LAMPIRAN

Lampiran 1. Surat Balasan Observasi

MEDINA

Nomor : 001/S-Kel/MDN/III/2025
Lampiran : -
Hal : Surat Balasan Permohonan Ijin Observasi

Kepada
Yth.
Ka.Prodi S.Tr. Teknik Informatika
Politeknik Harapan Bersama Tegal
Di Tempat

Dengan hormat,

Sebelumnya kami mengucapkan terima kasih atas surat yang Anda sampaikan kepada kami, perihal Penelitian/ Observasi yang akan dilaksanakan oleh salah satu mahasiswa Anda yang bernama :

Nama : Siska Aulia Utami
NIM : 21090048
Prodi : Sarjana Terapan Teknik Informatika

Dengan ini kami dari PT. Medika Digital Nusantara memberikan izin kepada yang bersangkutan untuk melaksanakan observasi di tempat kami dengan tetap mematuhi aturan dan ketentuan yang berlaku di instansi kami. Kami berharap bahwa observasi ini dapat berjalan dengan lancar.

Adapun penelitian yang akan dilakukan berjudul "Rancang Bangun Aplikasi Manajemen Customer Service Dalam Meningkatkan Resolusi Keluhan Client Berbasis Website (PT. Medika Digital Nusantara)". Kami mendukung penelitian ini karena sejalan dengan upaya peningkatan kualitas layanan pelanggan melalui pemanfaatan teknologi.

Demikian surat balasan ini kami sampaikan. Semoga kegiatan observasi ini dapat berjalan dengan baik dan memberikan hasil yang diharapkan. Atas perhatian dan kerjasamanya, kami ucapkan terima kasih.

Hormat kami,

MEDINA

Sekar Jati Budi Sulistyowati
Human Relation
PT Medika Digital Nusantara

Jl. Puntadewa No.60 Pugeran, Maguwoharjo, Depok, Sleman, Yogyakarta 55282
Telp : 0274-4331795
Fax : 0274-4331795
Website : www.morbis.id
E-mail : info@morbis.id

Lampiran 2. Surat Izin Observasi

POLITEKNIK HARAPAN BERSAMA
The True Vocational Campus

Sarjana Terapan Teknik Informatika

Nomor : 06.03/TI.PHB/II/2025
Lampiran : -
Hal : Permohonan Izin Observasi
Kepada Yth. : **Kepala PT. Medika Digital Nusantara**
Di Tempat

Dengan hormat, mahasiswa dengan identitas berikut ini:

nama : Siska Aulia Utami
NIM : 21090048
prodi : Sarjana Terapan Teknik Informatika

Bermaksud mela0kukan penelitian untuk keperluan Skripsi dengan judul **"Perancangan Aplikasi Manajemen Customer Service Dalam Meningkatkan Resolusi Keluhan Berbasis Website (PT. Medika Digital Nusantara)"**. Kami memohon Bapak/Ibu memberikan izin kepada mahasiswa yang bersangkutan agar memperoleh data, keterangan, dan bahan yang diperlukan.

Demikian permohonan ini disampaikan, atas perhatian kami ucapkan terima kasih.

Tegal, 20 Februari 2025
Ka. Prodi S.Tr. Teknik Inormatika,

Dyah Aprilliani, S.T., M.Kom
NIPY : 09.015.225

Jl. Mataram No. 9 Kota Tegal 52143, Jawa Tengah, Indonesia.
(0283)352000
informatika@poltektegal.ac.id
poltektegal.ac.id

Lampiran 3. Surat Kesepakatan Bimbingan

1. Surat Kesepakatan Bimbingan Pembimbing I

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama	: Siska Aulia Utami
NIM	: 21090048
Program Studi	: Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama	: Dega Surono Wibowo, S.T., M.Kom.
Status	: Dosen
NIDN	: 0607108202
Jabatan Fungsional	: Lektor
Pangkat/Golongan	: Penata /III/C

