Jurnal Publikasi Ilmu Komputer dan Multimedia

E-ISSN: 2808-8999
P-ISSN: 2808-9375

Analisis dan Perancangan Sistem Informasi Manajemen Tugas Multi-Peran Berbasis Web

M. Iqbal Alifudin

Teknologi Informasi Universitas Bina Sarana Informatika; email : m.mif@bsi.ac.id
Dewi Sartika No.289 4, Cawang, Kec. Kramat jati, Kota Jakarta Timur, Daerah Khusus Ibukota Jakarta 13630

Abstract: The development of information technology has encouraged various sectors to transform into digital systems, including in task management and team coordination. Many organizations still face challenges in monitoring work progress, distributing tasks, and coordinating between roles involving multiple parties such as Managers, Workers, and Reviewers. This study aims to design and develop a Web-Based Multi-Role Task Management Information System that facilitates the workflow from task creation and execution to review and approval processes. The system was developed using the Laravel framework with the waterfall development methodology. The main features include task management, status tracking (to-do, in-progress, revision, done, and approved), role management, notifications, and role-specific analytical dashboards. This system is expected to serve as an integrated solution to improve coordination efficiency and task monitoring across various types of organizations and can be further developed to support integration with other collaboration platforms.

Keywords: Task Management; Information System; Laravel; Multi-Role;

Abstrak: Perkembangan teknologi informasi mendorong berbagai bidang untuk bertransformasi ke sistem digital, termasuk dalam pengelolaan tugas dan koordinasi tim. Banyak organisasi masih menghadapi kendala dalam pemantauan progres pekerjaan, distribusi tugas, dan koordinasi antar peran yang melibatkan beberapa pihak seperti Manager, Worker, dan Reviewer. Penelitian ini bertujuan untuk merancang dan membangun Analisis dan Perancangan Sistem Informasi Manajemen Tugas Multi-Peran Berbasis Web yang memfasilitasi alur kerja mulai dari pembuatan tugas, pelaksanaan, hingga proses peninjauan dan persetujuan. Sistem dikembangkan menggunakan framework Laravel dengan metode pengembangan *waterfall*. Fitur utama yang dikembangkan meliputi manajemen tugas, pelacakan status (to-do, in-progress, revision, done, dan approved), manajemen peran, notifikasi, serta dashboard analitik yang disesuaikan untuk masing-masing peran. Sistem ini diharapkan menjadi solusi terintegrasi untuk meningkatkan efisiensi koordinasi dan pemantauan tugas pada berbagai jenis organisasi serta dapat dikembangkan lebih lanjut untuk mendukung integrasi dengan platform kolaborasi lainnya.

Kata kunci: Manajemen Tugas; Sistem Informasi; Laravel; Multi-Peran;

Nakah Masuk: 13 Agustus 2025
Direvisi: 18 Agustus 2025
Diterima: 20 Agustus 2025
Diterbitkan: 22 Agustus 2025
Ver.Sekarang: 30 September 2025

1. Pendahuluan

Perkembangan teknologi informasi telah mendorong transformasi digital dalam manajemen tugas dan koordinasi tim. Sistem informasi berbasis web terbukti meningkatkan produktivitas melalui kebaruan akses data secara real-time dan penyatuan informasi dalam satu platform [1]. Dalam organisasi dengan struktur multi-peran—seperti Manager, Worker, Reviewer—koordinasi menjadi lebih kompleks. Sistem yang terintegrasi diperlukan agar alur kerja seperti pembuatan tugas, pemantauan progres, hingga peninjauan dan persetujuan dapat berjalan secara efisien dan terdokumentasi [2].

Pendekatan manual melalui spreadsheet, email, atau aplikasi chat sering kali menyebabkan keterlambatan komunikasi, miskomunikasi, dan duplikasi tugas, yang menghambat efektivitas tim [3]. Penelitian menunjukkan bahwa sistem manajemen tugas berbasis web dengan fitur notifikasi otomatis dan pelacakan tugas mampu meningkatkan efisiensi tim dan pengendalian pekerjaan [4], [5]. Misalnya, sistem berbasis web yang dirancang dengan MVC dan notifikasi email terbukti membantu koordinasi dalam konteks pendidikan, terutama untuk tugas kelompok [6].

Lebih lanjut, penelitian di lingkungan korporat mencatat bahwa penerapan sistem manajemen tugas berbasis web memiliki dampak signifikan terhadap kinerja karyawan, dengan peningkatan yang nyata setelah integrasi sistem [7], [8]. Selain itu, dalam sistem e-health berbasis Laravel, framework ini terbukti efektif dalam meningkatkan performa aplikasi seperti waktu respons dan stabilitas melalui caching dan manajemen antrian [9], [10].

Metodologi pengembangan waterfall memberikan alur sistematis mulai dari analisis, desain, implementasi, pengujian, hingga pemeliharaan — sangat cocok untuk proyek dengan ruang lingkup yang terdefinisi sejak awal [11], [12]. Sementara itu, pilihan Laravel sebagai framework mendukung struktur implementasi yang modular, aman, dan teruji di berbagai bidang pengembangan aplikasi web modern [13].

Dengan sistem ini, diharapkan koordinasi antar-peran menjadi lebih responsif, miskomunikasi berkurang, dan pemantauan progres tugas dapat dilakukan secara real-time. Penelitian ini diharapkan memberi kontribusi sebagai model bagi organisasi yang membutuhkan solusi manajemen tugas yang terstruktur dan adaptif.

