

ANALISIS SISTEM MANAJEMEN TUGAS INTERNAL

1. LATAR BELAKANG

Saat ini, perusahaan menghadapi tantangan dalam melacak dan mengelola pekerjaan harian di berbagai tim seperti IT, HR, dan Operasional. Pengelolaan tugas yang masih dilakukan secara manual tanpa adanya sistem terpusat mengakibatkan sejumlah permasalahan, seperti kurangnya visibilitas yang jelas mengenai siapa yang bertanggung jawab atas tugas tertentu, kesulitan dalam memantau status progres tugas, serta hubungan antara tugas dan proyek yang sulit dilacak. Selain itu, manajemen mengalami hambatan dalam mendapatkan gambaran keseluruhan tentang produktivitas tim dan status proyek yang sedang berjalan. Risiko keterlambatan penyelesaian tugas pun meningkat karena tidak adanya sistem peringatan tenggat waktu. Kurangnya data terpusat juga menyulitkan evaluasi kinerja dan alokasi sumber daya secara efektif. Kondisi ini menyebabkan terjadinya inefisiensi, potensi duplikasi pekerjaan, dan berbagai tantangan dalam memastikan setiap tugas dapat diselesaikan tepat waktu.

2. TUJUAN

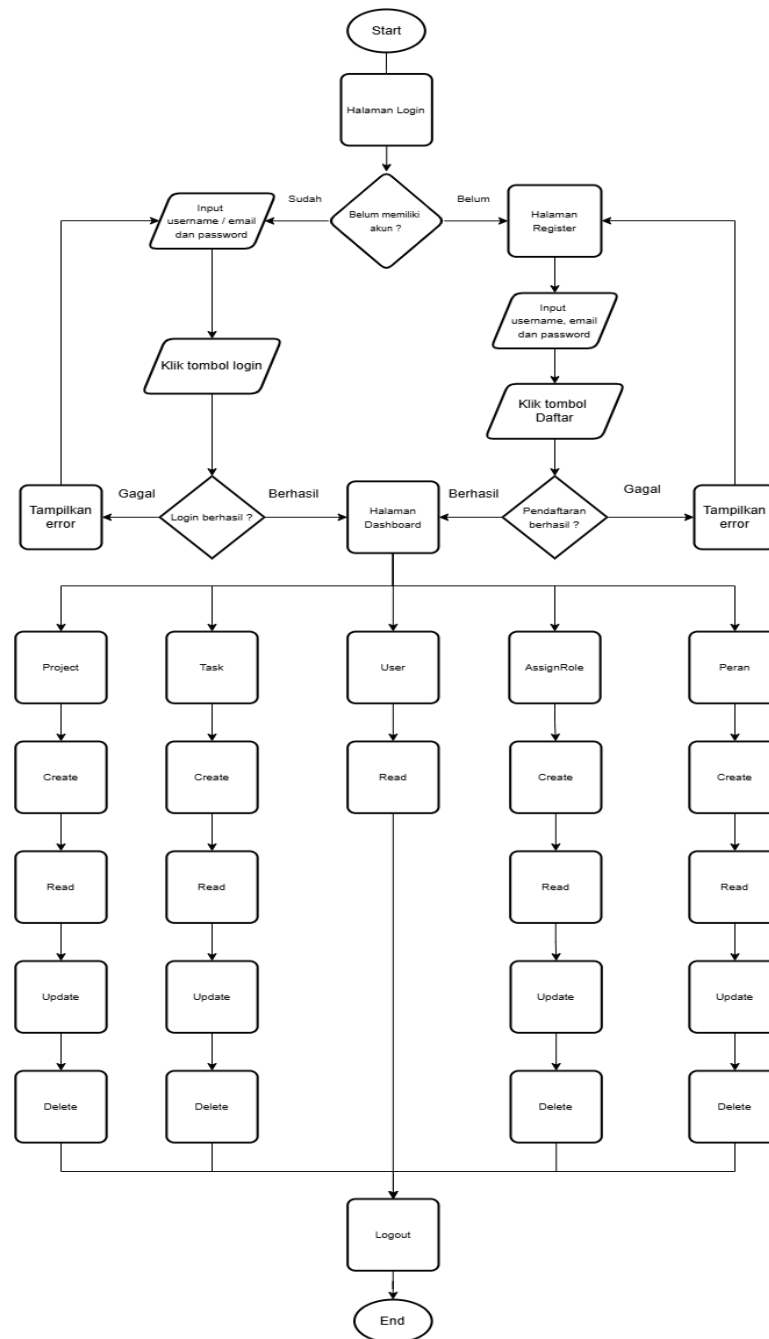
Tujuan pembangunan sistem manajemen tugas internal adalah:

1. Sentralisasi Data: Menciptakan satu sumber informasi untuk semua tugas dan proyek perusahaan
2. Peningkatan Transparansi: Memungkinkan semua pemangku kepentingan melihat status tugas secara real-time
3. Peningkatan Akuntabilitas: Menetapkan tanggung jawab yang jelas untuk setiap tugas
4. Efisiensi Pengelolaan: Memudahkan manajemen untuk memantau progress, mengalokasikan sumber daya, dan mengidentifikasi bottleneck
5. Pelaporan Terstruktur: Menyediakan laporan ringkas tentang status tugas berdasarkan proyek dan pengguna
6. Kontrol Akses: Memastikan informasi hanya dapat diakses oleh personel yang berwenang sesuai dengan peran mereka
7. Peningkatan Kolaborasi: Memfasilitasi kerja tim yang lebih baik dengan informasi yang transparan dan terpusat

Sistem ini diharapkan akan meningkatkan produktivitas, mempercepat penyelesaian proyek, dan memberikan data yang diperlukan untuk pengambilan keputusan yang lebih baik oleh manajemen.

3. FLOWCHART

Untuk mendukung pengelolaan tugas yang lebih efektif dan terstruktur, sistem manajemen tugas ini dirancang dengan alur proses utama yang mencakup tahapan login, navigasi menu, pengelolaan proyek dan tugas, manajemen user, ringkasan laporan, hingga proses logout. Alur ini memastikan setiap fungsi penting dalam pengelolaan pekerjaan harian dapat diakses dengan mudah dan sistematis. Untuk flowchart dapat dilihat pada gambar berikut.



Gambar 1. Flowchart

Flowchart di atas menggambarkan alur proses utama dalam sistem manajemen tugas:

1. Proses Login dan Autentikasi:
 - User memulai dengan login ke sistem
 - Sistem melakukan verifikasi kredensial
 - Jika gagal, tampilkan pesan error dan kembali ke halaman login
 - Jika berhasil, arahkan ke dashboard
2. Menu Utama: Setelah login, jika admin memiliki akses ke lima menu utama:
 - Manajemen Proyek
 - Manajemen Tugas
 - Manajemen User
 - Manajemen AssignRole
 - Manajemen Peran
3. Menu Utama: Setelah login, jika user memiliki akses ke tiga menu utama:
 - Manajemen Proyek
 - Manajemen Tugas
 - Manajemen User
4. Manajemen Proyek:
 - Lihat daftar proyek
 - Tambah proyek baru
 - Edit proyek yang ada
5. Manajemen Tugas:
 - Lihat daftar tugas
 - Tambah tugas baru
 - Edit status tugas (To Do, In Progress, Done)
 - Assign tugas ke user tertentu
6. Manajemen Pengguna:
 - Lihat daftar user
7. Manajemen Pemberian Peran
 - Lihat daftar Peran

- username (varchar(255), Not Null)
- email (varchar(255), Not Null)
- email_verified_at (timestamp)
- password (varchar(255), Not Null)
- remember_token (varchar(100))
- deleted_at (timestamp) - untuk soft delete
- created_at (timestamp)
- updated_at (timestamp)