Pada hari ini Jum'at tanggal 25 April 2025 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Skripsi Pihak Pertama dengan syarat melakukan bimbingan 1 kali dalam seminggu atau setidaktidaknya 3 kali bimbingan dalam satu bulan (dengan progress), apabila saya tidak memenuhi syarat tersebut maka saya tidak berhak meminta surat rekomendasi mengikuti sidang skripsi. Dan saya juga berjanji akan memenuhi persyaratan tersebut dan menyelesaikan skripsi baik produk maupun laporan tepat waktu.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

Tegal, 25 April 2025

Pihak Pertama	Pihak Kedua
Siska Aulia Utami	Dega Surono Wibowo, S.T., M.Kom. NIPY. 06.014.183

Mengetahui
Ketua Program Studi Sarjana Terapan Teknik Informatika

Dyah Apriliani, S.T., M.Kom.
NIPY. 09.015.225

2. Surat Kesepakatan Bimbingan Pembimbing II

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama	: Siska Aulia Utami
NIM	: 21090048
Program Studi	: Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama	: Dega Surono Wibowo, S.T., M.Kom.
Status	: Dosen
NIDN	: 0607108202
Jabatan Fungsional	: Lektor
Pangkat/Golongan	: Penata /III/C

Pada hari ini Jum'at tanggal 25 April 2025 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Skripsi Pihak Pertama dengan syarat melakukan bimbingan 1 kali dalam seminggu atau setidaktidaknya 3 kali bimbingan dalam satu bulan (dengan progress), apabila saya tidak memenuhi syarat tersebut maka saya tidak berhak meminta surat rekomendasi mengikuti sidang skripsi. Dan saya juga berjanji akan memenuhi persyaratan tersebut dan menyelesaikan skripsi baik produk maupun laporan tepat waktu.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

Tegal, 25 April 2025

Pihak Pertama	Pihak Kedua
Siska Aulia Utami	Dega Surono Wibowo, S.T., M.Kom. NIPY. 06.014.183

Mengetahui
Ketua Program Studi Sarjana Terapan Teknik Informatika

Dyah Apriliani, S.Pd., M.Kom.
NIPY. 09.015.225

Lampiran 4 Lembar Bimbingan

1. Lembar Bimbingan Pembimbing I

D IV TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

Nama	: Siska Aulia Utami
NIM	: 21090048
No. Ponsel	: 081779923491
Judul TA	: Rancang Bangun Aplikasi Manajemen Customer Service Dalam Meningkatkan Resolusi Keluhan Berbasis Website (PT. Medika Digital Nusantara)
Dosen Pembimbing I	: Dega Surono Wibowo,S.T., M.Kom

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	16/5 2025	perbaiki konsep / proses bisnis. perbaiki usecase berdasarkan proses bisnis		
2.	22/5 2025	silahkan buat perancangan sistem / antarmuka.		
3.	4/6 2025	silahkan di develop sambil ditest mengurutkan selenium		
4.	13/6 2025	silahkan belajar coding. sambil buat manual book.		
5.	19/6 2025	P1 ada. P3 ada. P2 tidak ada? buat apa		

6	25/6 2025	tambah proses revisi, tanpa pengajuan baru lagi		
7.	7/7 2025	Silahkan Daftar HKI.		
8	18/7 2025	Lihat & Laporan - Sesuaikan format skripsi - class diagram pelajari MVC.		
9.	21/7 2025	Silahkan Daftar Sidang		

Acc
Silahkan
daftar
21/7 2025

Tegal, 21/7 2025
Pembimbing 1,

Dega Surono Wibowo,S.T., M.Kom
NIPY. 06.014.183

2. Lembar Bimbingan Pembimbing 2

D IV TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

Nama	: Siska Aulia Utami
NIM	: 21090048
No. Ponsel	: 081779923491
Judul TA	: Rancang Bangun Aplikasi Manajemen Customer Service Dalam Meningkatkan Resolusi Keluhan Berbasis Website (PT. Medika Digital Nusantara)
Dosen Pembimbing I	: Muhammad Fikri Hidayatullah, S.T., M.Kom

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	28 Mei 2025	Prototipe	- Bisnis proses harus rasional & efisien - Ada beberapa fitur yg tidak dibutuhkan	
2	29/-2025 Juni			
3	8 Juni 2025	Aplikasi	Belum bisa lisa coding	
4.	9 Juli	Aplikasi	—"—	
5.	10 Juli	Aplikasi	—"—	