2. Bahan dan Metode

Langkah awal dalam penelitian ini adalah melakukan pengumpulan data untuk memahami permasalahan yang ada dan merancang solusi sistem yang tepat. Proses pengumpulan data dilakukan melalui:

1. Observasi
   Pengamatan langsung dilakukan pada alur kerja pengelolaan tugas yang melibatkan peran Manager, Worker, dan Reviewer. Observasi difokuskan pada proses pembuatan, distribusi, pelaksanaan, dan peninjauan tugas untuk mengidentifikasi hambatan koordinasi dan kendala teknis.
2. Wawancara
   Dilakukan sesi tanya jawab dengan pihak yang terlibat, termasuk Manager, Worker, dan Reviewer, untuk memperoleh informasi terkait kebutuhan sistem, ekspektasi fitur, serta masalah yang sering terjadi dalam proses kolaborasi dan monitoring tugas.
3. Studi Pustaka
   Studi literatur dilakukan untuk memperoleh landasan teori mengenai sistem informasi manajemen tugas, manajemen peran, serta metodologi pengembangan perangkat lunak. Referensi diperoleh dari buku, artikel ilmiah, dan publikasi daring yang relevan.

Model pengembangan yang digunakan dalam penelitian ini adalah metode waterfall, yang dikenal sebagai pendekatan sekuensial dan sistematis, di mana setiap tahapan harus diselesaikan sebelum melanjutkan ke tahap berikutnya [14]. Model ini dipilih karena sesuai untuk proyek dengan kebutuhan yang telah terdefinisi dengan baik sejak awal [15]. Tahapan metode waterfall meliputi:

1. Analisis Kebutuhan
   Pada tahap ini dilakukan identifikasi terhadap kebutuhan fungsional dan non-fungsional sistem. Hasil analisis menunjukkan bahwa diperlukan sistem yang mampu mengelola status tugas (to-do, in-progress, revision, done, approved, dan cancelled), mengatur peran pengguna, mengirim notifikasi, dan menyediakan dashboard analitik untuk masing-masing peran.

2. Desain Sistem
   Desain sistem dibuat menggunakan Unified Modeling Language (UML) yang meliputi use case diagram, activity diagram, dan sequence diagram untuk memodelkan perilaku sistem, serta class diagram untuk memodelkan struktur sistem [16]. Perancangan basis data dilakukan dengan Entity Relationship Diagram (ERD) menggunakan MySQL.
3. Pembangunan Sistem (Code Generation)
   Implementasi sistem dilakukan dengan bahasa pemrograman PHP menggunakan framework Laravel, yang mendukung arsitektur MVC (Model-View-Controller) untuk memisahkan logika bisnis, antarmuka pengguna, dan pengelolaan data [17].
4. Pengujian (Testing)
   Pengujian dilakukan dengan metode black box testing untuk memastikan setiap fitur berfungsi sesuai kebutuhan pengguna tanpa memeriksa kode sumber secara langsung [18].
5. Pemeliharaan (Support)
   Tahap ini meliputi perbaikan bug, pembaruan fitur, serta penyesuaian sistem agar tetap relevan dengan kebutuhan organisasi.

3. Hasil dan Pembahasan

Pengembangan sistem informasi ini dilakukan untuk menjawab permasalahan koordinasi, distribusi pekerjaan, dan pemantauan progres yang sebelumnya dilakukan secara manual dan terpisah di berbagai media. Sistem yang dibangun menyediakan alur kerja terintegrasi mulai dari pembuatan tugas oleh Manager, pelaksanaan oleh Worker, hingga peninjauan hasil oleh Reviewer. Pada bagian ini dipaparkan hasil rancangan dan implementasi sistem yang telah dikembangkan, meliputi analisa kebutuhan, desain sistem, implementasi fitur, serta pengujian fungsionalitas. Tahapan ini bertujuan memastikan bahwa sistem yang dibangun mampu memenuhi kebutuhan semua peran pengguna secara optimal dan mendukung efektivitas proses kerja tim.

3.1. Analisa Kebutuhan

Berdasarkan hasil analisis terhadap proses kerja sebelumnya dan kebutuhan pengguna, diperoleh spesifikasi kebutuhan sistem sebagai berikut:

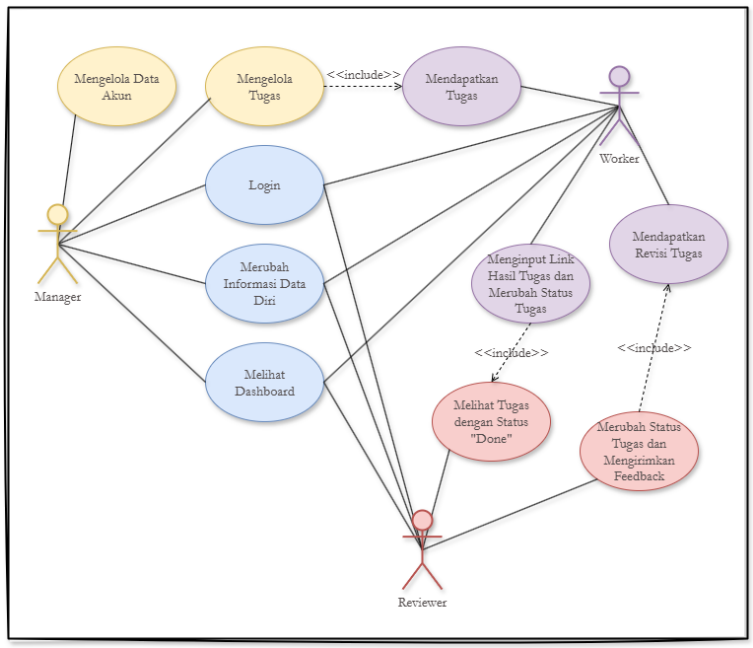
1. Kebutuhan Umum (Semua Pengguna)
   - Semua pengguna dapat melakukan login dengan akun yang terdaftar.
   - Semua pengguna dapat memperbarui profil akun masing-masing.
   - Semua pengguna dapat melihat daftar tugas yang relevan sesuai dengan perannya.
   - Semua pengguna mendapatkan notifikasi ketika ada pembaruan terkait tugas (tugas baru, perubahan status, atau komentar).
2. Kebutuhan untuk Role Manager
   - Manager dapat membuat tugas baru dengan mengisi judul, deskripsi, prioritas, penanggung jawab (Worker), dan tenggat waktu.
   - Manager dapat mengedit atau menghapus tugas yang telah dibuat.
   - Manager dapat menetapkan atau mengganti Worker untuk suatu tugas.
   - Manager dapat memantau perkembangan status tugas (to-do, in-progress, revision, done, approved).
   - Manager dapat melihat dan mengunduh laporan kinerja tim melalui dashboard analitik.
3. Kebutuhan untuk Role Worker
   - Worker dapat melihat daftar tugas yang diberikan kepadanya.
   - Worker dapat mengubah status tugas menjadi in-progress saat mulai mengerjakan, done saat selesai.
   - Worker dapat mengunggah link atau file hasil pekerjaan ke sistem.
   - Worker dapat membaca komentar atau catatan dari Reviewer.
   - Worker dapat melihat riwayat perubahan status setiap tugas.

