipahami.

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

A. Model-View-Controller (MVC)

Model-View-Controller (MVC) adalah konsep desain perangkat lunak yang membagi aplikasi menjadi tiga komponen utama: *Model*, *View*, dan *Controller*. *Model* mengelola data aplikasi, *View* menampilkan data kepada pengguna, dan *Controller* menangani input pengguna dan mengupdate *Model*. Konsep ini pertama kali diperkenalkan oleh Trygve Reenskaug pada tahun 1979 di Smalltalk-80.

B. PHP sebagai Bahasa Pemrograman

PHP digunakan sebagai bahasa utama dalam pengembangan website. PHP biasanya digunakan untuk mengelola logika *backend*, mengakses basis data, dan me render halaman web.

C. 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.

D. 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. Sedangkan *Controller* menangani logika aplikasi, seperti menerima input dari pengguna, memproses data, dan mengarahkan aplikasi ke *view* yang sesuai.

E. 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.

III. METODE PENELITIAN

A. Model Pengembangan Perangkat Lunak

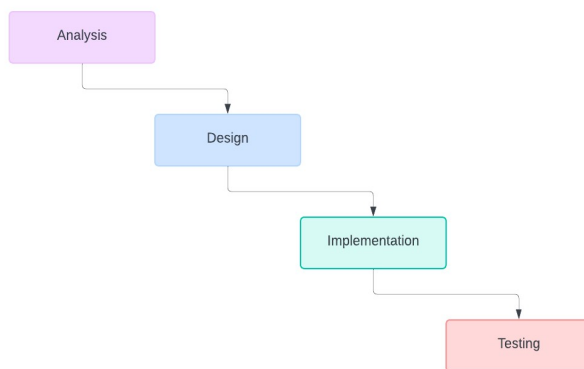
Model pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah model Waterfall yang dimodifikasi.

1) Pemilihan model ini didasarkan pada beberapa pertimbangan:

- Kebutuhan sistem telah terdefinisi dengan jelas di awal proyek
- Skala proyek relatif kecil dengan tim yang terbatas
- Waktu pengembangan yang singkat dan terbatas
- Kemudahan dalam manajemen dan kontrol proyek karena tahapan yang sekuensial.

2) Tahapan model Waterfall yang diimplementasikan meliputi:

1. Analisis Kebutuhan: Identifikasi dan dokumentasi kebutuhan fungsional dan non-fungsional website.
2. Perancangan: Perancangan arsitektur sistem, struktur database, dan antarmuka pengguna berdasarkan hasil analisis.
3. Implementasi: Pengembangan kode program dengan menerapkan arsitektur MVC menggunakan PHP.
4. Pengujian: Pelaksanaan unit testing dan evaluasi keseluruhan website.



Gbr. 1 Diagram Metode *Waterfall*.

3) Untuk meningkatkan fleksibilitas, beberapa adaptasi diterapkan:

- Review di setiap akhir tahap sebelum melanjutkan ke tahap berikutnya
- Iterasi terbatas antar tahap yang berdekatan jika diperlukan
- Penggunaan GitHub untuk manajemen kode sumber dan kolaborasi tim