6.	15 Juli	Liva Coding	ok	
7.	23 Juli	Laporan	- UML - Tinjauan Pustaka	
8.	24 Juli	Laporan	Tinjauan Pustaka	
9	24 Juli	Laporan	Bab	
10	25 Juli	Laporan	ACC	

Tegal,
Pembimbing 2,

Muhammad Fikri Hidayatullah, S.T., M.Kom
NIPY. 09.016.307

Lampiran 5 Surat Pengajuan HKI

1. Surat Pengalihan HKI

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini

1	Nama	:	Siska Aulia Utami
	Kewarganegaraan	:	Indonesia
	Alamat	:	Desa Pagedangan RT.007 RW.001 Kec Adiwerna, Kota Tegal
2.	Nama	:	Dega Surono Wibowo, S.T., M.Kom
	Kewarganegaraan	:	Indonesia
	Alamat	:	Perumahan Sapphire Regency Blok H, No.1, RT.004, RW.001,Kelurahan Pulosari, Kecamatan Brebes, 52213.
3.	Nama	:	Muhammad Fikri Hidayattullah, S.T., M.Kom
	Kewarganegaraan	:	Indonesia
	Alamat	:	Jl. Glatik No. 68 Randugunting, Kec Tegal Selatan, Kota Tegal.

Adalah **Pihak I** selaku pencipta, dengan ini menyerahkan karya ciptaan saya kepada

Nama	:	Pusat Penelitian dan Pengabdian Masyarakat (P3M)
Alamat	:	Jl. Mataram No. 9 Pesurungan Lor Tegal

Adalah **Pihak II** selaku Pemegang Hak Cipta berupa Program Komputer dengan judul "Aplikasi Manajemen Customer Service Berbasis Website (PT. Medika Digital Nusantara)" untuk didaftarkan di Direktorat Hak Cipta dan Desain Industri, Direktorat Jenderal Kekayaan Intelektual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.

Demikianlah surat pengalihan hak ini kami buat, agar dapat dipergunakan sebagaimana mestinya.

Tegal, Juli 2025

Pemegang Hak Cipta
Kepala P3M

(Muhammad Fikri Hidayattullah, S.T., M.Kom.)

Pencipta

(Siska Aulia Utami)

(Dega Surono Wibowo, S.T., M.Kom)

(Muhammad Fikri Hidayattullah, S.T., M.Kom)

2. Surat Pernyataan HKI

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta.

1. Nama : Siska Aulia Utami
   Kewarganegaraan : Indonesia
   Alamat : Desa Pagedangan RT.007 RW.001 Kec Adiwerna, Kota Tegal.

2. Nama : Dega Surono Wibowo, S.T., M.Kom.
   Kewarganegaraan : Indonesia
   Alamat : Perumahan Sapphire Regency Blok H, No.1, RT.004. RW.001,Kelurahan Pulosari, Kecamatan Brebes, 52213.

3. Nama : Muhammad Fikri Hidayattullah, S.T., M.Kom.
   Kewarganegaraan : Indonesia
   Alamat : Jl. Glatik No. 68 Randugunting, Kec Tegal Selatan, Kota Tegal

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:
   Berupa : Program Komputer
   Berjudul : Aplikasi Manajemen Customer Service Berbasis Website (PT. Medika Digital Nusantara)
   - Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyek kekayaan intelektual lainnya sebagaimana dimaksud dalam Pasal 68 ayat (2);
   - Bukan merupakan Ekspresi Budaya Tradisional sebagaimana dimaksud dalam Pasal 38;
   - Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
   - Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;
   - Bukan merupakan Ciptaan seni lukis yang berupa logo atau tanda pembeda yang digunakan sebagai merek dalam perdagangan barang/jasa atau digunakan sebagai lambang organisasi, badan usaha, atau badan hukum sebagaimana dimaksud dalam Pasal 65 dan;
   - Bukan merupakan Ciptaan yang melanggar norma agama, norma susila, ketertiban umum, pertahanan dan keamanan negara atau melanggar peraturan perundang-undangan sebagaimana dimaksud dalam Pasal 74 ayat (1) huruf d Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

2. Sebagai pemohon mempunyai kewajiban untuk menyimpan asli contoh ciptaan yang dimohonkan dan harus memberikan apabila dibutuhkan untuk kepentingan penyelesaian sengketa perdata maupun pidana sesuai dengan ketentuan perundang-undangan.

