



Pengembangan Aplikasi Web Manajemen Tugas Akademik Berbasis Model View Controller untuk Efisiensi Waktu Mahasiswa

Danendra Althaf Yuganfa*, Reno Dwi Aderelyan, Daniel Aquaries Pratama, Arya Pradipta, Stephanus Abryan Agung Pratama, Cinantya Paramita

Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Dian Nuswantoro, Semarang, Indonesia

Email: ¹*111202315022@mhs.dinus.ac.id, ²111202315017@mhs.dinus.ac.id, ³111202315003@mhs.dinus.ac.id,

⁴111202315457@mhs.dinus.ac.id, ⁵111202314998@mhs.dinus.ac.id, ⁶cinantya.paramita@dsn.dinus.ac.id

Email Penulis Korespondensi: 111202315022@mhs.dinus.ac.id

Abstrak—Kebutuhan akan sistem informasi yang mampu mengelola aktivitas dan jadwal akademik secara terintegrasi semakin meningkat di lingkungan pendidikan tinggi. Hal ini disebabkan oleh mahasiswa yang sering menghadapi kesulitan dalam mengelola waktu secara efisien dan efektif. Penelitian ini bertujuan untuk mengembangkan aplikasi manajemen tugas berbasis web yang tidak hanya fungsional, tetapi juga teruji kualitas pengalamannya. Dengan fitur-fitur seperti pencatatan tugas, kalender, dan notifikasi secara real-time. Metode pengembangan dilakukan menggunakan pendekatan Waterfall, mulai dari pengumpulan kebutuhan sampai desain sistem yang mencakup tiga komponen Utama yaitu: Use Case Diagram, Entity Relationship Diagram, dan Activity Diagram. Pada tahap Implementasi didukung dengan arsitektur Model View Controller (MVC) untuk memastikan bahwa proses pengembangan berjalan secara efisien dan terorganisir. Kemudian, pada tahapan pengujian Black Box Testing dan User Acceptance Test (UAT) dengan metode Mean Opinion Score (MOS) dilakukan untuk memvalidasi sistem. Pada Hasil penelitian menunjukkan bahwa seluruh fungsi berjalan sesuai rancangan, dengan tingkat kepuasan pengguna tertinggi pada aspek fungsionalitas dan performa. Kontribusi penelitian ini terletak pada efektivitas notifikasi real-time berbasis cloud yang terbukti lebih unggul dalam meningkatkan kesadaran pengguna dibandingkan metode konvensional, serta ketersediaan solusi manajemen waktu yang terverifikasi memiliki tingkat penerimaan pengguna yang tinggi.

Kata Kunci: Manajemen Waktu; Website; Waterfall; MVC; CI4; Black Box Testing; UAT; MOS

Abstract—The need for an information system capable of managing academic activities and schedules in an integrated manner is increasing in higher education environments. This is due to students often facing difficulties in managing their time efficiently and effectively. This study aims to develop a web-based task management application that is not only functional but also has a proven quality of experience. With features such as task recording, calendar, and real-time notifications. The development method is carried out using the Waterfall approach, starting from gathering requirements to system design that includes three main components, namely: Use Case Diagram, Entity Relationship Diagram, and Activity Diagram. The Implementation stage is supported by the Model View Controller (MVC) architecture to ensure that the development process runs efficiently and organized. Then, at the testing stage, Black Box Testing and User Acceptance Test (UAT) with the Mean Opinion Score (MOS) method are carried out to validate the system. The results of the study show that all functions run according to design, with the highest level of user satisfaction in terms of functionality and performance. The contribution of this study lies in the effectiveness of cloud-based real-time notifications which are proven to be superior in increasing user awareness compared to conventional methods, as well as the availability of time management solutions that are verified to have a high level of user acceptance.

Keywords: Time Management; Website; Waterfall; MVC; CI4; Black Box Testing; UAT; MOS

1. PENDAHULUAN

Dalam kehidupan jaman sekarang efisiensi waktu sangat ditunjang dalam kinerja baik itu di industri maupun di akademis. Hal ini menjadi pemicu beban psikologis pada diri manusia, karena mereka dituntut untuk mengerjakan berbagai tugas secara efisien tanpa mengurangi kualitas hasil kerjanya. Hal ini sering terjadi pada kalangan mahasiswa karena banyaknya tugas yang harus diselesaikan. Menurut data yang diambil dari penelitian yang dilakukan oleh Maulana dan Syakarofath pada tahun 2025, terdapat 199 mahasiswa yang mengalami *burnout*. Hasil penelitian tersebut juga menyatakan sebanyak 81,9% dari 199 mahasiswa mengalami *burnout* kategori sedang. Hal ini dipicu karena banyak mahasiswa yang tidak bisa mengatur waktu dan tugas mereka, pernyataan ini didukung dengan data yang menunjukkan 53,3% dari 199 mahasiswa tergolong melakukan prokrastinasi akademik kategori sedang, yang memiliki persentase tertinggi daripada kategori rendah dan kategori tinggi (Maulana & Syakarofath, 2025).

Mengelola waktu dengan tidak efektif seringkali menjadi akar masalah utama yang mendorong mahasiswa untuk menunda-nunda tugas akademik mereka. Penundaan (prokrastinasi) jenis ini lebih dari sekadar menunda penyelesaian tugas, ini adalah pilihan yang disengaja untuk menunda tugas penting, suatu tindakan yang hampir selalu menyebabkan penurunan kualitas hasil. Jika pola penangguhan yang berulang ini tidak ditangani dengan strategi manajemen yang terstruktur, mahasiswa sangat rentan terhadap *burnout* akademik. *Burnout* akademik sendiri didefinisikan sebagai kondisi yang melibatkan kelelahan emosional, fisik, dan mental yang parah, yang timbul akibat tuntutan studi yang berlebihan dan terus-menerus (Cuevas-Caravaca et al., 2024). Konsekuensi dari kelelahan ini sangat merugikan, mulai dari penurunan signifikan dalam Indeks Prestasi Kumulatif (IPK) hingga kehilangan total motivasi untuk belajar.

Dari masalah yang timbul ini pada dasarnya kemampuan untuk memanajemen waktu dengan baik menjadi faktor yang penting dalam mencapai sebuah tujuan. Dimana penjadwalan memiliki peran penting dalam manajemen waktu untuk kehidupan manusia. Manajemen Waktu bukan hanya tentang bagaimana mengisi waktu dengan aktivitas sehari hari, tapi lebih ke cara bagaimana agar dapat menentukan keputusan yang tepat untuk mencapai tujuan yang lebih besar (Febrian et al., 2025). Dengan manajemen waktu, seseorang dapat menyelesaikan tugas dengan efisien tanpa harus



mengurangi nilai efektivitasnya. Selain itu dengan manajemen waktu yang baik seseorang jadi lebih percaya diri sehingga beban stresnya juga berkurang. Maka dari itu untuk membantu mahasiswa dalam mengelola waktunya diperlukan pula sistem yang dapat mengelola tugas bagi mahasiswa.

Untuk merealisasikan solusi tersebut, diperlukan sebuah *platform* yang dapat diakses secara fleksibel serta metode pengembangan yang terstruktur guna meminimalisir risiko kegagalan proyek. Pada penelitian ini penulis berfokus dengan pengembangan aplikasi berbasis web dengan mengintegrasikan berbagai fitur yang dapat mendukung pengelolaan tugas dan pengingat jadwal bagi mahasiswa. Perancangan aplikasi ini mengikuti pendekatan pengembangan perangkat lunak *Waterfall*, sebuah pendekatan yang memungkinkan perencanaan, analisis kebutuhan, dan estimasi waktu dilakukan secara sistematis dan berurutan (Pressman & Maxim, 2020; Putri et al., 2021). Pengembangan aplikasi ini melibatkan arsitektur *Model-View-Controller* (MVC) yang diterapkan untuk memisahkan antara kode pengelola data (*Model*) dan tampilan pengguna (*View*) (Permana & Sihanato, 2024; Salam et al., 2023). Dalam implementasinya, penulis didukung dengan *framework CodeIgniter* (CI) yang memiliki kinerja yang cocok dengan arsitektur MVC (Krisna et al., 2022; Sallaby & Kanedi, 2020).

Keputusan untuk menggunakan platform berbasis web didasarkan pada pertimbangan utama, yaitu aksesibilitas lintas perangkat (*cross-platform*), yang memungkinkan mahasiswa membuka aplikasi melalui gawai apa pun, baik laptop, tablet, maupun smartphone (Pressman & Maxim, 2020). Tanpa perlu repot mengunduh atau menginstal aplikasi yang memakan banyak ruang memori. Di sisi pengembangan, kami memilih *Framework CodeIgniter* (CI) karena karakteristiknya sebagai kerangka kerja yang ringan (*lightweight*) dan mampu menunjukkan kecepatan eksekusi yang lebih cepat dibandingkan dengan *framework PHP* lain. Kecepatan akses ini merupakan aspek penting dalam menentukan *User Experience* (UX), karena aplikasi *to-do* harus dapat diakses secara instan saat mahasiswa berada di tengah kesibukan mereka. Lambatnya kinerja sistem akan menghasilkan dampak bertentangan dengan tujuan desain, berisiko memperparah ketidakpuasan dan kekecewaan pengguna.