4. Kebutuhan untuk Role Reviewer
   - Reviewer dapat melihat daftar tugas yang harus direview.
   - Reviewer dapat membuka detail tugas, memeriksa hasil pekerjaan Worker, dan memberikan catatan atau komentar.
   - Reviewer dapat mengubah status tugas menjadi revision jika perlu perbaikan, atau approved jika pekerjaan sudah sesuai.
   - Reviewer dapat memantau riwayat revisi tugas.

3.2. Desain Sistem

Pada tahap desain, dilakukan pemodelan sistem untuk menggambarkan alur kerja, interaksi antar pengguna, serta struktur data yang digunakan. Pemodelan ini mencakup Use Case Diagram, Activity Diagram, Entity Relationship Diagram (ERD), Class Diagram, Sequence Diagram, Component Diagram, dan Deployment Diagram.

1. Use Case Diagram
   Use Case Diagram digunakan untuk memvisualisasikan fungsi-fungsi yang ada pada sistem, serta menetapkan aktor (Manager, Worker, Reviewer, dan Sistem) yang berwenang menggunakan fungsi tersebut.



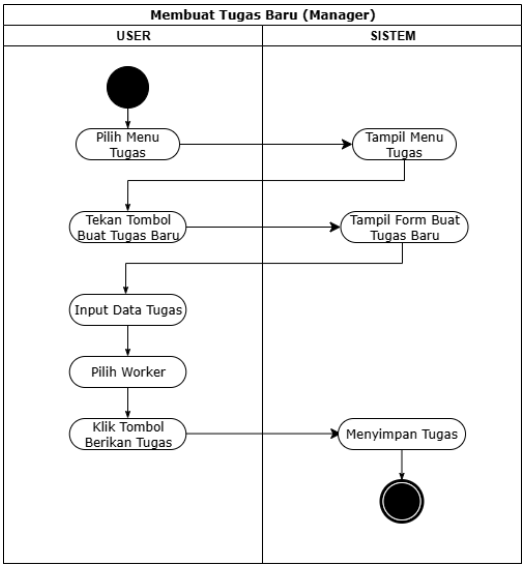
Gambar 1. Use Case Diagram

Manager memiliki hak untuk mengelola akun, mengelola tugas, memperbarui profil, melihat dashboard, dan melakukan login. Worker dapat melihat dan mengerjakan tugas, mengunggah hasil pekerjaan, menerima revisi, serta memperbarui status tugas. Reviewer berperan dalam memeriksa tugas yang telah selesai, memberikan umpan balik, dan mengubah status menjadi revisi atau disetujui.

Relasi <<include>> menunjukkan bahwa beberapa proses selalu melibatkan langkah tertentu, seperti pengelolaan tugas yang mencakup pengambilan tugas atau peninjauan hasil pekerjaan.

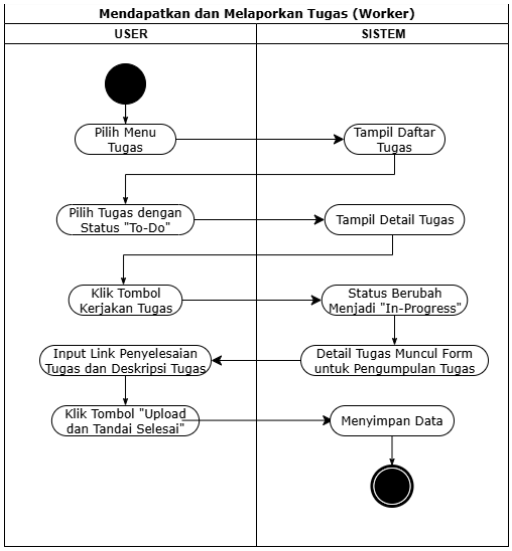
2. Activity Diagram
   Activity diagram digunakan untuk menggambarkan urutan aktivitas dan interaksi antara aktor dengan sistem dalam proses manajemen tugas multi-peran. Pada penelitian ini terdapat tiga aktor utama, yaitu Manager, Worker, dan Reviewer, yang masing-masing memiliki alur aktivitas berbeda sesuai tanggung jawabnya. Pada proses pembuatan tugas, Manager memilih menu tugas, mengisi form pembuatan tugas baru beserta detail seperti

judul, deskripsi, deadline, prioritas, dan penanggung jawab, kemudian memilih Worker yang akan mengerjakan dan menyimpan data sehingga tugas tercatat dalam sistem.