B. Alat dan Teknologi Pengembangan

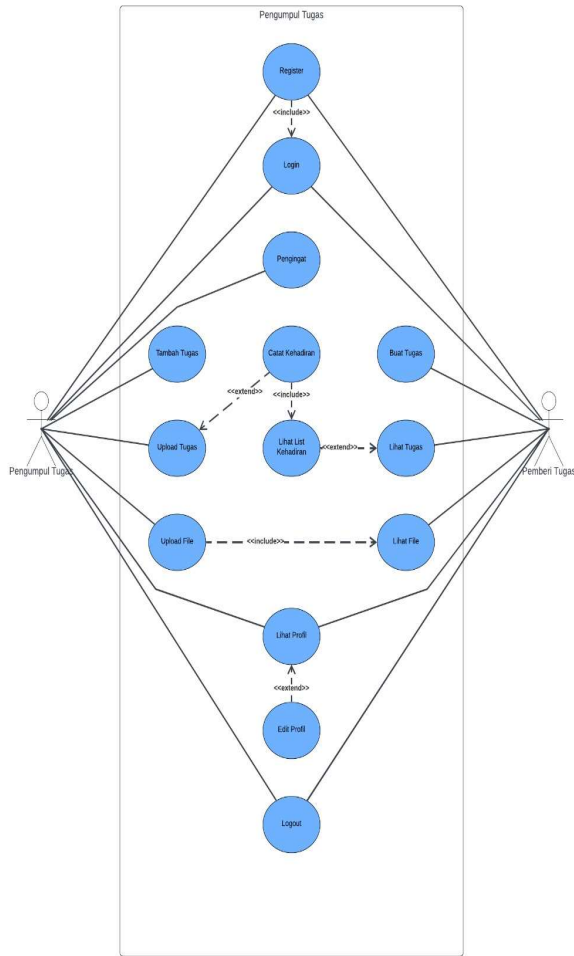
Pengembangan website menggunakan bahasa pemrograman PHP, HTML, CSS, dan JavaScript. Framework yang digunakan adalah Bootstrap untuk CSS dan jQuery untuk JavaScript. Sistem Manajemen Basis Data yang digunakan adalah MySQL.

GitHub digunakan untuk manajemen kode sumber dan kolaborasi tim.

C. 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 Gbr. 1 dan Gbr. 2.

1) Use Case Diagram:

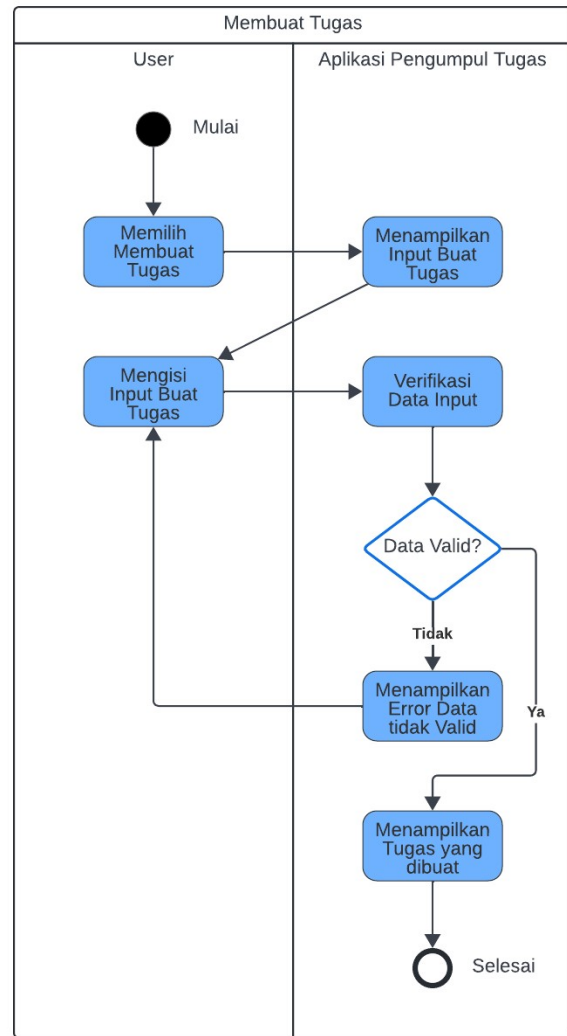


Gbr. 2 Use Case Diagram.

Use Case merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat, mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat [8]. Use case diagram tersebut menggambarkan interaksi antara pengguna dengan sistem pengelolaan tugas. Pengguna dibagi menjadi dua peran utama: Pengumpul Tugas dan Pemberi Tugas. Pengumpul Tugas dapat melakukan berbagai aktivitas seperti registrasi, login, menambah dan mengupload tugas, mencatat kehadiran, serta mengelola profil. Pemberi Tugas memiliki hak untuk membuat tugas, melihat

tugas, melihat file, dan mengelola kehadiran, serta kedua peran tersebut dapat melakukan logout.