Adapun penelitian sebelumnya yang telah membahas pengembangan aplikasi berbasis *web* dengan pendekatan *Waterfall Development* dan memvalidasi bahwa dengan metode ini setiap tahapan pengembangan dapat dilaksanakan secara efektif dan teratur (Panjaitan et al., 2025). Selain dari sisi metode pengembangan, keandalan arsitektur sistem juga menjadi perhatian utama. Beberapa penelitian serupa yang sudah menerapkan arsitektur MVC dengan *Framework CI*, yang juga memvalidasi bahwa pemisahan logika dari MVC selaras dengan kinerja *framework* (Tabrani et al., 2021; Zulfahmi Syahputra et al., 2025). Ada juga penelitian yang mengembangkan aplikasi menggunakan pendekatan *Waterfall* dengan arsitektur MVC yang dibantu oleh *framework CodeIgniter* (Natasya & Mukhlis, 2024).

Berdasarkan penelitian sebelumnya, telah memvalidasi bahwa pengembangan sistem dengan pendekatan *Waterfall* dan arsitektur MVC (CI) saling berhubungan. Meskipun beberapa penelitian sebelumnya telah berhasil mengimplementasikan pendekatan *Waterfall* dengan arsitektur MVC berbasis *CodeIgniter*, mayoritas studi tersebut masih berfokus pada keberhasilan fungsionalitas semata dan belum mengintegrasikannya dengan pengujian kualitas pengalaman pengguna yang terukur secara statistik. Oleh karena itu, penelitian ini bertujuan mengembangkan sistem yang tidak hanya fungsional secara teknis, tetapi juga teruji kualitas penggunaannya. Untuk mengidentifikasi kualitas fungsionalitas dari perpaduan model *Waterfall* dan arsitektur MVC berbasis *CodeIgniter*, penelitian ini menggunakan *Black Box Testing*. Sementara itu, kualitas pengalaman pengguna dievaluasi melalui *User Acceptance Test* (UAT) dengan menerapkan metode *Mean Opinion Score* (MOS) yang menggunakan standar tertentu untuk mengukur fungsionalitas, kegunaan, dan kepuasan.

2. METODOLOGI PENELITIAN

Di tahapan perancangan dan pembuatan sistem aplikasi manajemen tugas yang memiliki tujuan untuk membantu pengelolaan aktivitas dengan terstruktur. Perancangan sistem dilakukan untuk mendefinisikan kebutuhan yang kemudian diubah menjadi bentuk visual dan konseptual, sehingga alur kerja sistem, interaksi pengguna, serta hubungan antar komponen dapat dipahami. Diagram yang disajikan pada Gambar 1 menunjukkan representasi awal struktur yang menjadi dasar dalam pengembangan aplikasi secara menyeluruh.



Gambar 1. Alur Penelitian

2.1 Infrastruktur Teknologi

Untuk memastikan kestabilan dalam pengembangan aplikasi, serta kinerja yang optimal dalam menjalankan dan menguji sistem pada penelitian ini. Diperlukan sebuah perangkat keras yang memadai untuk melakukan penelitian, sehingga pada penelitian ini digunakan perangkat yang terdiri atas prosesor *intel Core i5-120P*, *GPU Intel Iris Xe Graphics*, dan *RAM 16GB DDR4 3200MHz* dengan penyimpanan berupa *SSD* sebesar *512GB*.

Selain perangkat keras, penelitian ini juga menggunakan perangkat lunak standar yang meliputi *Visual Studio Code* sebagai kode editor utama, serta *framework CodeIgniter 4* (CI 4). CI4 dipilih karena ringan, memiliki struktur *Model-View-Controller* (MVC) yang sudah terintegrasi secara *native*, dan mudah untuk di *deploy* di berbagai

lingkungan *server*. Selain itu arsitektur ini mendukung pemisahan antara logika bisnis, pengelolaan data, dan antarmuka pengguna, sehingga proses pemeliharaan serta pengembangan fitur menjadi lebih efisien.

Dari segi bahasa pemrograman, penelitian ini menggunakan PHP sebagai bahasa utama untuk sisi *server-side*, sedangkan pada sisi *client-side* digunakan JavaScript. Pendekatan pengembangan sistem berbasis web dengan pemisahan logika aplikasi dan integrasi layanan backend diperlukan agar sistem tetap modular, mudah dikembangkan, dan responsif terhadap interaksi pengguna. Praktik pengembangan seperti ini telah diterapkan pada sistem web interaktif yang memanfaatkan layanan API dan arsitektur modular (Pradana et al., 2025). Untuk bagian antarmuka pengguna peneliti memilih *Tailwind CSS* guna memastikan tampilan yang dapat beradaptasi di berbagai ukuran layar. Dikarenakan aplikasi yang dikembangkan akan di *deploy* di *website*. Peneliti juga menerapkan komponen *Flatpickr*, komponen ini digunakan sebagai date-time picker interaktif untuk meningkatkan kenyamanan pengguna, juga *Firebase Cloud Messaging (FCM)* digunakan sebagai sistem notifikasi berbasis *push notification* untuk mengingatkan pengguna terhadap tenggat tugas secara *real-time* sehingga membuat pengguna tidak melewatkan acara atau tugas (Mumcu & Çebi, 2025). infrastruktur teknologi dioptimalkan sesuai dengan *Requirement Gathering* yang didapatkan

2.2 Requirement Gathering

Pengumpulan kebutuhan dilakukan melalui kombinasi benchmarking pada aplikasi-aplikasi besar seperti aplikasi *Google Tasks*, *Google Calendar*, analisis pustaka terkait desain aplikasi *to-do/task management* yang dilakukan menggunakan *search engine*. Pada tinjauan literatur menunjukkan mengenai fitur inti yang mendukung efektivitas sistem manajemen tugas, meliputi pembuatan dan pengelolaan tugas (CRUD), pengelompokan dan prioritas tugas, serta perancangan antarmuka yang mendukung produktivitas dan personalisasi pengguna (Torino et al., 2025).

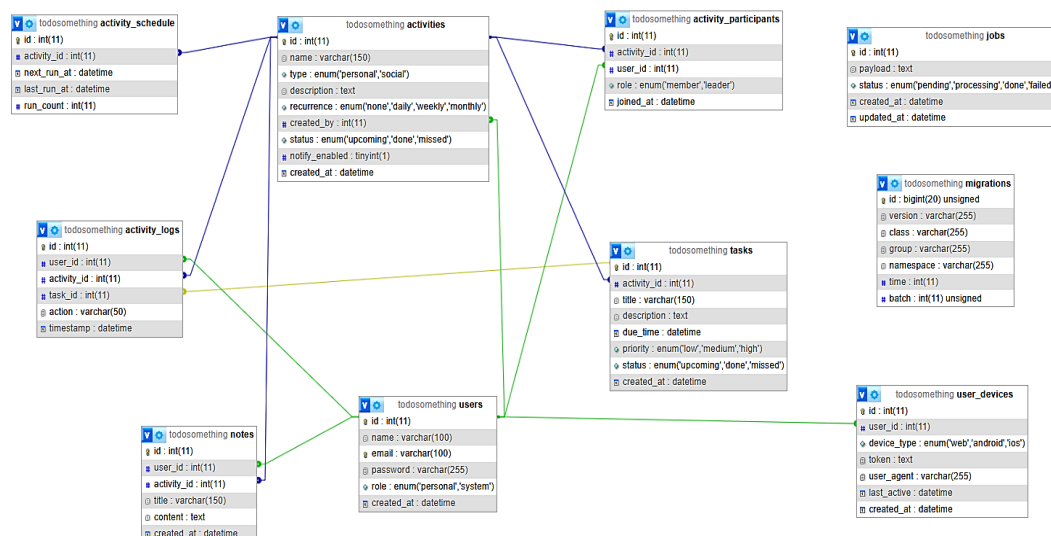
Lebih spesifik lagi, studi desain dari aplikasi *to-do* modern menggaris bawahi beberapa fitur tambahan yang meningkatkan utilisasi dan efektivitas. Fitur yang dimaksud meliputi : integrasi notifikasi *real-time (push notification)*, kemampuan sinkronisasi *multi-device*, opsi *subtasks* atau *checklist*, kemampuan penambahan catatan/*attachment*, serta tampilan kalender/agenda untuk visualisasi tenggat. Studi kasus pengembangan aplikasi *to-do* pada konteks pendidikan dan produktivitas juga merekomendasikan fitur-fitur ini sebagai prioritas fungsional ketika tujuan aplikasi adalah membantu manajemen waktu dan monitoring tugas akademik. (Imani et al., 2024; Torino et al., 2025). Sehingga kebutuhan fungsional sistem meliputi: pembuatan dan pengelolaan tugas, Mekanisme pengingat, pengelompokan tugas/jadwal, integrasi notifikasi *real-time*, *subtask*, *checklist*, tampilan kalender, visualisasi tenggat.