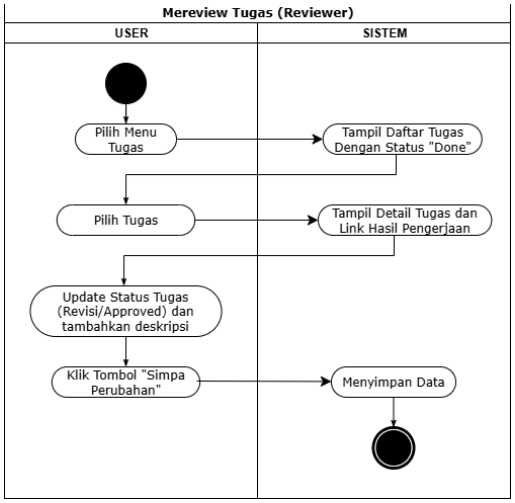
Gambar 2. Activity Diagram Membuat Tugas Baru (Manager)

Selanjutnya, Worker dapat melihat daftar tugas yang diberikan oleh Manager, memilih tugas dengan status *To-Do*, memulai pekerjaan sehingga status berubah menjadi *In-Progress*, lalu mengunggah hasil pekerjaan melalui link penyelesaian dan menandai tugas sebagai *Done*.



Gambar 3. Activity Diagram Mendapatkan dan Melaporkan Tugas (Worker)

Setelah itu, Reviewer mengakses daftar tugas dengan status *Done*, memeriksa detail dan hasil pekerjaan, lalu memutuskan apakah pekerjaan tersebut memerlukan revisi atau dapat langsung disetujui (*Approved*); jika revisi diperlukan, Reviewer memberikan catatan dan mengubah status tugas menjadi *Revision*.

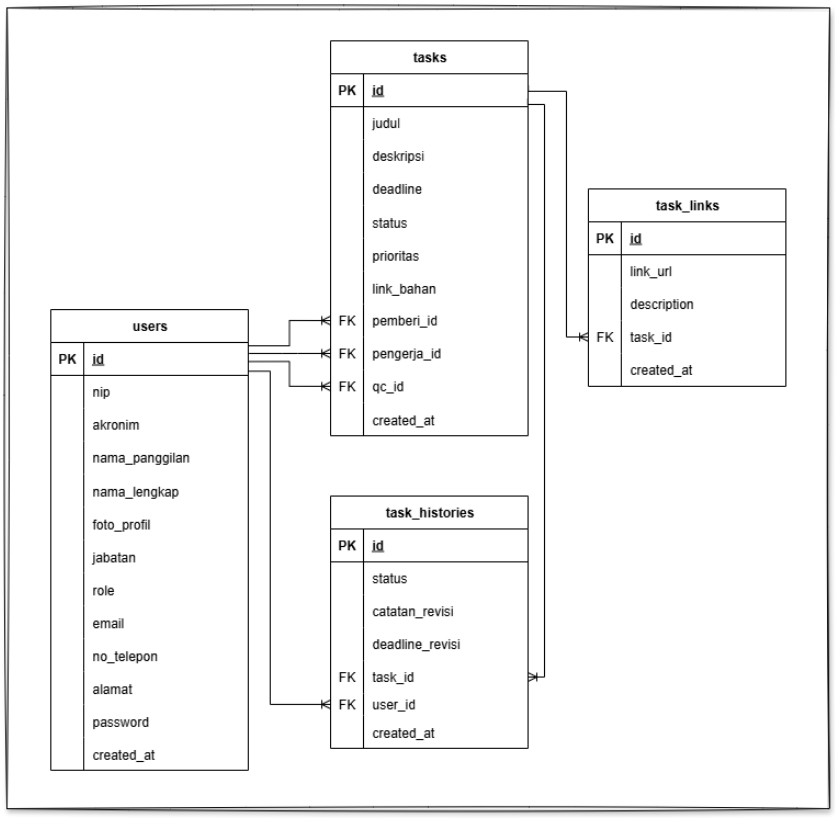


Gambar 4. Activity Diagram Mereview Tugas (Worker)

Dengan demikian, activity diagram ini menggambarkan secara menyeluruh alur kerja kolaboratif antara ketiga peran tersebut dari awal pembuatan tugas hingga tahap akhir persetujuan.