2) Activity Diagram:

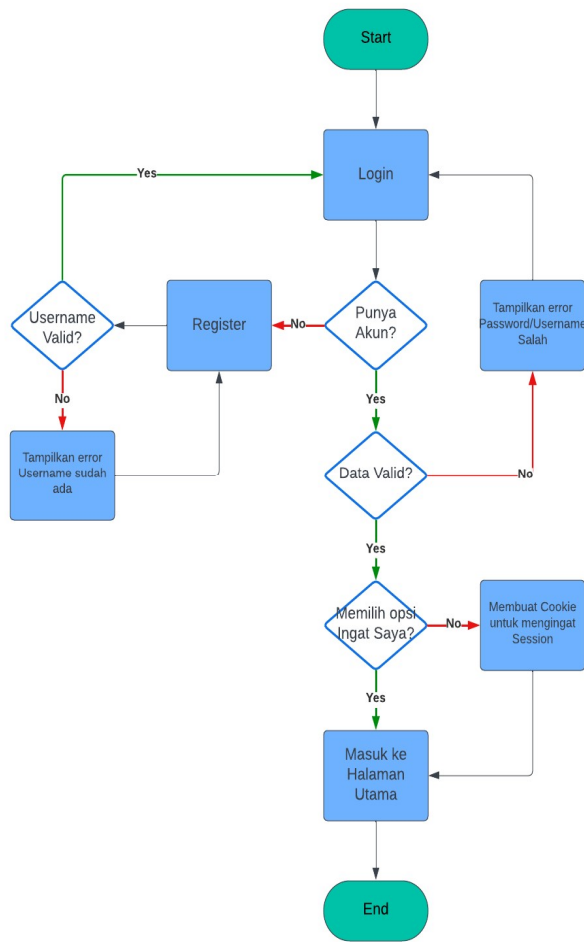


Gbr. 3 Activity Diagram Membuat Tugas.

Activity diagram adalah diagram yang memaparkan urutan tindakan dalam sistem yang dibuat. Diagram ini menunjukkan urutan aktivitas yang terjadi, keputusan atau cabang yang diambil, serta hubungan antara aktivitas-aktivitas tersebut [9]. Gbr. 3 diatas menunjukkan Activity Diagram pada fitur Membuat Tugas, dimulai dengan memilih opsi membuat tugas kemudian aplikasi akan menampilkan input yang diperlukan untuk membuat tugas. Kemudian pengguna memasukkan data pada input yang ditampilkan, setelah itu aplikasi memverifikasi apakah data yang diinput pengguna sudah benar, jika iya maka aplikasi akan menampilkan tugas yang dibuat sedangkan jika tidak aplikasi akan menampilkan

pesan *error* bahwa data yang dimasukkan tidak valid lalu pengguna mengisi data lagi.

3) *User Flow Diagram*:

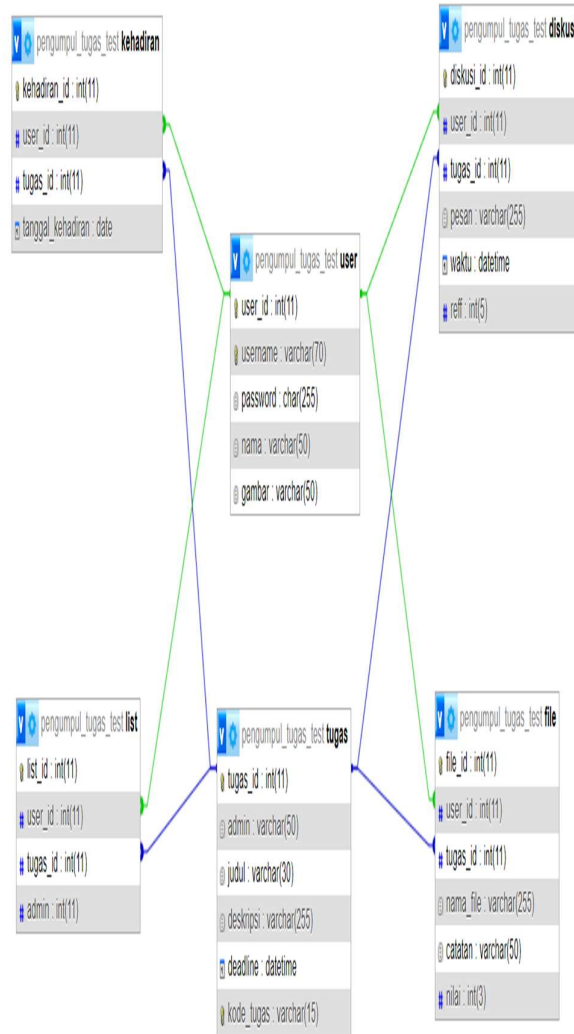


Gbr. 4 *User Flow Diagram* Sistem Autentikasi.