2.3 Design

Desain sistem mencakup dua komponen utama, yaitu *Entity Relationship Diagram (ERD)*, dan *Activity Diagram*, yang berfungsi untuk menggambarkan hubungan antar entitas, serta interaksi antar modul sistem.

2.3.1 Entity Relationship Diagram

ERD (Gambar 2) menggambarkan hubungan antar tabel pada basis data, termasuk entitas *Users*, *Tasks*, *Activities*, *Notes*, dan *Notifications*. Relasi yang terbentuk bersifat *one-to-many*, di mana satu pengguna dapat memiliki banyak tugas dan catatan. Desain *ERD* ini memastikan integritas data serta efisiensi *query* dalam pengambilan informasi terkait aktivitas dan status tugas.

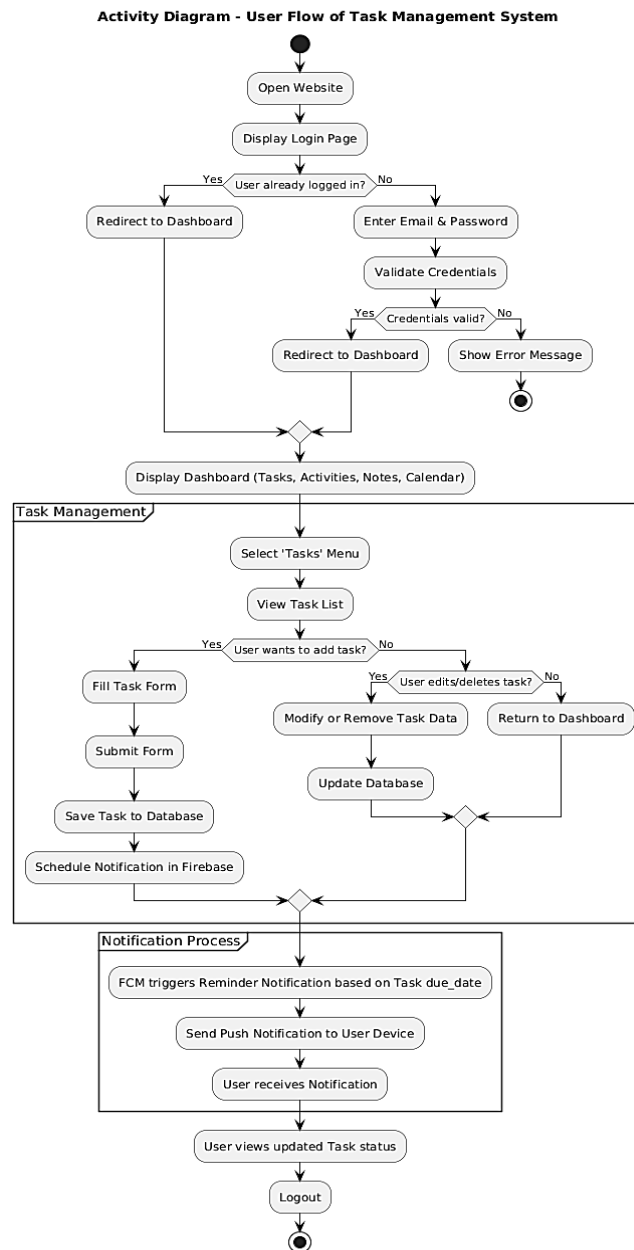


Gambar 2. Entity Relationship Diagram

2.3.2 Activity diagram

Activity diagram yang ditunjukkan Gambar 3, menunjukkan alur umum tentang proses pengguna dalam menggunakan aplikasi. Dimulai dari login sampai mengelola tugas. Diagram yang ditampilkan pada *activity diagram*, menampilkan

aliran yang diawali dengan autentikasi pengguna, kemudian navigasi menuju *dashboard*, setelah itu membuat tugas baru. Dan memicu pengingat otomatis sesuai waktu yang ditentukan. Diagram pada Gambar 3 memberikan gambaran menyeluruh tentang proses yang akan dijalani pengguna secara urut dan membantu mengidentifikasi titik keputusan dalam alur kerja penggunaan



Gambar 3. Activity Diagram

2.4 Implementation

2.4.1 Model

Pada penelitian ini, peneliti bereksperimen dengan studi kasus pengguna yang memiliki berbagai aktivitas terintegrasi dengan kalender sebagai pusat penjadwalan. Pengguna dapat membuat aktivitas yang di dalamnya terdapat catatan tambahan guna mencatat kebutuhan atau informasi penting terkait aktivitas yang dibuat oleh pengguna. Aktivitas yang dibuat pengguna menggunakan sebuah sistem yang dirancang agar aktivitas yang dibuat pengguna otomatis terhubung dengan fitur kalender dan juga mekanisme notifikasi *real-time* sesuai waktu yang ditentukan oleh pengguna. Aplikasi juga menyediakan fitur rekapitulasi aktivitas yang didasarkan pada data log aktivitas. Fitur ini memungkinkan pengguna untuk meninjau kembali seluruh kegiatan yang telah dilakukan oleh pengguna. Dengan model ini sistem akan dapat membantu pengguna dalam mengelola waktu, mencatat aktivitas serta memantau produktivitas secara lebih efisien dan terorganisir



2.4.2 View

View dirancang secara terstruktur dengan menerapkan pendekatan *layout* yang terpisah, hal ini dilakukan agar sistem mampu beradaptasi dengan data yang bersifat dinamis. Pada tahap ini, komponen *view* juga menggunakan sintaks *php* untuk menampilkan data yang diteruskan dari *controller*. Selain *php*, pada bagian *view* juga menggunakan *JavaScript* untuk menambah interaktivitas serta meningkatkan responsivitas antarmuka. *Framework Tailwind CSS* juga digunakan sebagai dasar *styling* agar tampilan antarmuka modern, serta mudah disesuaikan dengan kebutuhan desain. Dengan pendekatan ini, antarmuka menjadi lebih adaptif terhadap berbagai ukuran layar serta memberikan pengalaman pengguna yang lebih baik

2.4.3 Controller

Pada arsitektur *MVC*, *controller* berperan sebagai penghubung antara model dengan *view*. Pada penelitian ini *controller* mengatur alur kerja seperti penambahan aktivitas baru, penyimpanan catatan tambahan, integrasi jadwal dengan kalender, serta pengiriman notifikasi *real-time*. *Controller* juga menangani proses validasi input pengguna dan pengelolaan data log aktivitas untuk memastikan seluruh interaksi berjalan dengan baik dan sesuai dengan kebutuhan sistem. Dengan penerapan ini, sistem mampu menjaga konsistensi aliran data sekaligus memudahkan proses pengembangan serta pemeliharaan aplikasi di masa mendatang.

2.5 Application Testing

Pengujian yang digunakan pada aplikasi adalah *black box testing*, pengujian ini berfokus pada hasil keluaran dari aplikasi tanpa melihat kode internal atau struktur sistem (Waluyo et al., 2024). Bagian utama dari proses *testing* adalah *User Acceptance Testing (UAT)*, dimana aplikasi diberikan kepada pengguna akhir untuk melakukan evaluasi dan memberikan umpan balik. Tujuannya untuk mengumpulkan serta menganalisis data umpan balik pengguna, metode *mean opinion score (MOS)* diimplementasikan mulai dari fungsionalitas, kegunaan, sampai keandalan aplikasi. *MOS* merupakan metode yang efektif untuk mengukur persepsi subjektif pengguna tentang kualitas aplikasi. Skor yang dihasilkan dari kuesioner kemudian dianalisis untuk menghasilkan rata-rata, yang dijadikan indikasi tentang tingkat kepuasan pengguna terhadap aplikasi. Hasil dari *MOS* bukan hanya untuk menilai kinerja aplikasi yang diuji, namun juga untuk mengidentifikasi potensi peningkatan di masa depan.

Dalam menganalisis data menggunakan metode *MOS*, peneliti menerapkan persamaan 1 untuk menghasilkan skor dari survei yang telah dibagikan.

$$MOS C = \frac{r}{M} \times 100\% \tag{1}$$

Dengan, C: skor persentase yang dicari, r: perolehan skor oleh *validator*, M= skor maksimal. Bobot hasil pengujian didasarkan pada penilaian *MOS* pada Tabel 1.