3. Karya Cipta yang saya mohonkan pada Angka 1 tersebut di atas tidak pernah dan tidak sedang dalam sengketa pidana dan/atau perdata di Pengadilan.

4. Dalam hal ketentuan sebagaimana dimaksud dalam Angka 1 dan Angka 3 tersebut di atas saya / kami langgar, maka saya / kami bersedia secara sukarela bahwa:
   a. permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau
   b. Karya Cipta yang telah terdaftar dalam Daftar Umum Ciptaan Direktorat Hak Cipta, Direktorat Jenderal Hak Kekayaan Intelektual, Kementerian Hukum Dan Hak Asasi Manusia R.I dihapuskan sesuai dengan ketentuan perundang-undangan yang berlaku.
   c. Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam berperkara dan/atau sedang dalam gugatan di Pengadilan maka status kepemilikan surat pencatatan elektronik tersebut ditangguhkan menunggu putusan Pengadilan yang berkekuatan hukum tetap.

Tegal, Juli 2025

METERAI TEMPEL
5D5DAAMX253851820

(Siska Aulia Utami)
Pemegang Hak Cipta *

(Dega Surono Wibowo, S.T., M.Kom)
Pemegang Hak Cipta *

(Muhammad Fikri Hidayatullah, S.T., M.Kom)
Pemegang Hak Cipta *

* Semua pemegang hak cipta agar menandatangani di atas materai

REPUBLIK INDONESIA
KEMENTERIAN HUKUM

SURAT PENCATATAN CIPTAAN

Dalam rangka pelindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan	:	EC002025091198, 17 Juli 2025
Pencipta		
Nama	:	SISKA AULIA UTAMI, Dega Surono Wibowo,S.T., M.Kom dkk
Alamat	:	Desa Pagedangan RT 007 RW 001, Adiwerna, Kab. Tegal, Jawa Tengah, 52194
Kewarganegaraan	:	Indonesia
Pemegang Hak Cipta		
Nama	:	Pusat Penelitian dan Pengabdian Masyarakat (P3M) Politeknik Harapan Bersama
Alamat	:	Jalan Mataram No. 9, Pesurungan Lor, Kecamatan Margadana, Margadana, Kota Tegal, Jawa Tengah, 52142
Kewarganegaraan	:	Indonesia
Jenis Ciptaan	:	Program Komputer
Judul Ciptaan	:	Aplikasi Manajemen Customer Service Berbasis Website (PT. Medika Digital Nusantara)
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	:	17 Juli 2025, di Kota Tegal
Jangka waktu pelindungan	:	Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor Pencatatan	:	000931459

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL
u.b
Direktur Hak Cipta dan Desain Industri

Agung Damarsasongko,SH.,MH.
NIP. 196912261994031001

Balai Besar Sertifikasi Elektronik

Disclaimer:
1. Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.
2. Surat Pencatatan ini telah disegel secara elektronik menggunakan segel elektronik yang ditertibkan oleh Balai Besar Sertifikasi Elektronik, Badan Siber dan Sandi Negara.
3. Surat Pencatatan ini dapat dibuktikan keasliannya dengan memindai kode QR pada dokumen ini dan informasi akan ditampilkan dalam browser.

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	SISKA AULIA UTAMI	Desa Pagedangan RT 007 RW 001 Adiwerna, Kab. Tegal
2	Dega Surono Wibowo,S.T., M.Kom	Perumahan Sapphire Regency Blok H, No.1, RT.004, RW.001, Kelurahan Pulosari Brebes, Kab. Brebes
3	Muhammad Fikri Hidayattullah, S.T., M.Kom.	Jl. Glatik No. 68 Randugunting Tegal Selatan, Kota Tegal

MANUAL BOOK

APLIKASI CUSTOMER SERVICE MANAJEMENT BERBASIS WEBSITE

OLEH :
SISKA AULIA UTAMI
DEGA SURONO WIBOWO,S.T., M.KOM.
MUHAMMAD FIKRI HIDAYATTULLAH, S.T., M.KOM.