Diagram User Flow tersebut menunjukkan proses autentikasi pada aplikasi. Alur dimulai dari pengguna memasukkan akun yang sudah terdaftar, jika pengguna belum pernah membuat akun, maka akan diarahkan ke halaman Register. Setelah pengguna memasukkan data yang diperlukan aplikasi akan memverifikasi apakah data yang dimasukkan valid, jika iya maka pengguna akan diarahkan ke halaman Login sedangkan jika tidak aplikasi akan menampilkan error dan pengguna mengisi ulang data dengan benar. Pada halaman Login jika pengguna memasukkan data yang tidak valid maka aplikasi akan menampilkan pesan error dan meminta pengguna untuk memasukkan data yang benar, sebaliknya jika data yang dimasukkan sudah benar maka aplikasi akan memeriksa apakah pengguna memilih opsi untuk ingat saya, jika ya maka aplikasi akan membuat *cookie* untuk mengingat session dan

mengarahkan pengguna ke halaman utama, jika tidak maka pengguna akan langsung diarahkan ke halaman utama.

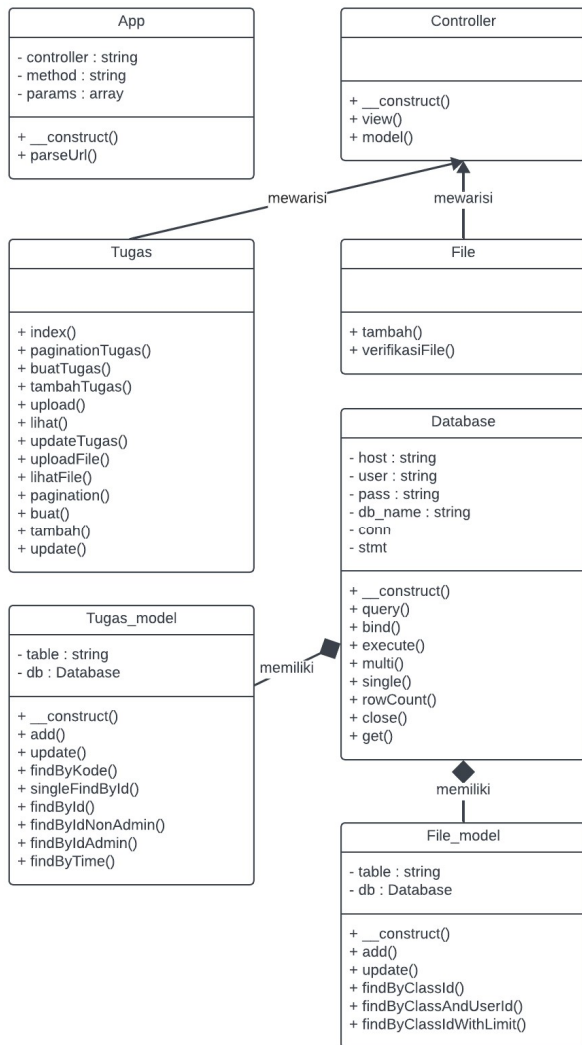
4) *Entity Relationship Diagram*:



Gbr. 5 *Entity Relationship Diagram*.

Terdapat enam tabel yang diperlukan aplikasi di dalam database, setiap tabel memiliki hubungan pada tabel lain, seperti tabel list yang terhubung ke tabel user dan tugas yang menandakan bahwa tabel list merupakan tabel penghubung antara user yang memiliki tugas. Selain itu terdapat tabel file untuk menyimpan data file, tabel diskusi untuk menyimpan data diskusi, dan tabel kehadiran untuk menyimpan data kehadiran setiap user.

5) Class Diagram:



Gbr. 6 Class Diagram Tugas dan File.