Tabel 1. Bobot Penilaian MOS

Kategori	Bobot	MOS
Sangat setuju/Sangat baik	5	SS/SB
Setuju/Baik	4	S/B
Netral/Cukup	3	N/C
Kurang setuju/Kurang baik	2	KS/KB
Tidak baik/Tidak setuju	1	TB/TS

3. HASIL DAN PEMBAHASAN

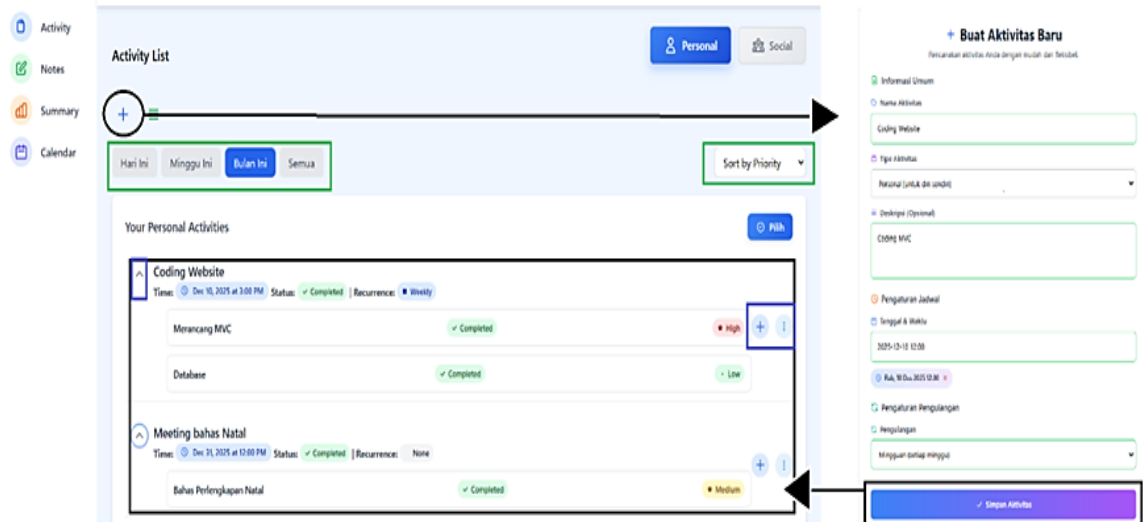
3.1 Implementasi Sistem

Berdasarkan analisis kebutuhan dan desain proyek, terbentuklah aplikasi dengan nama *Routine Master*. Pada aplikasi *Routine Master* terdiri dari berbagai fitur untuk mendukung manajemen waktu sebagai berikut:

- Activity*: Merupakan fitur paling penting dalam aplikasi ini dimana pengguna dapat menambahkan jadwal aktivitas mereka baik aktivitas tunggal, aktivitas berlanjut, dan juga aktivitas rutin.
- Task*: Fitur yang memungkinkan pengguna untuk menambahkan detail tugas pada masing masing aktivitas yang terjadwal.
- Calendar*: Menampilkan detail aktivitas beserta tugas pengguna dalam bentuk kalender, serta dilengkapi dengan fitur pengingat dalam bentuk notifikasi apabila jadwal aktivitas sudah mendekati waktu implementasi..

Tiga fitur utama tersebut saling terkait satu sama lain. Ada fitur *Activity* untuk membuat jadwal aktivitas pada tanggal dan waktu tertentu. Fitur *Task* untuk memberikan penjelasan apa saja yang perlu dikerjakan dalam aktivitas tersebut. Dan fitur *Calendar* untuk memonitoring jadwal aktivitas pengguna, sehingga pengguna tidak perlu bingung “Besok saya ada jadwal apa ya?”. Selain itu dengan dukungan fitur pengingat berupa notifikasi, untuk meminimalisir dampak dari pengguna yang sering lupa dalam menjalankan aktivitasnya.

3.1.1 Fitur Activity

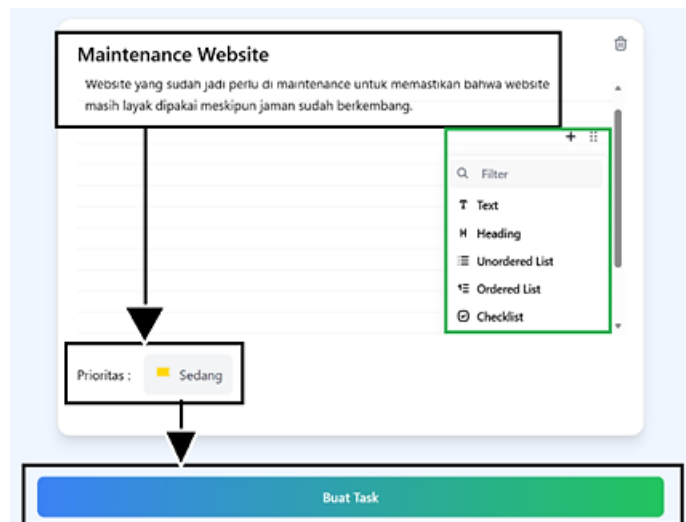


Gambar 4. User Interface Fitur Activity

Pada fitur *Activity* pengguna akan diberikan informasi berupa daftar jadwal aktivitas mereka. Pengguna dapat berinteraksi dengan tampilan daftar jadwal aktivitas mereka dengan memilih tombol yang diberi tanda kotak hijau pada Gambar 4. Tombol tersebut berfungsi untuk memfilter dan menyorting tampilan pada daftar jadwal aktivitas. Apabila pengguna belum memiliki jadwal aktivitas, maka pengguna perlu membuat jadwal terlebih dahulu dengan menekan tombol “+” pada kiri atas, kemudian mengisi *form* yang disediakan.

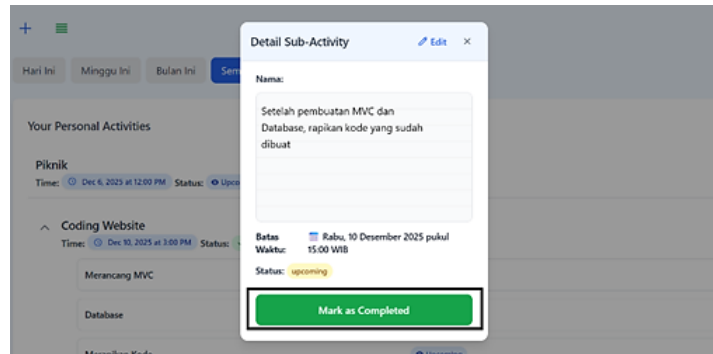
Pengguna juga dapat memilih tanggal jadwal baik untuk jadwal tunggal, jadwal yang berlanjut, dan juga jadwal berupa rutinitas. Setelah yakin dengan jadwal yang diketik, maka pengguna dapat menekan tombol “Buat Aktivitas” dan jadwal akan otomatis tampil ke dalam daftar jadwal aktivitas. Proses pembuatan jadwal aktivitas ini ditunjukkan pada Gambar 4 yang diberi tanda dengan kotak warna hitam. Adapun tanda yang diberi kotak berwarna biru adalah tanda untuk pengguna dapat berinteraksi dengan jadwal aktivitas mereka, salah satunya adalah memberikan detail penugasan.

3.1.2 Fitur Task



Gambar 5. User Interface Menambahkan Tugas

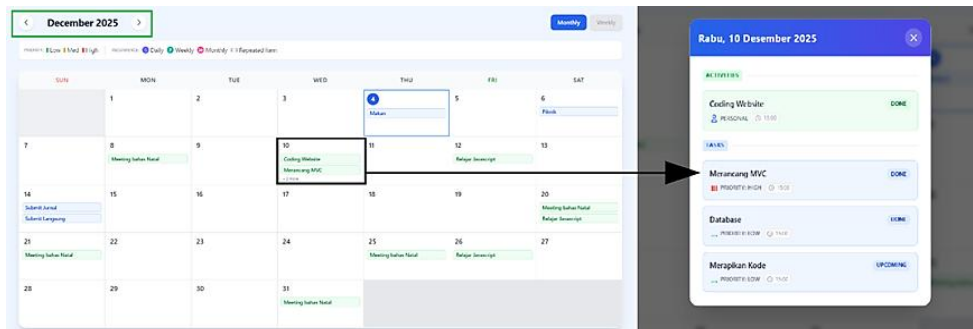
Fitur ini masih kelanjutan dari fitur *Activity*, hanya saja fitur ini difokuskan untuk memberikan detail apa saja yang perlu dikerjakan atau tugas apa saja yang ada dalam suatu aktivitas. Setelah menjadwalkan suatu aktivitas maka pengguna perlu menambahkan detail penugasan apa saja yang ada dalam aktivitas tersebut. Pengguna dapat membuat detail penugasan dengan menekan tombol “+” pada bagian sebelah kanan daftar aktivitas, yang ditunjukkan dengan kotak berwarna biru pada Gambar 4. Setelah itu *form* pengisian akan tampil dan pengguna dapat mengisi judul, deskripsi, dan prioritas penugasan seperti yang tampil pada Gambar 5 dengan penunjuk kotak berwarna hitam. Pada bagian deskripsi pengguna juga dapat memilih tipe deskripsi penugasan seperti dalam bentuk “*text*” atau “*checklist*”, yang ditunjukkan dengan kotak berwarna hijau.