1. PENDAHULUAN

Aplikasi *Customer service Manajement* (CSM) adalah sistem berbasis *Web*site yang dirancang untuk membantu proses penanganan keluhan pelanggan secara lebih terstruktur dan efisien. Aplikasi ini memfasilitasi dua jenis pengguna, yaitu Admin dan *Client*/*User*, untuk melakukan proses manajemen keluhan, monitoring proyek, hingga komunikasi melalui fitur chat. Manual Book ini disusun untuk memudahkan pengguna dalam memahami serta mengoperasikan fitur-fitur yang tersedia di aplikasi CSM.

1.1. Tujuan Pembuatan Dokumen

Tujuan dibuatnya dokumen petunjuk penggunaan *Web*site Aplikasi *Customer service Manajement* berbasis *Website* ini dibuat untuk tujuan sebagai berikut :

1. Petunjuk Penggunaan ini digunakan untuk menjelaskan kepada pengguna (*User/*admin) mengenai penggunaan Aplikasi *Customer service Manajement.*
2. Untuk menggambarkan penggunaan Aplikasi *Customer service Manajement* berbasis *Website.*
3. Mempermudah penggunaan (*User*/admin) dalam menggunakan Aplikasi *Customer service Manajement* berbasis *Website.*

1.2. Deskripsi Umum Sistem

1.2.1. Deskripsi Umum Aplikasi

Deskripsi umum aplikasi meliputi dari aplikasi yang telah dibuat dengan memfasilitasi komunikasi antara pelanggan dan tim layanan pelanggan

(*Customer service*) mulai dari pendaftaran *client*, pengajuan keluhan, pemantauan proyek, hingga komunikasi melalui fitur chat yang menjadi bahan acuan pembuatan aplikasi.

1.2.2. Deskripsi Umum Kebutuhan Aplikasi

Deskripsi kebutuhan aplikasi yang akan diimplementasikan meliputi semua informasi yang bersifat teknis yang menjadi acuan dalam pengembangan aplikasi.

1.2.3. Deskripsi Dokumen

Dokumen ini dibuat untuk memberikan panduan penggunaan Aplikasi *Customer service Manajement* berbasis *Website* di PT. Medika Digital Nusantara. Dokumen ini berisi informasi sebagai berikut:

1. **BAB I**

   Berisi informasi umum yang merupakan bagian pendahuluan, yang meliputi tujuan pembuatan dokumentasi, deskripsi umum dan sistem serta deskripsi dokumen.

2. **BAB II**

   Berisi perangkat yang dibutuhkan untuk pembuatan Aplikasi *Customer service Manajement* berbasis *Website* di PT. Medika Digital Nusantara meliputi perangkat lunak (*Software*) dan perangkat keras (*Hardware*).

3. **BAB III**

Berisi panduan penggunaan (*Manual Book*) Aplikasi *Customer service Manajement* berbasis *Website* di PT. Medika Digital Nusantara.

2. PERANGKAT YANG DIBUTUHKAN

2.1. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan :

1. Windows sebagai Operating System.
2. Google Chrome
3. *Laragon*
4. *Visual Studio Code*
5. *Laravel* 11
6. *Selenium*

2.2. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan meliputi :

1. Laptop Asus
2. Ram 8 GB
3. Storage 512 GB

3. MENU DAN CARA PENGGUNAAN

3.1. Struktur Menu

Adapun struktur menu pada aplikasi penggunaan *Web*site Aplikasi *Customer service Manajement* berbasis *Website* adalah sebagai berikut :