Class diagram adalah sebuah diagram yang menjelaskan hubungan antara kelas dalam rancangan model sebuah sistem. Diagram ini melihat struktur statis dari sistem dengan menyoroti kelas-kelas yang ada, atribut yang dimiliki oleh kelas tersebut, dan hubungan antara kelas tersebut [10]. Class diagram tersebut menggambarkan struktur dan hubungan antar kelas dalam aplikasi pengelolaan tugas. Kelas `App` berfungsi menginisialisasi controller, method, dan parameter URL, sedangkan kelas `Controller` menyediakan metode untuk tampilan dan model. Kelas `Tugas` dan `File` mengelola tugas dan file, dengan `Tugas` menangani berbagai operasi tugas seperti menambah, mengupload, dan memperbarui tugas. Kelas `Database` dan `File_model` mengelola koneksi basis data dan operasi file, mendukung fungsionalitas CRUD dan query data.

D. Implementasi

Dalam tahap implementasi, terdapat tantangan berupa konflik kode antar programmer dalam tim. Hal ini diatasi dengan komunikasi yang baik, pembagian tugas yang jelas, dan penggunaan GitHub untuk menghindari konflik saat menggabungkan kode.

E. Pengujian

Metode pengujian yang dilakukan adalah unit testing untuk memastikan setiap komponen bekerja dengan baik secara terpisah. Hasil pengujian unit disajikan dalam bentuk deskripsi naratif. Selain itu, dilakukan evaluasi website yang melibatkan programmer untuk menilai kualitas dan kinerja website secara keseluruhan.

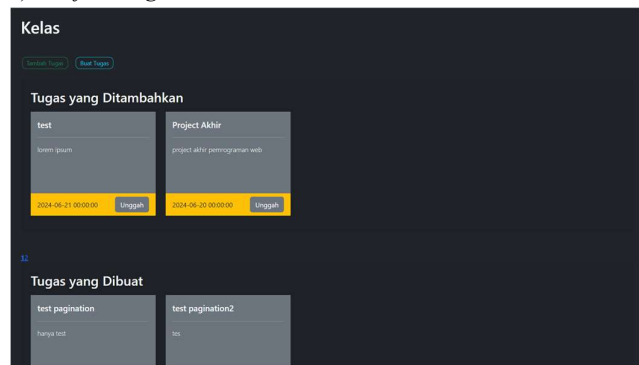
IV. HASIL & PEMBAHASAN

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

A. Hasil Program

Berikut adalah hasil program yang didapatkan setelah melewati proses pengembangan.

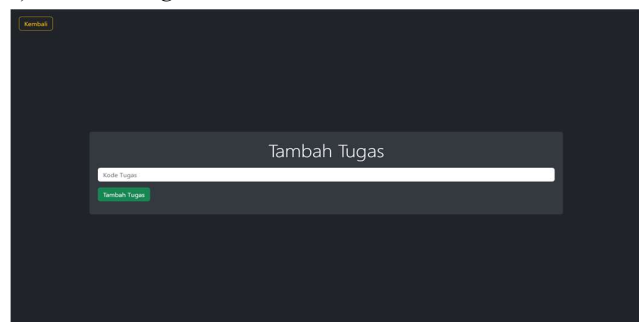
1) Daftar Tugas:



Gbr. 7 Tampilan Daftar Tugas.

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

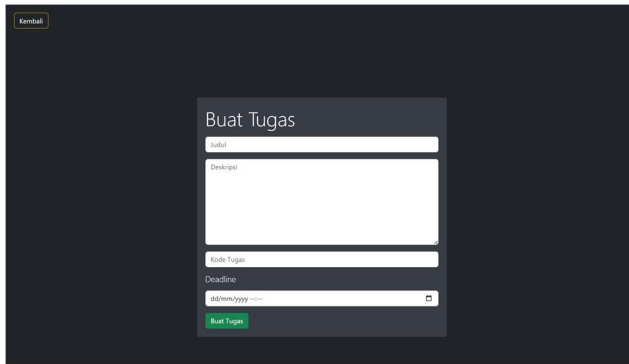
2) Tambah Tugas:



Gbr. 8 Tampilan Tambah Tugas.

Tambah tugas memiliki fungsi untuk menambahkan tugas dengan memasukkan kode tugas dari tugas yang sudah dibuat..