Gambar 6. Dialog Detail *Task*

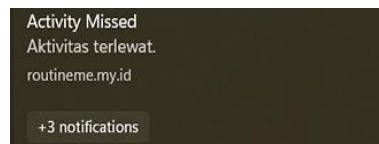
Pengguna juga dapat melihat detail tugas dengan menekan tombol “panah ke bawah” kemudian klik salah satu tugas pada daftar jadwal aktivitas, yang ditunjukkan dengan kotak berwarna biru pada gambar 4. Setelah itu akan tampil *UI* berbentuk dialog, berisi informasi tentang detail tugas seperti yang tampil pada gambar 6. Apabila tugas sudah terselesaikan pengguna dapat mengkonfirmasi dengan menekan tombol “*Mark as Completed*” yang ditunjukkan dengan kotak berwarna hitam pada gambar 5.

3.1.3 Fitur *Calendar*



Gambar 7. User Interface Fitur *Calendar*

Pengguna dapat membuat jadwal aktivitas dan memberikan detail penugasan yang perlu diselesaikan dalam suatu aktivitas, tapi itu baru satu jadwal, bagaimana jika pengguna memiliki jadwal yang banyak, akan kesulitan apabila hanya mengandalkan daftar jadwal aktivitas yang dapat difilter dan di-*sorting*. Oleh karena itu ada fitur *Calendar* yang menyesuaikan jadwal dengan pengguna. Pada kalender akan menampilkan keseluruhan tanggal yang memiliki detail jadwal yang sebelumnya pengguna sudah buat seperti yang tampil pada Gambar 7. Pengguna juga dapat melihat detail aktivitas dan tugas pada tanggal tertentu hanya dengan menekannya seperti yang ditunjukkan dengan kotak berwarna hitam. Pengguna juga dapat menggeser tampilan kalender ke bulan selanjutnya untuk memonitoring jadwal bulan depan atau tahun depan dengan menekan tombol “panah ke samping” seperti yang ditunjukkan oleh penanda kotak berwarna hijau.



Gambar 8. Notifikasi Peningat Jadwal Aktivitas

Fitur notifikasi ini sebenarnya adalah fitur yang terpisah dari ketiga fitur sebelumnya, karena ini adalah fitur tambahan pada kalender pengguna. Fitur notifikasi bertujuan untuk mengingatkan pengguna akan waktu jadwal aktivitas mereka yang sudah dekat. Pada gambar 8 ditunjukkan bahwa ada tiga notifikasi yang muncul dari aplikasi *Routine Master*. Notifikasi pengingat ini akan muncul pada browser pengguna dengan kondisi 60 menit dan 30 menit sebelum aktivitas dimulai dan akan muncul lagi pada saat aktivitas dimulai.

3.2 Hasil pengujian sistem

3.2.1 Black Box Testing

Pengujian *Black Box* dilakukan dengan membagi tiga pengujian. Fitur pertama yang diuji adalah *Activity*, kemudian dilanjutkan dengan menguji fitur *Task* dan fitur terakhir yang diuji adalah *Calendar*. Ketiga pengujian ini dilakukan karena ketiga fitur tersebut saling terkait dan mendukung satu sama lain. Oleh karena itu pengujian perlu dilakukan secara terpisah untuk ketiga fitur tersebut.



a. Pengujian Fitur Pembuatan Aktivitas

Tabel 2 berikut hasil pengujian pada fitur Activity.

Tabel 2. Pengujian Fitur Activity

Kasus Uji	Hasil Yang Diharapkan	Langkah Pengujian	Hasil Uji
Validasi <i>Form</i> apabila tidak diisi	Sistem tidak dapat menyimpan aktivitas dan pengguna diwajibkan mengisi form yang kosong.	<ol style="list-style-type: none"> Masuk menu <i>Activity</i>. Klik tombol ikon "+". Kosongkan form yang ada. Klik tombol "Simpan Aktivitas" Verifikasi gagal menyimpan aktivitas dengan menampilkan pesan eror pada <i>form</i> yang wajib diisi. 	PASSED
Memverifikasi aktivitas dengan tanggal tunggal	Memastikan bahwa aktivitas dengan satu kali tanggal eksekusi dapat terjadwal pada sistem.	<ol style="list-style-type: none"> Masuk menu <i>Activity</i>. Klik tombol ikon "+". Isi <i>form</i> nama, tipe, dan deskripsi aktivitas. Pilih tanggal untuk aktivitas misal "14 Desember 2025" Klik tombol "Simpan Aktivitas" Verifikasi dengan memeriksa daftar aktivitas pada aktivitas yang dibuat berhasil tampil. 	PASSED
Memverifikasi aktivitas dengan tanggal ganda dan waktu yang sama	Memastikan bahwa aktivitas dengan beberapa hari eksekusi di waktu yang sama dapat terjadwal pada sistem.	<ol style="list-style-type: none"> Masuk menu <i>Activity</i>. Klik tombol ikon "+". Isi <i>form</i> nama, tipe, dan deskripsi aktivitas. Pilih tanggal dan waktu untuk aktivitas misal "14 dan 15 Desember 2025" dengan waktu pukul "12:00 WIB". Klik tombol "Simpan Aktivitas" Verifikasi dengan memeriksa daftar aktivitas pada aktivitas yang dibuat berhasil tampil. 	PASSED
Memverifikasi aktivitas dengan tanggal ganda dan waktu yang berbeda	Memastikan bahwa aktivitas beberapa hari eksekusi di waktu yang berbeda dapat terjadwal pada sistem.	<ol style="list-style-type: none"> Masuk menu <i>Activity</i>. Klik tombol ikon "+". Isi <i>form</i> nama, tipe, dan deskripsi aktivitas. Pilih tanggal dan waktu untuk aktivitas misal "14 dan 15 Desember 2025" dengan waktu masing-masing pukul "12:00 WIB" dan "13:00 WIB". Klik tombol "Simpan Aktivitas" Verifikasi dengan memeriksa daftar aktivitas pada aktivitas yang dibuat berhasil tampil. 	PASSED
Memverifikasi aktivitas dengan rutinitas	Memastikan bahwa aktivitas dengan rutinitas harian dapat terjadwal dalam sistem.	<ol style="list-style-type: none"> Masuk menu <i>Activity</i>. Klik tombol ikon "+". Isi <i>form</i> nama, tipe, dan deskripsi aktivitas. Pilih tanggal dan waktu untuk dimulainya aktivitas tersebut semisal "16 Desember 2025" pukul "12:00 WIB". Pilih opsi pengaturan pengulangan dengan "Harian". Klik tombol "Simpan Aktivitas" Verifikasi dengan memeriksa daftar aktivitas pada aktivitas yang dibuat berhasil tampil. 	PASSED

b. Pengujian Fitur Pembuatan dan Verifikasi Tugas pada Aktivitas

Tabel 3 berikut hasil pengujian pada fitur Task.