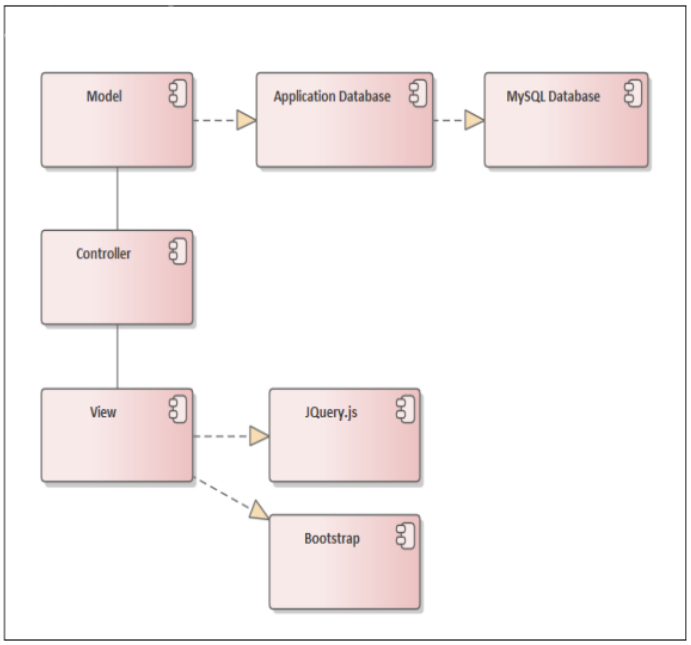
3. Entity Relationship Diagram

Entity Relationship Diagram (ERD) pada sistem ini memodelkan hubungan antara tiga entitas utama, yaitu users, tasks, task_histories, dan task_links. Tabel users menyimpan data pengguna dengan peran manager, worker, dan reviewer. Tabel tasks mencatat informasi tugas yang dibuat oleh manager, dikerjakan oleh worker, dan direview oleh reviewer. Riwayat perubahan status tugas tersimpan pada task_histories, sedangkan task_links menyimpan tautan file atau bahan pendukung tugas. Relasi antar tabel menggunakan foreign key sehingga data antar entitas tetap konsisten dan terintegrasi.



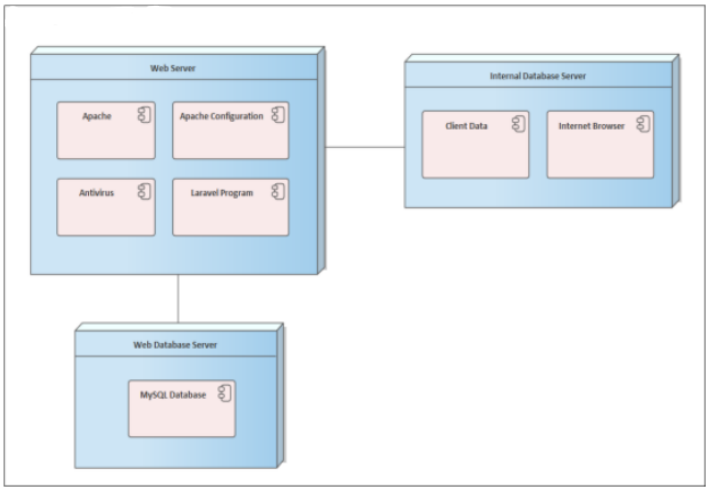
Gambar 5. Entity Relationship Diagram

4. Component Diagram
Component Diagram pada sistem ini menggambarkan hubungan antar komponen dalam arsitektur aplikasi berbasis Laravel. Komponen Model berinteraksi dengan Application Database yang kemudian terhubung ke MySQL Database untuk pengelolaan data. Controller bertugas mengatur alur logika aplikasi dan menjadi penghubung antara Model dan View. Komponen View menangani tampilan antarmuka pengguna dan memanfaatkan JQuery.js untuk interaksi dinamis serta Bootstrap untuk pengaturan desain responsif. Diagram ini menunjukkan bahwa setiap komponen saling terintegrasi untuk mendukung fungsionalitas sistem secara efisien dan terstruktur.



Gambar 6. Component Diagram

5. Deployment Diagram
Deployment Diagram pada sistem ini menjelaskan arsitektur fisik dan lingkungan eksekusi aplikasi. Sistem dijalankan pada Web Server yang terdiri dari komponen Apache, Apache Configuration, Antivirus, dan Laravel Program. Web Server terhubung dengan Web Database Server yang mengelola MySQL Database sebagai penyimpanan utama. Selain itu, terdapat Internal Database Server yang memuat Client Data dan Internet Browser sebagai media akses sistem oleh pengguna. Diagram ini menunjukkan alur distribusi dan interaksi antar server untuk memastikan aplikasi dapat berjalan stabil, aman, dan terintegrasi dengan baik.



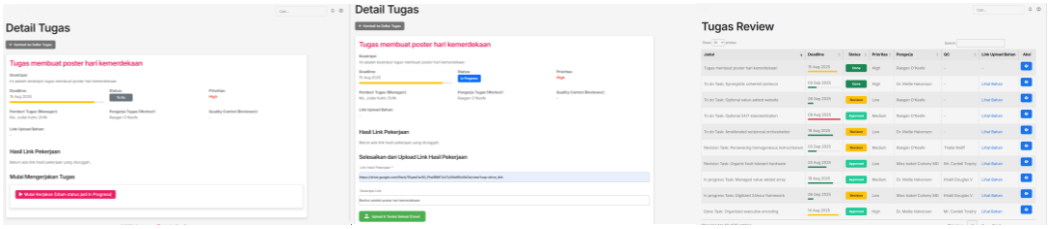
Gambar 6. Component Diagram

3.3. Code Generation

Proses pembuatan tugas pada sistem dimulai dari halaman Daftar Tugas milik Manager. Pada halaman ini, Manager dapat melihat seluruh tugas yang ada beserta status, prioritas, dan siapa pengerjanya. Untuk membuat tugas baru, Manager menekan tombol "Buat Tugas Baru" yang terletak di bagian kanan atas halaman. Manager wajib mengisi seluruh field yang memiliki tanda * sebagai tanda wajib. Setelah semua terisi, Manager menekan tombol "Berikan Tugas", dan data akan tersimpan di database. Selanjutnya, tugas otomatis akan muncul di halaman Daftar Tugas milik Manager dan juga pada halaman Daftar Tugas Saya milik Worker yang ditugaskan.