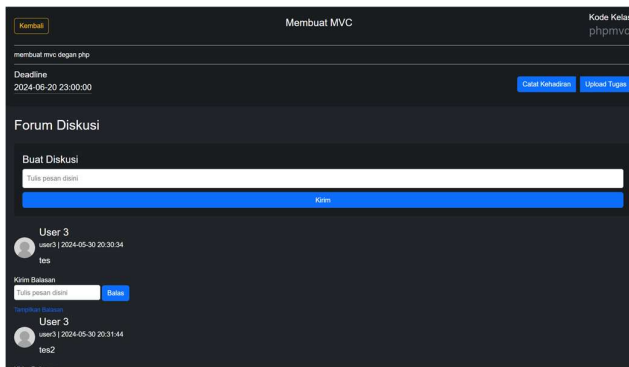
3) *Buat Tugas:*



Gbr. 9 Tampilan Buat Tugas.

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

4) *Upload Tugas:*



Gbr. 10 Tampilan 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

Dari hasil Program yang didapatkan, maka dilakukan Testing dengan metode Black Box pada Tabel 1.

TABEL I
BLACKBOX HASIL UJI COBA

Skenario dan Hasil Uji				
Fitur	Test Case	Pengujian	Output Harapan	Status
Login	Validasi input kosong	Username: (kosong)	Menampilkan pesan untuk	Berhasil

		Password: (kosong)	mengisi Input	
	Validasi input salah	Username: admin Password: Pass	Pesan error: Username/ Password Salah	Berhasil
	Validasi input benar	Username: kalpinkenjeran Password: kalpin123	Menuju ke halaman Home	Berhasil
Register	Validasi input kosong	Username: (kosong) Nama: (kosong) Password: (kosong)	Menampilkan pesan untuk mengisi Input	Berhasil
	Validasi input salah	Username: testing123 Nama: Tes 123 Password: tes123	Menampilkan Pesan error: Username sudah ada!	Berhasil
	Validasi input benar	Username: test123 Nama: Tes 123 Password: tes123	Menuju ke halaman Login	Berhasil
Home	Tampilkan Pengingat	Masuk ke halaman Pengingat	Menampilkan Pengingat jika ada	Berhasil
List Tugas	Tampilkan List Tugas	Masuk ke halaman List Tugas	Data semua tugas muncul	Berhasil
Tambah Tugas	Menambahkan Tugas dengan Kode Tugas yang kosong	Kode Tugas: (kosong)	Menampilkan pesan untuk mengisi Kode Tugas	Berhasil
	Menambahkan Tugas dengan Kode Tugas yang salah	Kode Tugas: tambahtugas	Menampilkan pesan error: Tugas tidak ditemukan	Berhasil
	Menambahkan Tugas	Kode Tugas:	Menuju ke halaman	Berhasil