**Tabel 3.** Pengujian Fitur *Task*

Kasus Uji	Hasil Yang Diharapkan	Langkah Pengujian	Hasil Uji
Memverifikasi tugas tunggal pada suatu aktivitas.	Memastikan bahwa satu tugas dapat terjadwal dalam suatu aktivitas.	<ol style="list-style-type: none"> 1. Masuk menu <i>Activity</i>. 2. Pastikan sudah memiliki jadwal aktivitas. 3. Klik tombol ikon "+" pada aktivitas. 4. Isi <i>form</i> judul dan catatan. 5. Klik tombol "Buat <i>Task</i>" 6. Verifikasi dengan memeriksa tugas sudah tampil pada aktivitas dengan klik tombol "panah kebawah". 	PASSED
Memverifikasi tugas ganda pada suatu aktivitas.	Memastikan bahwa tugas berganda dapat terjadwal dalam suatu aktivitas.	<ol style="list-style-type: none"> 1. Masuk menu <i>Activity</i>. 2. Pastikan sudah memiliki jadwal aktivitas. 3. Klik tombol ikon "+" pada aktivitas. 4. Isi <i>form</i> judul dan catatan. 5. Klik tombol "+". 6. Ulangi langkah 3 dan 4, dan isi disesuaikan dengan keperluan. 7. Klik tombol "Buat <i>Task</i>" 8. Verifikasi dengan memeriksa tugas sudah tampil pada aktivitas dengan klik tombol "panah kebawah". 	PASSED
Memverifikasi tugas dengan prioritas pada suatu aktivitas.	Memastikan bahwa tugas dengan prioritas (Tinggi, Sedang, atau Rendah) dapat terjadwal dalam suatu aktivitas.	<ol style="list-style-type: none"> 1. Masuk menu <i>Activity</i>. 2. Pastikan sudah memiliki jadwal aktivitas. 3. Klik tombol ikon "+" pada aktivitas. 4. Isi <i>form</i> judul dan catatan. 5. Pilih prioritas semisal "Tinggi" yang berwarna merah. 6. Klik tombol "Buat <i>Task</i>". 7. Verifikasi dengan memeriksa tugas sudah tampil dengan prioritas berwarna hijau pada aktivitas dengan klik tombol "panah kebawah". 	PASSED
Memverifikasi prioritas tugas yang berganda pada suatu aktivitas.	Memastikan bahwa tugas dengan prioritas yang beragam dapat terjadwal dalam suatu aktivitas.	<ol style="list-style-type: none"> 1. Masuk menu <i>Activity</i>. 2. Pastikan sudah memiliki jadwal aktivitas. 3. Klik tombol ikon "+" pada aktivitas. 4. Isi <i>form</i> judul dan catatan. 5. Pilih prioritas sesuai dengan warna hijau (rendah), kuning (sedang), atau merah (tinggi). 6. Klik tombol "Buat <i>Task</i>". 7. Ulangi langkah 3 sampai 6 dengan pengisian <i>form</i> dan prioritas yang berbeda. 8. Verifikasi dengan memeriksa semua tugas yang dibuat dapat terjadwal pada aktivitas dengan klik tombol "panah kebawah". 	PASSED
Memverifikasi tugas yang sudah selesai	Memastikan bahwa tugas yang sudah selesai dapat dinyatakan sebagai status " <i>Completed</i> "	<ol style="list-style-type: none"> 1. Masuk menu <i>Activity</i>. 2. Pastikan sudah memiliki tugas yang terjadwal pada aktivitas. 3. Klik tombol ikon "panah kebawah" pada aktivitas. 4. Pilih tugas yang sudah selesai. 5. <i>Checklist</i> penugasan sesuai yang selesai. 6. Apabila sudah <i>checklist</i> semua klik tombol "<i>Completed</i>" untuk menyatakan 	PASSED



Kasus Uji	Hasil Yang Diharapkan	Langkah Pengujian	Hasil Uji
-----------	-----------------------	-------------------	-----------

status tugas sudah selesai.

c. Pengujian Fitur Kalender Terjadwal
Tabel 4 berikut hasil pengujian pada fitur Calendar.

Tabel 4. Pengujian Fitur *Calendar*

Kasus Uji	Hasil Yang Diharapkan	Langkah Pengujian	Hasil Uji
Memvalidasi aktivitas yang tampil sesuai dengan tanggal pada kalender.	Memastikan bahwa detail tugas pada aktivitas tampil sesuai dengan status dan jadwal.	<ol style="list-style-type: none"> 1. Pastikan sudah memiliki aktivitas dan tugas yang terjadwal. 2. Masuk menu <i>Calendar</i>. 3. Klik tanggal pada kalender yang mengandung informasi jadwal aktivitas. 4. Verifikasi dengan tampilnya dialog perincian status aktivitas dan tugas pada tanggal yang dipilih. 	PASSED
Memvalidasi aktivitas dengan rutinitas yang tampil pada kalender.	Memastikan bahwa aktivitas dengan rutinitas harian dapat menampilkan detail status tugas pada kalender.	<ol style="list-style-type: none"> 1. Pastikan sudah memiliki aktivitas dengan rutinitas dan tugas yang terjadwal. 2. Masuk menu <i>Calendar</i>. 3. Pastikan bahwa hari ini adalah satu hari sebelum jadwal rutinitas harian. 4. Klik tanggal pada kalender. 5. Verifikasi dengan tampilnya dialog perincian status aktivitas dan tugasnya pada tanggal yang dipilih. 	PASSED
Memvalidasi jadwal aktivitas dan tugas apabila dijadwalkan pada bulan selanjutnya.	Memastikan bahwa jadwal aktivitas dan tugas tetap dapat tampil di bulan selanjutnya	<ol style="list-style-type: none"> 1. Pastikan sudah memiliki aktivitas yang terjadwal di bulan selanjutnya. 2. Masuk menu <i>Calendar</i>. 3. Geser tampilan kalender ke bulan selanjutnya. 4. Klik tanggal pada kalender yang mengandung informasi jadwal aktivitas. 5. Verifikasi dengan tampilnya dialog perincian status aktivitas dan tugas pada tanggal yang dipilih. 	PASSED
Uji Notifikasi aktivitas	Memastikan bahwa sistem menampilkan pemberitahuan pengingat perihal aktivitas yang terjadwal.	<ol style="list-style-type: none"> 1. Pastikan sudah memiliki aktivitas yang terjadwal. 2. Pastikan aktivitas akan dimulai satu jam atau tiga puluh menit setelah waktu saat ini. 3. Verifikasi dengan munculnya pemberitahuan status aktivitas pada halaman notifikasi di browser. 	PASSED

Pengujian pertama adalah fitur utama yaitu *Activity*, pengujian dilakukan berfokus pada pembuatan jadwal aktivitas, mulai dari form yang kosong, jadwal dengan tanggal tunggal dan ganda, sampai dengan jadwal rutinitas seperti harian, mingguan, dan bulanan. Pada pengujian tahap ini bertujuan untuk memastikan bahwa fitur dasar *Activity* benar-benar berfungsi dengan baik. Pernyataan tersebut terbukti dari tabel 2 yang memperlihatkan bahwa lima kasus uji berhasil dilewati. Kemudian pengujian pada tahap kedua yang bertujuan untuk memastikan bahwa setelah aktivitas dibentuk maka pengguna dapat memberikan detail penugasan pada jadwal aktivitas mereka. Dengan hasil pengujian pada tabel 3, membuktikan bahwa proses tersebut dapat dilakukan oleh fitur *Task* dengan benar.

Pada tahap pengujian akhir yang ditunjukkan pada tabel 4, berfokus pada fitur *Calendar*. Pada kalender, pengguna dapat memonitor jadwal aktivitas dan tugas dalam kurun waktu satu bulan, pada bulan saat ini. Karena kegiatan ini sistem hanya memberikan informasi maka, tujuan pengujian ini adalah untuk memastikan bahwa informasi

yang didapatkan oleh pengguna itu valid. Dibuktikan pada tabel 4 bahwa fitur kalender terjadwal memang memberikan informasi yang valid bagi pengguna, ditambah juga notifikasi reminder yang memberikan informasi bahwa sudah jadwal aktivitas sudah dekat sangat berguna bagi pengguna yang sering lupa akan jadwal mereka.

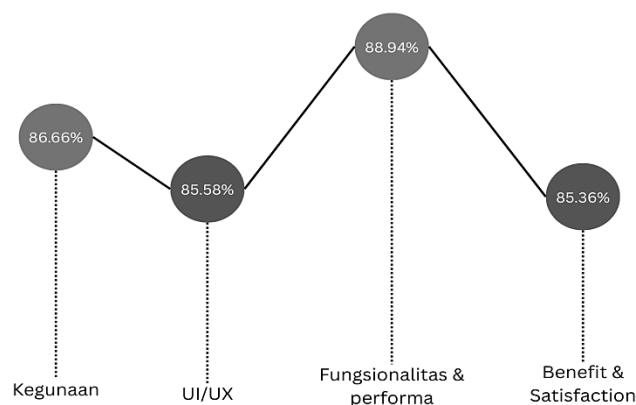
Keterkaitan antar fitur dari fitur *Activity*, *Task*, sampai *Calendar* yang diuji dengan *black box testing* menunjukkan bahwa semua fitur itu berhasil beroperasi dengan benar. Keberhasilan pada hasil semua kasus pengujian menegaskan efisiensi sistem dan keandalannya dalam mengoperasikan berbagai fitur yang telah diintegrasikan di dalamnya.