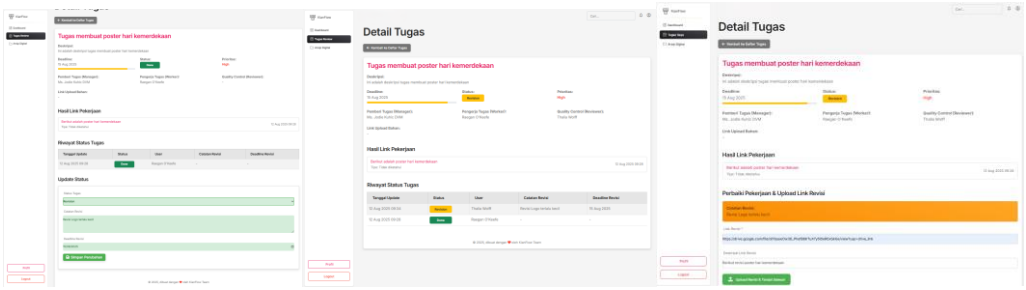
Gambar 7. Membuat Tugas Baru

Setelah tugas berhasil dibuat oleh Manager dan muncul di halaman Daftar Tugas Saya milik Worker, pengguna dengan peran Worker dapat membuka halaman Detail Tugas. Pada tahap awal, status tugas berada di posisi "To Do". Worker harus menekan tombol "Mulai Kerjakan (Ubah status jadi In Progress)" untuk memulai pekerjaan. Setelah status berubah menjadi "In Progress", sistem menampilkan form untuk mengunggah hasil pekerjaan. Setelah Worker menekan tombol "Upload & Tandai Selesai (Done)", status tugas berubah menjadi "Done" dan otomatis masuk ke daftar Tugas Review milik Reviewer.



Gambar 8. Mengumpulkan Tugas

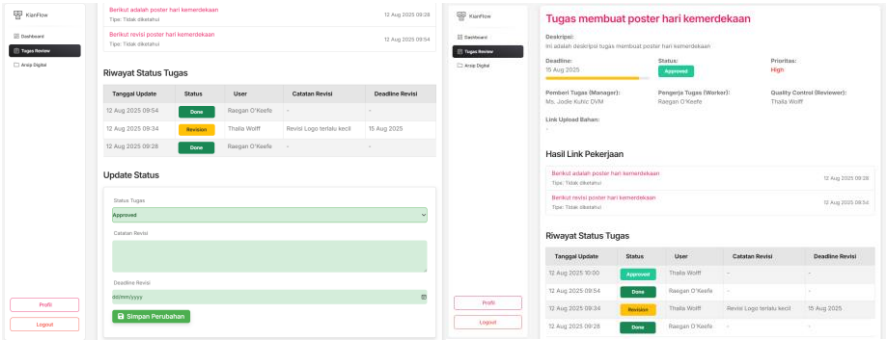
Setelah Worker mengumpulkan tugas dan statusnya menjadi "Done", tugas tersebut masuk ke halaman Tugas Review milik Reviewer. Pada tahap ini, Reviewer membuka halaman detail tugas untuk melakukan pengecekan hasil pekerjaan. Jika ditemukan kekurangan, Reviewer dapat mengubah status menjadi "Revision" melalui form Update Status. Setelah disimpan, sistem mengubah status tugas menjadi "Revision" dan mencatat perubahan tersebut pada riwayat status. Status revisi ini akan otomatis mengirimkan tugas kembali ke Worker. Pada tampilan Worker,bagian Perbaiki Pekerjaan & Upload Link Revisi akan muncul, menampilkan Catatan Revisi dari Reviewer serta form unggah link revisi yang harus diisi Worker. Setelah Worker mengunggah revisi, status akan berubah kembali menjadi "Done" dan masuk lagi ke tahap review untuk pemeriksaan akhir.



Gambar 9. Mereview Tugas

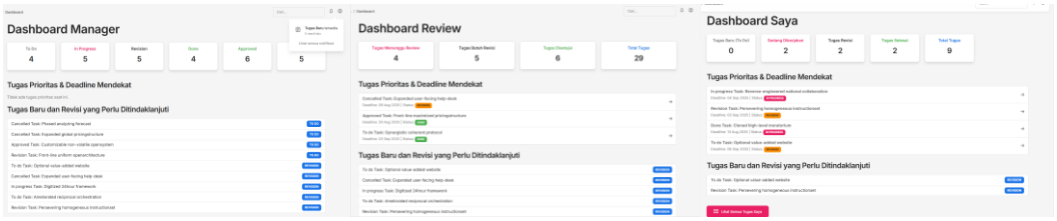
Setelah Worker mengunggah revisi sesuai catatan Reviewer, sistem akan mengubah status tugas menjadi "Done" dan mengirimkannya kembali ke halaman Tugas Review milik Reviewer. Reviewer dapat membuka detail tugas untuk memeriksa hasil revisi yang diunggah Worker. Jika revisi sudah

sesuai dengan permintaan sebelumnya, Reviewer dapat mengubah status menjadi "Approved" melalui form Update Status. Pada tahap ini, Reviewer tidak perlu mengisi Catatan Revisi atau Deadline Revisi karena tugas dianggap selesai. Setelah status disimpan, sistem akan mengubah status tugas menjadi "Approved" (Gambar 2), menandakan bahwa pekerjaan telah diterima dan disetujui sepenuhnya. Tugas ini akan tersimpan dalam riwayat dengan urutan status yang lengkap, mulai dari "To Do" → "In Progress" → "Done" → "Revision" → "Done" (Revisi) → "Approved". Dengan alur ini, proses pengelolaan tugas menjadi transparan, terdokumentasi dengan baik, dan meminimalkan miskomunikasi antara Worker dan Reviewer.