1. Menu *User* (*Customer Client*)

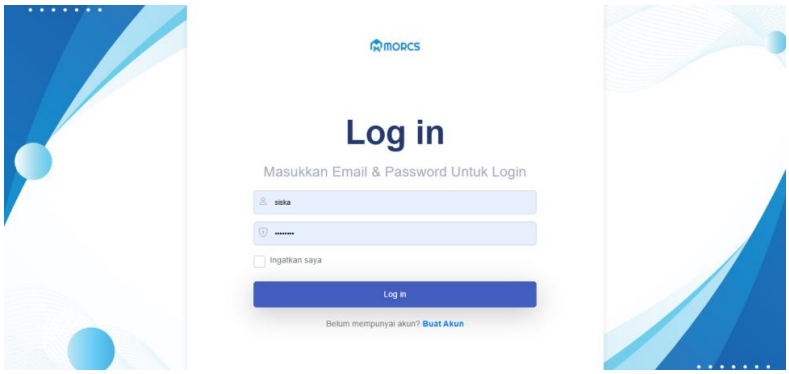
- *Login*
- *Dashboard User*
- Tambah keluhan
- *Profile*
- Projek
- *chatbot*

2. **Menu Admin (*Customer service*)**

- *Register*
- *Login*
- *Dashboard* admin
- *Profile*
- Projek
- Data *client*
- *chatbot*

3.2. Pengguna

Bagian ini merupakan panduan penggunaan aplikasi dengan menggunakan gambar dan penjelasan mengenai fitur – fitur yang tersedia pada aplikasi dan panduan pengimputan data ke dalam aplikasi tersebut.



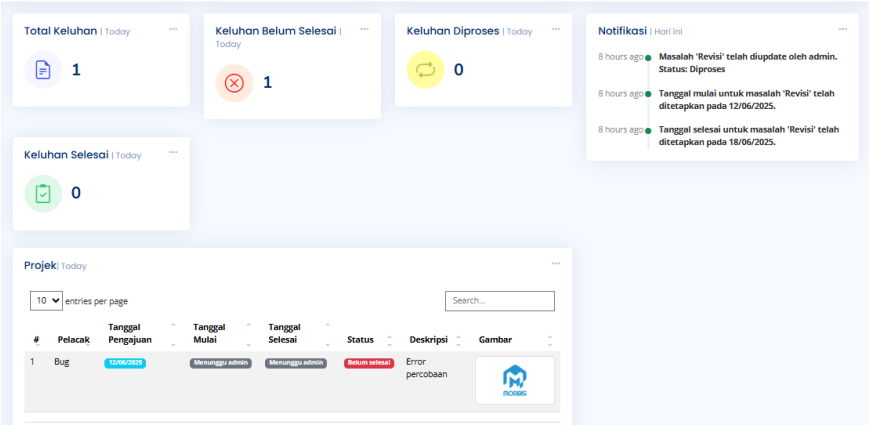
Gambar 3.1 *Login all role*

User dan admin di haruskan mengisi *form login* agar dapat masuk ke halaman selanjutnya. Jika sudah memasukkan *Email* dan *password* kemudian klik tombol *login* maka akan masuk ke halaman *dashboard User* atau admin.

3.2.1. Halaman *User*

a. *Dashboard User*

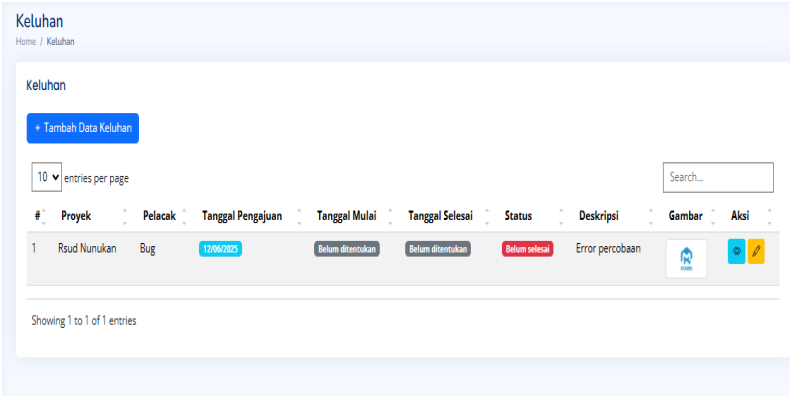
Dashboard User merupakan halaman awal ketika *User* berhasil *Login*, pada aplikasi *Customer service Manajement* berbasis *Website.* Halaman *dashboard User* menampilkan statistik personal seperti jumlah keluhan yang pernah diajukan, status keluhan, dan proyek yang sedang berlangsung. *Dashboard* ini dirancang agar *User* dapat memonitor aktivitas dan perkembangan penanganan keluhan secara langsung. Berikut merupakan tampilan dashboard *User* dapat dilihat pada gambar 3.3 dibawah ini.