3.2.2 Mean Opinion Score (MOS)

Tabel 5. *User Acceptance* Dengan MOS

No	Pertanyaan	Kategori	SB	B	C	KB	TB
P1	Aplikasi ini mudah dipahami dan digunakan tanpa memerlukan banyak panduan.	Kegunaan	27	26	8	0	1
P2	Fitur-fitur seperti pembuatan aktivitas, catatan, dan kalender mudah diakses dan digunakan.	Kegunaan	33	24	3	1	1
P3	Proses menandai aktivitas sebagai “sudah dilakukan” atau “belum dilakukan” berjalan dengan lancar.	Kegunaan	29	27	5	0	1
P4	Tampilan antarmuka aplikasi ini menarik dan nyaman dilihat.	UI/UX	27	24	9	1	1
P5	Tata letak menu dan ikon memudahkan saya untuk menemukan fitur yang saya butuhkan.	UI/UX	32	21	7	1	1
P6	Warna dan elemen visual membantu meningkatkan fokus dan pengalaman penggunaan.	UI/UX	30	24	6	1	1
P7	Semua fitur utama (<i>activity</i> , <i>calendar</i> , <i>note</i> , <i>summary</i> , <i>insight</i>) berfungsi dengan baik sesuai harapan.	Fungsionalitas & Performa	30	26	4	1	1
P8	Aplikasi berjalan dengan cepat dan tidak mengalami kendala berarti selama digunakan.	Fungsionalitas & Performa	36	24	0	1	1
P9	Aplikasi merespons setiap aksi pengguna (seperti membuat aktivitas, mengedit, atau melihat kalender) dengan cepat tanpa jeda yang mengganggu.	Fungsionalitas & Performa	35	25	1	0	1
P10	Aplikasi ini membantu saya mengatur waktu dan aktivitas harian dengan lebih terstruktur.	<i>Benefit & Satisfaction</i>	29	23	7	2	1
P11	Aplikasi ini memberikan manfaat nyata dalam membantu saya meningkatkan produktivitas dan pengelolaan aktivitas sehari-hari.	<i>Benefit & Satisfaction</i>	25	27	6	3	1
P12	Secara keseluruhan, saya puas dengan kinerja dan manfaat aplikasi ini.	<i>Benefit & Satisfaction</i>	32	25	4	0	1

Tabel 5 yang ditampilkan memberikan analisis mendalam terhadap hasil *User Acceptance Testing (UAT)*, yang telah dilaksanakan dengan melibatkan 12 orang pekerja dan 50 orang mahasiswa sebagai responden. *MOS* menjadi metode yang digunakan untuk pengujian ini, yaitu sebuah pendekatan kuantitatif untuk mengukur perspektif pengguna dalam aspek atau kategori tertentu dari sistem. Pengujian ini memiliki 4 kategori utama: *Kegunaan*, *UI/UX*, *Fungsionalitas & Performa*, dan *Benefit & Satisfaction*, dengan setiap kategori memiliki 2-3 pertanyaan, dan total 10 pertanyaan.



Gambar 9. Hasil Mean Opinion Score

Dari hasil yang ditampilkan pada Gambar 9, menunjukkan tingkat penerimaan yang tinggi dari pengguna terhadap aplikasi Routine Master. Evaluasi berbasis pengalaman pengguna melalui *User Acceptance Test (UAT)* dengan pendekatan *Mean Opinion Score (MOS)* memungkinkan penilaian subjektif pengguna terhadap aspek kegunaan,



tampilan, dan kenyamanan sistem. Pendekatan evaluasi ini juga banyak digunakan dalam penelitian yang menekankan kualitas UI/UX sebagai faktor penting dalam keberhasilan aplikasi berbasis sistem informasi (Mahendra et al., 2025). Usability yang menilai kemudahan, kelancaran, dan aksesibilitas pengguna dengan fungsi aplikasi, mendapatkan skor 86.66%. UI/UX yang menilai tampilan, kenyamanan, visual, dan antarmuka mendapatkan skor 85.58%. Fungsionalitas dan performa yang menilai kelengkapan, fitur, fungsi, dan kinerja aplikasi mendapatkan skor 88.94% yang merupakan paling tinggi. Terakhir, *Benefit & Satisfaction* yang menilai praktis dan kepuasan dalam penggunaan mendapatkan skor 85.36%. Berdasarkan hasil evaluasi menunjukkan bahwa aplikasi memperoleh tingkat kepuasan yang cukup tinggi dan memenuhi aspek penggunaan, tampilan, fungsi, performa dan manfaat untuk digunakan sebagai alat manajemen waktu bagi pengguna.

3.3 Pembahasan

Penelitian ini menggunakan 2 pendekatan untuk menguji kinerja dari aplikasi *Routine master*, yaitu dengan pendekatan fungsional yang dilakukan dengan *black box testing*, dan pendekatan pengguna menggunakan MOS. Dari hasil pengujian fungsional pada *black box testing*. Sistem *Routine Master* mampu mengimplementasikan fungsional sesuai yang telah dirancang. Dari tabel 2 *activity*, tabel 3 *task*, dan tabel 4 *calendar* dengan status *passed*. Dengan hasil *black box testing* tersebut dapat dikaitkan dengan penggunaan metode *waterfall* dalam pengembangannya. Dimana metode *waterfall* membantu dalam pendefinisian yang jelas pada tahap awal, sehingga dapat disimpulkan bahwa metode pengembangan *waterfall* meminimalisir adanya kesalahan pada aplikasi, hal ini selaras dengan pernyataan Panjaitan et al. (2025) yang menyatakan bahwa pendekatan terstruktur *waterfall* sangat baik untuk stabilitas sistem.

Hasil pengujian yang dilakukan dengan MOS pada Gambar 9 memberikan tingkat penerimaan yang sangat positif dengan skor yang cukup tinggi di atas 80% pada seluruh kategori. Terutama pada aspek Fungsionalitas & Performa mendapatkan skor tertinggi yaitu 88.94%. Hasil ini memvalidasi pemilihan arsitektur MVC berbasis *CodeIgniter 4* yang diterapkan terbukti efisien dalam menangani proses logika dan data. Hal ini mendukung penelitian Natasya (2024) yang menyimpulkan bahwa pemisahan logika (*Controller*) dan tampilan (*View*) berkontribusi signifikan terhadap performa sistem dan kemudahan pemeliharaan kode.

selain dari aspek Fungsionalitas dan Performa, aspek Kegunaan dan UI/UX juga mendapatkan skor yang cukup tinggi sebesar 86.66% dan 85.58% yang menunjukkan bahwa antarmuka yang diterapkan pada aplikasi *Routine Master* mudah dipahami dan nyaman digunakan oleh pengguna. hal ini juga diperkuat oleh skor yang didapatkan *Benefit & Satisfaction* sebesar 85.36%, sehingga dapat disimpulkan bahwa pengguna puas dengan aplikasi *Routine Master*. Hasil ini sejalan dengan penelitian Zhu et al., (2022) yang menunjukkan bahwa desain antarmuka dan pengalaman pengguna (*user interface* dan *user experience*) pada aplikasi *to-do list* memiliki peran penting dalam meningkatkan kenyamanan, kemudahan penggunaan, dan penerimaan pengguna. Dengan demikian, *Routine Master* menawarkan nilai tambah melalui fitur *Summary* yang menyajikan visualisasi grafik efektivitas penyelesaian tugas, sehingga mendukung pemahaman pengguna terhadap progres aktivitas secara lebih intuitif.

Berbeda dengan aplikasi *MañanaList* oleh Torino et al. (2025) yang memanfaatkan notifikasi berbasis *email* untuk pengingat tugas, *Routine Master* mengadopsi *Firebase Cloud Messaging (FCM)* untuk mengirimkan *push notification* secara *real-time* ke perangkat pengguna. Pendekatan ini lebih efektif dalam memastikan atensi pengguna terhadap jadwal mendesak tanpa perlu membuka surel, Hal ini sejalan dengan penelitian oleh Mumcu dan Çebi (2025) yang menunjukkan bahwa penggunaan *push notifications* berperan signifikan dalam meningkatkan keterlibatan dan *self-regulation* mahasiswa serta mengurangi kecenderungan *academic procrastination*. Temuan ini mengindikasikan bahwa sistem notifikasi *real-time* memiliki kontribusi penting dalam mendukung kedisiplinan dan manajemen waktu mahasiswa