Gambar 10. Mengirim Revisi Tugas

Sistem ini memiliki tiga jenis dashboard sesuai peran pengguna, yaitu Manager, Reviewer, dan Worker, yang masing-masing menampilkan ringkasan status tugas dan daftar pekerjaan yang perlu ditindaklanjuti. Dashboard Manager menampilkan jumlah tugas per status (To Do, In Progress, Revision, Done, Approved) serta daftar tugas prioritas dan revisi yang harus dipantau. Dashboard Reviewer menampilkan jumlah tugas menunggu review, butuh revisi, dan disetujui, disertai daftar tugas prioritas dan revisi yang harus diperiksa. Sementara itu, Dashboard Worker menampilkan jumlah tugas To Do, Sedang Dikerjakan, Revisi, dan Selesai, lengkap dengan daftar tugas prioritas dan revisi yang perlu segera dikerjakan. Tampilan dashboard ini memastikan setiap peran memiliki visibilitas jelas terhadap progres pekerjaannya sehingga koordinasi tim berjalan efisien.



Gambar 11. Dashboard

3.4. Testing

Pengujian sistem manajemen tugas ini dilakukan menggunakan metode black box testing dengan cara mencoba langsung seluruh fitur sesuai alur kerja (flow) yang telah dibangun. Tujuan pengujian ini adalah untuk memastikan setiap fungsi berjalan sesuai spesifikasi tanpa memeriksa kode program secara langsung. Proses pengujian dilakukan mulai dari pembuatan tugas oleh Manager, pengerjaan tugas oleh Worker, proses review oleh Reviewer, hingga monitoring melalui dashboard. Setiap skenario diuji dengan berbagai kondisi, baik input kosong, input tidak lengkap, maupun input benar, untuk memastikan sistem mampu memberikan output yang sesuai harapan.

Tabel 1. Hasil Pengujian Black Box – Form Buat Tugas (Manager)

Skenario Pengajuan	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
Mengosongkan semua form	Judul: (kosong), Deskripsi: (kosong), Deadline: (kosong), Prioritas: (kosong), Pengerja: (kosong)	Sistem menolak dan menampilkan pesan error bahwa semua field wajib diisi	Sesuai harapan	Valid
Mengisi sebagian form	Judul: Ada, Deskripsi: (kosong), lainnya terisi sebagian	Sistem menolak dan memberi pesan bahwa data belum lengkap	Sesuai harapan	Valid
Mengisi seluruh form dengan benar	Judul: "Tugas Poster", Deskripsi: "Poster Hari Kemerdekaan", Deadline: 15/08/2025, Prioritas: High, Pengerja: Raegan	Sistem menyimpan tugas dan menampilkannya di daftar tugas	Sesuai harapan	Valid

Tabel 2. Hasil Pengujian Black Box – Pengerjaan Tugas (Worker)

Skenario Pengajuan	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
Mengubah status ke In Progress	Klik tombol "Mulai Mengerjakan"	Status berubah menjadi "In Progress"	Sesuai harapan	Valid
Mengirim tugas tanpa link hasil	Status In Progress, form link kosong	Sistem menolak dan memberi pesan bahwa link wajib diisi	Sesuai harapan	Valid
Mengirim tugas dengan link benar	Link: https://drive.google.com/..., deskripsi terisi	Status berubah menjadi "Done" dan data tersimpan	Sesuai harapan	Valid

Tabel 3. Hasil Pengujian Black Box – Review Tugas (Reviewer)

Skenario Pengajuan	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
Memberikan status revisi tanpa catatan revisi	Status: Revision, Catatan: (kosong)	Sistem menolak dan meminta catatan revisi diisi	Sesuai harapan	Valid
Memberikan status revisi dengan catatan	Status: Revision, Catatan: "Logo terlalu kecil", Deadline revisi: 15/08/2025	Status berubah menjadi "Revision" dan tugas kembali ke Worker	Sesuai harapan	Valid
Menyetujui tugas	Status: Approved, tanpa catatan revisi	Status berubah menjadi "Approved" dan proses selesai	Sesuai harapan	Valid

3.5. Support

Apabila terjadi kesalahan, error, atau gangguan pada sistem manajemen tugas, pengelola sistem bertanggung jawab untuk segera melakukan pengecekan, analisis, dan perbaikan agar proses kerja tidak terganggu. Pengelola juga harus memastikan database tetap aman dan melakukan backup data secara berkala untuk menghindari kehilangan informasi tugas. Selain itu, pembaruan perangkat lunak dan keamanan sistem (security patch) harus dilakukan secara rutin, termasuk pembaruan antivirus untuk mencegah serangan malware atau virus yang dapat mengganggu kinerja sistem. Tim support juga perlu menyediakan panduan penggunaan dan layanan bantuan (helpdesk) agar pengguna, baik Manager, Worker, maupun Reviewer, dapat melaporkan masalah atau mendapatkan solusi dengan cepat.

4. Kesimpulan dan Saran

4.1. Kesimpulan

Sistem manajemen tugas yang dibangun berhasil memfasilitasi alur kerja mulai dari pembuatan tugas oleh Manager, pengerjaan oleh Worker, proses review oleh Reviewer, hingga penyelesaian dan approval tugas. Fitur-fitur seperti pembaruan status, catatan revisi, pengunggahan link hasil kerja, serta dashboard khusus untuk setiap peran, membuat proses koordinasi menjadi lebih terstruktur dan transparan. Berdasarkan hasil pengujian menggunakan metode black box testing, seluruh fungsi utama berjalan sesuai harapan tanpa ditemukan bug yang signifikan. Sistem ini dapat membantu meningkatkan efisiensi kerja tim, meminimalisir miskomunikasi, dan memudahkan monitoring progres tugas.