	dengan Kode Tugas yang benar	tes123	List Tugas	
Buat Tugas	Membuat Tugas baru dengan input kosong	Judul: tes Deskripsi: tes123 Deadline: 30-06-2024 Kode Tugas: (kosong)	Menampilkan pesan untuk mengisi Kode Tugas	Berhasil
	Membuat Tugas baru dengan input yang salah	Judul: tes Deskripsi: tes123 Deadline: 30-06-2024 Kode Tugas: tes123	Menampilkan pesan error: Kode Tugas sudah ada	Berhasil
	Membuat Tugas baru dengan mengisi semua input dengan benar	Judul: tes Deskripsi: tes123 Deadline: 30-06-2024 Kode Tugas: tes1234	Tugas berhasil ditambahkan	Berhasil
Upload Tugas	Catat Kehadiran jika belum	Menekan tombol Catat kehadiran	Menampilkan Pesan sukses: Berhasil mencatat kehadiran	Berhasil
	Catat Kehadiran jika belum	Menekan tombol Catat kehadiran	Menampilkan Pesan peringatan: Anda sudah mencatat kehadiran	Berhasil
	Upload/Edit File dengan syarat file yang salah	Upload File: (ekstensi salah) Catatan: tes	Menampilkan Pesan error: Gagal Upload File, Ekstensi File Salah!	Berhasil
	Upload/Edit File dengan syarat file yang benar	Upload File: (memenuhi semua syarat)	Menampilkan Pesan sukses: Berhasil Upload	Berhasil
		Catatan: tes123	File	
	Menampilkan Diskusi	Menekan tombol Upload Tugas	Menampilkan semua diskusi dan balasannya	Berhasil
	Mengirim diskusi baru	Isi Diskusi: Halo, untuk tugasnya nanti ekstensi file nya apa ya?	Memuat ulang halaman dan menampilkan diskusi yang dibuat	Berhasil
	Membalas Diskusi	Memilih diskusi yang ingin dibalas; Isi Balasan: baik, terima kasih	Memuat ulang halaman dan menampilkan balasan yang dibuat	Berhasil
Lihat Tugas	Menampilkan Catatan Kehadiran	Menekan tombol Catatan Kehadiran	Menampilkan semua Catatan Kehadiran yang ada	Berhasil
	Menampilkan Daftar File	Menekan tombol Semua File	Menampilkan semua File yang pernah di unggah	Berhasil
	Menampilkan Diskusi	Menekan tombol Upload Tugas	Menampilkan semua diskusi dan balasannya	Berhasil
	Mengirim diskusi baru	Isi Diskusi: Halo, semuanya materi nya saya kirim di grup ya	Memuat ulang halaman dan menampilkan diskusi yang dibuat	Berhasil
	Membalas Diskusi	Memilih diskusi yang ingin dibalas; Isi Balasan: untuk ekstensi file nya pdf mas	Memuat ulang halaman dan menampilkan balasan yang dibuat	Berhasil
Profil	Menampilkan Profil Pengguna	Menekan tombol Profil	Menampilkan semua data Profil	Berhasil

			Pengguna	
Edit Profil tanpa mengubah password	Nama: Kalfin Syah Username: kalpinkenjeran Password Baru: (kosong) Konfirmasi Password: (kosong) Password Lama: kalpin123	Menampilkan pesan sukses: Profil berhasil diubah	Berhasil	
Edit Profil dan mengubah password dengan input yang salah	Nama: Kalfin Syah Username: kalpinkenjeran Password Baru: kalpin1234 Konfirmasi Password: kalpin12 Password Lama: kalpin123	Menampilkan pesan error: Gagal mengubah profil, konfirmasi password salah	Berhasil	
Edit Profil dan mengubah password dengan input yang benar	Nama: Kalfin Syah Username: kalpinkenjeran Password Baru: kalpin1234 Konfirmasi Password: kalpin1234 Password Lama: kalpin123	Menampilkan pesan sukses: Profil berhasil diubah	Berhasil	

V. KESIMPULAN

Implementasi arsitektur Model-View-Controller (MVC) pada website pengumpul tugas dengan PHP telah memberikan keunggulan signifikan dalam pengembangan dan pemeliharaan aplikasi, berkat pemisahan komponen yang memudahkan debugging dan penambahan fitur baru. Pengembangan paralel menjadi lebih efisien, memungkinkan anggota tim bekerja secara independen pada komponen yang berbeda. Hasil pengujian menunjukkan fitur utama seperti pengelolaan daftar tugas, pembuatan dan pengunggahan tugas, serta tampilan status tugas, berfungsi sesuai spesifikasi dan memberikan pengalaman pengguna yang baik. Penggunaan metode

pengembangan Waterfall yang dimodifikasi efektif dalam memastikan kebutuhan sistem terdefinisi dengan jelas, dan proyek dapat dikelola dengan baik melalui komunikasi yang baik dan penggunaan GitHub untuk manajemen kode sumber, mengatasi tantangan koordinasi tim dan konflik kode.

Untuk meningkatkan fungsionalitas dan keamanan website, disarankan untuk menambahkan fitur notifikasi dan pengingat tugas berbasis email atau SMS, serta integrasi dengan aplikasi kalender populer. Penerapan enkripsi data, otentikasi dua faktor (2FA), dan penggunaan protokol HTTPS harus diimplementasikan untuk meningkatkan keamanan data. Selain itu, optimalisasi pengalaman pengguna dapat dicapai melalui survei dan pengumpulan umpan balik secara berkala, memungkinkan penyesuaian desain antarmuka dan fungsionalitas yang lebih tepat sasaran, serta penyediaan pelatihan pengguna dan dokumentasi yang komprehensif untuk memaksimalkan pemanfaatan semua fitur.