2. Tasks

Tabel ini menyimpan informasi tentang tugas-tugas dengan atribut:

- id (Primary Key, char(36), Not Null) - menggunakan UUID
- title (varchar(255), Not Null)
- description (text)
- project_id (Foreign Key, char(36), Not Null) - referensi ke tabel projects
- status (tasks_status_enum, Not Null) - enumeration yang berisi status seperti "To Do", "In Progress", "Done"
- deadline (date)
- deleted_at (timestamp) - untuk soft delete
- created_at (timestamp)
- updated_at (timestamp)

3. Projects

Tabel ini menyimpan informasi tentang proyek dengan atribut:

- id (Primary Key, char(36), Not Null) - menggunakan UUID
- name (varchar(255), Not Null)
- description (text)
- start_date (date)
- end_date (date)

- created_at (timestamp)
- updated_at (timestamp)

4. Roles

Tabel ini menyimpan peran/role pengguna dalam sistem:

- id (Primary Key, bigint(20), Not Null)
- name (varchar(255), Not Null)
- guard_name (varchar(255), Not Null) - nama guard untuk Laravel authentication
- created_at (timestamp)
- updated_at (timestamp)

5. Permissions

Tabel ini menyimpan izin/hak akses dalam sistem:

- id (Primary Key, bigint(20), Not Null)
- name (varchar(255), Not Null)
- guard_name (varchar(255), Not Null)
- created_at (timestamp)
- updated_at (timestamp)

Tabel Relasi/Pivot

1. assigned_task_user

Tabel pivot untuk menghubungkan tugas dengan pengguna yang ditugaskan:

- task_id (Foreign Key, char(36), Not Null) - referensi ke tabel tasks
- user_id (Foreign Key, bigint(20), Not Null) - referensi ke tabel users

2. model_has_roles

Tabel pivot untuk menghubungkan model dengan peran (polymorphic relation):

- role_id (Foreign Key, bigint(20), Not Null) - referensi ke tabel roles
- model_type (varchar(255), Not Null) - nama model yang memiliki peran
- model_id (bigint(20), Not Null) - ID dari model yang memiliki peran

3. role_has_permissions

Tabel pivot untuk menghubungkan peran dengan izin:

- permission_id (Foreign Key, bigint(20), Not Null) - referensi ke tabel permissions
- role_id (Foreign Key, bigint(20), Not Null) - referensi ke tabel roles

Hubungan Antar Tabel

1. **Users dan Tasks:** Memiliki hubungan many-to-many melalui tabel pivot assigned_task_user. Satu pengguna dapat ditugaskan banyak tugas, dan satu tugas dapat ditugaskan ke beberapa pengguna.
2. **Projects dan Tasks:** Memiliki hubungan one-to-many. Satu proyek dapat memiliki banyak tugas.
3. **Users dan Roles:** Memiliki hubungan many-to-many melalui tabel pivot model_has_roles dengan model_type = 'User'. Satu pengguna dapat memiliki banyak peran, dan satu peran dapat dimiliki oleh banyak pengguna.
4. **Roles dan Permissions:** Memiliki hubungan many-to-many melalui tabel pivot role_has_permissions. Satu peran dapat memiliki banyak izin, dan satu izin dapat dimiliki oleh banyak peran.

Fitur Teknis

1. **Soft Delete:** Tabel users dan tasks mengimplementasikan soft delete dengan kolom deleted_at, yang memungkinkan penghapusan data tanpa benar-benar menghapusnya dari database.
2. **UUID:** Tabel projects dan tasks menggunakan UUID (char(36)) sebagai primary key, bukan incremental integer biasa.
3. **Timestamp:** Semua tabel menyimpan waktu pembuatan dan pembaruan data dengan kolom created_at dan updated_at, memudahkan pelacakan perubahan.
4. **Enum untuk Status:** Tabel tasks menggunakan tipe data enum untuk kolom status, membatasi nilai hanya pada opsi yang valid (To Do, In Progress, Done).