4.2. Saran

Agar sistem dapat semakin optimal, beberapa saran pengembangan ke depan antara lain:

1. Integrasi Notifikasi Real-time melalui email atau aplikasi pesan instan, agar pengguna mendapatkan pemberitahuan langsung terkait tugas baru, revisi, atau approval.Barang kedua;
2. Fitur Lampiran File selain link, sehingga worker dapat langsung mengunggah file hasil pekerjaan.
3. Pelaporan Kinerja Otomatis yang menampilkan rekap jumlah tugas yang dibuat, dikerjakan, direvisi, dan disetujui dalam periode tertentu.
4. Penguatan Keamanan Sistem dengan autentikasi dua faktor (2FA) untuk mencegah akses tidak sah.

Referensi

[1] I. Puspasari And L. Samboteng, "The Effect Of The Task Management System On The Employee's Performance Of The Audit Board At The Head Office," Sinomics Journal | Volume, Vol. 3, No. 2, 2024, Doi: 10.54443/Sj.V3i2.317.

[2] F. N. Shah And Et Al., "Web-Based Task Management System For Improving Group Work Collaboration," International Journal Of Academic Research In Progressive Education And Development, Vol. 11, No. 3, Pp. 1669–1680, 2022, Doi: 10.6007/Ijarped/V11-I3/15188.

[3] V. Soumya, "Laravel-Based Task Management System: Design, Development, And Implementation," International Journal Of Scientific Research In Engineering And Management, Vol. 9, No. 3, Pp. 1–9, 2025, Doi: 10.55041/Ijsrem43066.

[4] A. Rahad And Others, "A Web-Based Note And Task Organizer For Efficient Personal Productivity," In Ceur Workshop Proceedings, 2024.

[5] G. Atchula And Others, "Effi-Tasker: A Smart Task Management System For Enhanced Productivity And Workflow Optimization," In International Journal Of Innovative Research In Technology (Ijirt), 2024, Pp. 1511–1512.

[6] F. N. Ahmad And K. A. Wahab, "Web-Based Student Task Management System," Engineering Agriculture Science And Technology Journal (East-J), Vol. 1, No. 1, Pp. 50–56, 2022, Doi: 10.37698/Eastj.V1i1.119.

[7] R. M. Alshowair And M. Alkhattabi, "Impact Of A Web-Based Application For Employee Performance Management (Epms) On Employee Performance: Employees' Opinions," International Journal Of U- And E-Service, Science And Technology, Vol. 8, No. 4, Pp. 37–50, Apr. 2015, Doi: 10.14257/Ijunesst.2015.8.4.05.

[8] K. Baskaran, B. Keerthana, And N. Kumar, "A Study On Effective Performance Of Management System On Employee Performance," Article In International Research Journal Of Modernization In Engineering Technology And Science, 2025, Doi: 10.56726/Irjmets65737.

[9] P. C. Dhage, R. A. Thakker, And K. K. Warhade, "Laravel Technology Based Maternal E-Healthcare Systems With Improved Response Time And Stability," International Research Journal Of Multidisciplinary Technovation, Vol. 7, No. 3, Pp. 326–344, May 2025, Doi: 10.54392/Irjmt25324.

[10] N. Saputri, Z. Dahlia, J. Teknologi Informasi Dan Komputer Politeknik Negeri Lhokseumawe, And J. Teknik Elektro Politeknik Negeri Lhokseumawe, "Journal Of Artificial Intelligence And Engineering Applications Optimizing Web-Based Survey Applications With Laravel And Cloud Computing," 2024. [Online]. Available: Https://Ioinformatic.Org/

[11] S. Pressman, Software Engineering: A Practitioner's Approach, 8th Ed. New York: Mcgraw-Hill, 2015.

[12] W. Royce, "Managing The Development Of Large Software Systems," In Proceedings Of Ieee Wescon, 1970, Pp. 1–9.

[13] A. Duggirala, "Php Laravel – A Focus On Customization And Schedule Job Management," International Journal For Research In Applied Science & Engineering Technology (Ijraset), 2024, Doi: 10.22214/Ijraset.2024.63417.

[14] S. Mudassar And A. Khan, "Waterfall Model Used In Software Development Reference: Software Requirements Engineering Waterfall Model," 2023, Doi: 10.13140/Rg.2.2.29580.69764.

[15] H. J. M. Ruël, T. Bondarouk, And S. Smink, "The Waterfall Approach And Requirement Uncertainty," In Project Management Techniques And Innovations In Information Technology, Igi Global, Pp. 49–65. Doi: 10.4018/978-1-4666-0930-3.Ch004.

[16] H. Koç, A. M. Erdoğan, Y. Barjakly, And S. Peker, "Uml Diagrams In Software Engineering Research: A Systematic Literature Review," In The 7th International Management Information Systems Conference, Basel Switzerland: Mdpi, Mar. 2021, P. 13. Doi: 10.3390/Proceedings2021074013.

[17] G. B. Santoso, T. M. Sinaga, And A. Zuhdi, "Mvc Implementation In Laravel Framework For Development Web-Based E-Commerce Applications," Intelmatics, Vol. 1, No. 1, Jan. 2021, Doi: 10.25105/Itm.V1i1.7867.

[18] S. Nidhra, "Black Box And White Box Testing Techniques - A Literature Review," International Journal Of Embedded Systems And Applications, Vol. 2, No. 2, Pp. 29–50, Jun. 2012, Doi: 10.5121/Ijesa.2012.2204.