REFERENSI

- [1] Nurlailah, E., & Nova Wardani, K. R. (2023). PERANCANGAN WEBSITE SEBAGAI MEDIA INFORMASI DAN PROMOSI OLEH-OLEH KHAS KOTA PAGARALAM. *JUPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 8(4), 1175–1185. <https://doi.org/10.29100/jupi.v8i4.4006>
- [2] Riyanto, A. D. (2015). PEMBUATAN WEBSITE SEBAGAI MEDIA PROMOSI YANG TERPERCAYA. <http://smpn3delanggu.sch.id/>
- [3] Kurnianti, A., Setyawan, H., Firwan Firdaus, F., & Nira Yuma, S. (2018). Perancangan dan Pembuatan Website Pengambilan Ide Skripsi Pada Prodi Teknik Informatika UMY. *Semesta Teknika*, 21(2). <https://doi.org/10.18196/st.212228>
- [4] Putra, S. H., & Nehe, A. S. (2020). Perancangan Website Media Berita Elektronik dan Tutorial Pembelajaran dengan Menggunakan PHP dan MySQL. *Jurnal Minfo Polgan*, 9(1), 9-15. <https://doi.org/10.33395/jmp.v9i1.10957>
- [5] Rakib, M., Aris, V., & Ashdaq, M. (2022). Pelatihan Mendesain dan Membuat Website Bisnis bagi Siswa Sekolah Menengah Kejuruan di Kabupaten Sidenreng Rappang. *Jurnal Abdi Masyarakat Indonesia*, 2(6), 1841–1848. <https://doi.org/10.54082/jamsi.549>
- [6] Wijayanto, H., Kumarahadi, Y. K., & Prabowo, I. A. (2023). IMPLEMENTASI MODEL WATERFALL DAN PENGUJIAN SYSTEM USABILITY SCALE (SUS) PADA PEMBUATAN WEBSITE PROGRAM STUDI INFORMATIKA BERBASIS WORDPRESS. *Indonesian Journal of Business Intelligence (IJUBI)*, 6(1). <https://doi.org/10.21927/ijubi.v6i1.3375>
- [7] Sutjiadi, R., Rahmawati, T., & Thomas, I. (2022). Perancangan dan Pembuatan Website Jasa Titip Barang dengan Menggunakan Metode Prototyping. *KONSTELASI: Konvergensi Teknologi Dan Sistem Informasi*, 2(1). <https://doi.org/10.24002/konstelasi.v2i1.5540>
- [8] Chandy Ophelia S, Siska Nurul Marwiyah, Suyanti. "Perancangan Aplikasi Pemesanan Menu Pada Rumah Makan Segar Menggunakan Model Waterfall dan Berbasis Web." *Kajian Ilmiah Informatika dan Komputer*, vol. 2, no. 6, Juni 2022, pp. 192-198. ISSN 2723-3898.
- [9] Kurniawan, T. B. (2020). Perancangan Sistem Aplikasi Pemesanan Makanan dan Minuman Pada Cafeteria NO Caffe di Tanjung Balai Karimun Menggunakan Bahasa Pemrograman PHP dan MySQL. *Jurnal TIKAR*, 1(2), 192–206. https://ejournal.universitaskarimun.ac.id/index.php/teknik_informatika/article/download/153/121
- [10] Hendini, A. (2016). Pemodelan UML Sistem Informasi Monitoring Penjualan dan Stok Barang (Studi Kasus: Distrik Zhezha Pontianak). *Communications of the ACM*, 4, 107–11
- [11] Sutarnan, T. (2020). *Pengantar Teknologi Informasi*. Jakarta: Bumi Aksara.
- [12] Dewi, R. (2019). *Sistem Informasi Manajemen*. Yogyakarta: Andi.

- [13] Nugroho, A. (2017). Pengembangan Aplikasi Web. Surabaya: Graha Ilmu.
- [14] Purnama, B. (2018). Dasar-dasar Sistem Informasi. Malang: Universitas Brawijaya Press.
- [15] Anwar, A. (2021). Pengenalan Pemrograman Web dengan PHP. Jakarta: PT Elex Media Komputindo.