4. KESIMPULAN

Penelitian ini berhasil mengembangkan dan mengimplementasikan sistem *Routine Master* sebagai solusi manajemen aktivitas rutin berbasis web yang mampu membantu pengguna, khususnya mahasiswa, dalam mengelola waktu dan aktivitas secara lebih terstruktur. Keberhasilan pengembangan sistem dibuktikan melalui pengujian fungsionalitas menggunakan *Black Box Testing*, yang menunjukkan seluruh fitur utama seperti manajemen aktivitas, catatan, kalender, dan notifikasi berjalan sesuai dengan kebutuhan sistem. Penerapan metode *Waterfall* terbukti mendukung kestabilan proses pengembangan, sementara penggunaan arsitektur Model-View-Controller (MVC) berbasis *CodeIgniter 4* memberikan efisiensi pada struktur sistem dan performa aplikasi. Hal ini diperkuat dengan hasil evaluasi pengalaman pengguna melalui *User Acceptance Test (UAT)* menggunakan metode *Mean Opinion Score (MOS)*, di mana seluruh aspek penilaian memperoleh skor di atas 80%, termasuk aspek fungsionalitas dan performa, kegunaan, UI/UX, serta manfaat dan kepuasan pengguna. Implementasi *Firebase Cloud Messaging (FCM)* sebagai notifikasi *real-time* juga memberikan kontribusi positif dalam meningkatkan kedisiplinan dan efektivitas manajemen waktu pengguna dibandingkan notifikasi berbasis email. Meskipun demikian, penelitian ini memiliki beberapa keterbatasan, antara lain keterbatasan pengalaman peneliti dalam pengembangan sistem *real-time* dan desain UI/UX, perubahan spesifikasi akibat kurangnya detail kebutuhan awal, keterbatasan performa saat sistem berada pada beban kerja tinggi, serta belum terealisasinya beberapa fitur pendukung karena keterbatasan waktu dan sumber daya. Oleh karena itu, penelitian selanjutnya disarankan untuk melakukan pengembangan lebih lanjut dengan meningkatkan optimasi dan skalabilitas



sistem, serta mengintegrasikan Machine Learning, khususnya pendekatan Time Series Forecasting, guna memprediksi pola aktivitas pengguna sehingga sistem dapat berkembang menjadi asisten pribadi cerdas yang lebih adaptif, personal, dan efektif dalam mendukung perencanaan aktivitas ke depan.

REFERENCES

- Cuevas-Caravaca, E., Sánchez-Romero, E. I., & Antón-Ruiz, J. A. (2024). Academic Burnout, Personality, and Academic Variables in University Students. *European Journal of Investigation in Health, Psychology and Education*, 14(6), 1561–1571. <https://doi.org/10.3390/ejihpe14060103>
- Febrian, R., Apriliani, A. N., Fauziah, S. N., & Mutia, A. A. (2025). Pentingnya Manajemen Waktu bagi Mahasiswa Semester Akhir di Tengah Tugas Akhir. *Jurnal Ilmu Pendidikan Dan Kearifan Lokal*, 3(2), 294–305. <https://doi.org/10.52152/801841>
- Imani, Z. S., Pamungkas, A. F. A., & Jamaluddin, M. (2024). Efektivitas Teknik To-Do List terhadap Time Management Mahasiswa Psikologi UIN Malang dalam Belajar. *Jurnal Psikologi*, 1(4), 9. <https://doi.org/10.47134/pjp.v1i4.3136>
- Krisna, W., Muhammad, H. J., & Ambadar, N. (2022). Rancang Bangun Sistem Informasi Akademik Menggunakan Framework Codeigniter Pada universitas Muhammadiyah Purworejo. *Jurnal Sistem Cerdas*, 5(2), 107–116. <https://doi.org/10.37396/jsc.v5i2.187>
- Mahendra, B. A., Supriyanto, C., Paramita, C., Safar, N. Z. B. M., & Dewi, I. N. (2025). Development of a Smartphone-Based Cataract Detection System Using YOLOv10x and Ionic Framework with a UI/UX Centric Approach. *2025 International Conference on Smart Computing, IoT and Machine Learning (SIML)*, 1–5. <https://doi.org/10.1109/SIML65326.2025.11081150>
- Maulana, A., & Syakarofath, N. A. (2025). Peran academic burnout terhadap prokrastinasi akademik pada mahasiswa kota Malang. *Jurnal Integrasi Riset Psikologi*, 3(2), 21–34. <https://doi.org/10.26486/intensi.v3i2.4593>
- Natasya, A. Rizky., & Mukhlis, I. Ramadhani. (2024). Sistem Informasi Pemesanan Tiket Wisata Kota Surabaya Berbasis Web Menggunakan Metode Model View Controller. *Informatech: Jurnal Ilmiah Informatika Dan Komputer*, 1(1), 1–9. <https://doi.org/10.69533/bfb9x126>
- Mumcu, B. B., & Çebi, A. (2025). You have a notification: the role of push notifications in shaping students' engagement, self-regulation and academic procrastination. *International Journal of Educational Technology in Higher Education*, 22(1), 36. <https://doi.org/10.1186/s41239-025-00537-x>
- Panjaitan, C. K., Ashillah, S., Suleho, F., Imelda, Y., & Perdana, A. (2025). Pengembangan Aplikasi Manajemen Tugas Mahasiswa Berbasis Web Menggunakan Metode Waterfall. *Journal of Education Transportation and Business*, 2(1), 388–396. <https://doi.org/10.57235/jetbus.v2i1.6497>
- Permana, P. T. I., & Sihanato, A. N. (2024). Implementasi Arsitektur MVC Dalam Pengembangan Aplikasi Customer Relationship Portal. *Jurnal Teknologi Informasi*, 10(1), 50–57. <https://doi.org/10.52643/jti.v10i1.2344>
- Pradana, N. A., Tasleem, F. El, & Paramita, C. (2025). Optimalisasi Layanan Informasi melalui Implementasi Chatbot QnA Berbasis ChatGPT API pada Website Coffee Shop. *Indonesian Journal on Software Engineering (IJSE)*, 11(1), 25–34. <https://doi.org/10.31294/ijse.v11i1.25890>
- Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.
- Putri, I. A. V. M., Wedashwara, I. G. P. W., & Suta, I. M. (2021). Sistem Informasi E-Arsip Kelurahan Cakranegara Utara Berbasis Website. *Jurnal Begawe Teknologi Informasi (JBegati)*, 2(2), 148–158. <https://doi.org/10.29303/jbegati.v2i2.466>
- Salam, I. A., Prihandani, K., & Purnamasari, I. (2023). Rancang Bangun Aplikasi Profit Penjualan Motor Berbasis Desktop Konsep Arsitektur Model View Controller (MVC). *Jurnal Informatika Dan Teknik Elektro Terapan*, 11(3s1). <https://doi.org/10.23960/jitet.v11i3s1.3495>
- Sallaby, A. F., & Kanedi, I. (2020). Perancangan Sistem Informasi Jadwal Dokter Menggunakan Framework Codeigniter. *JURNAL MEDIA INFOTAMA*, 16(1), 48–53. <https://doi.org/10.37676/jmi.v16i1.1121>
- Tabrani, Muhamad., Suhardi, & Hananda, Priyandaru. (2021). Sistem Informasi Manajemen Berbasis Website Pada Unl Studio Dengan Menggunakan Framework Codeigniter. *Jurnal Ilmiah M-Progress*, 11(1), 13–21. <https://doi.org/10.35968/m-pu.v11i1.598>
- Torino, M. F., Fabro, M. N., Pastrana, C. K. M., & Gatdula, Ma. M. V. (2025). MañanaList App: A Mobile To-Do List Application for Efficient and Personalized Task Management. *International Journal of Research and Innovation in Social Science*, IX(IX), 2507–2523. <https://doi.org/10.47772/IJRISS.2025.909000215>
- Waluyo, D. E., Paramita, C., Kinasih, H. W., Pergiwati, D., & Rafrastara, F. A. (2024). Aplikasi Prediksi IHSG Berbasis Web Dengan Integrasi Multi-Algoritma. *Jurnal Informatika: Jurnal Pengembangan IT*, 9(2), 121–129. <https://doi.org/10.30591/jpit.v9i2.6193>
- Zhu, D., Wang, D., Huang, R., Jing, Y., Qiao, L., & Liu, W. (2022). User Interface (UI) Design and User Experience Questionnaire (UEQ) Evaluation of a To-Do List Mobile Application to Support Day-To-Day Life of Older Adults. *Healthcare*, 10(10), 2068. <https://doi.org/10.3390/healthcare10102068>
- Zulfahmi Syahputra, Rian Farta Wijaya, Khairul, & Ayu Husniyyah. (2025). Pengembangan Website Organisasi APTIKOM SUMUT Menggunakan Framework CodeIgniter dengan Pendekatan MVC untuk Optimalisasi



TIN: Terapan Informatika Nusantara

Vol 6, No 9, February 2026, page 1742-1755

ISSN 2722-7987 (Media Online)

Website <https://ejournal.seminar-id.com/index.php/tin>

DOI 10.47065/tin.v6i9.8946

Penyebaran Informasi. *Jurnal Komputer Teknologi Informasi Sistem Informasi (JUKTISI)*, 4(2), 1454–1460.
<https://doi.org/10.62712/juktisi.v4i2.703>