Gambar 3.2 Dashboard *User*

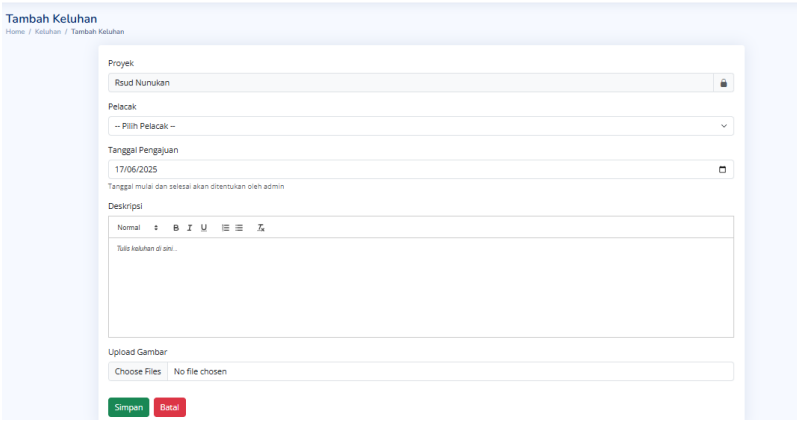
b. Halaman Keluhan

Fitur ini merupakan bagian utama bagi *client* untuk memantau masalah atau keluhan yang telah mereka ajukan oleh pengguna (*client*). Pengguna dapat melihat perkembangan keluhan mereka secara *real-time* dengan tampilan yang terorganisir ini. Berikut merupakan tampilan halaman keluhan *User* dapat dilihat pada gambar 3.4 dibawah ini.



Gambar 3.3 Halaman Keluhan

Selain itu, halaman ini memiliki berbagai fitur pendukung untuk mempermudah interaksi dan tindak lanjut antara *client* dan tim *cutomer service*. Untuk menambahkan keluhan baru client dapat menekan tombol "**Tambah Data Keluhan**" > sistem akan membuka ***form* pengisian keluhan** > **Simpan** jika sudah terisi dengan benar. Berikut gambar fitur tambah keluhan dapat dilihat pada gambar 3.5 dibawah ini:

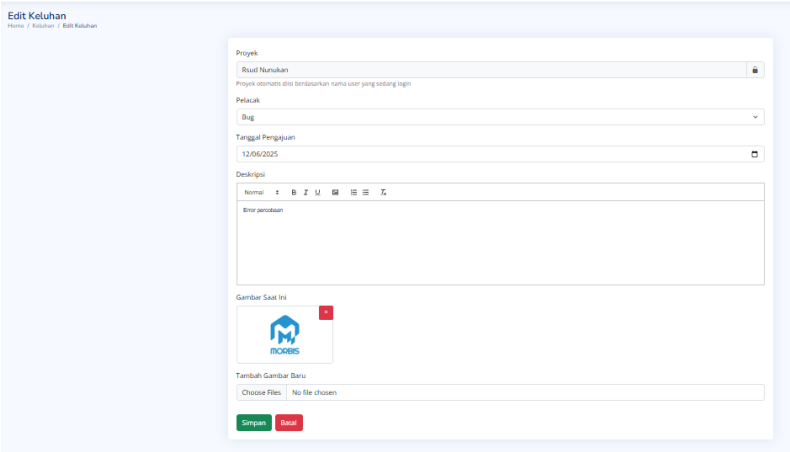


Gambar 3.4 *Tambah Data Keluhan*

Di tabel keluhan, memiliki dua tombol utama berada di kolom Aksi. Untuk mengedit jika ada keluhan yang telah diajukan kurang tepat,

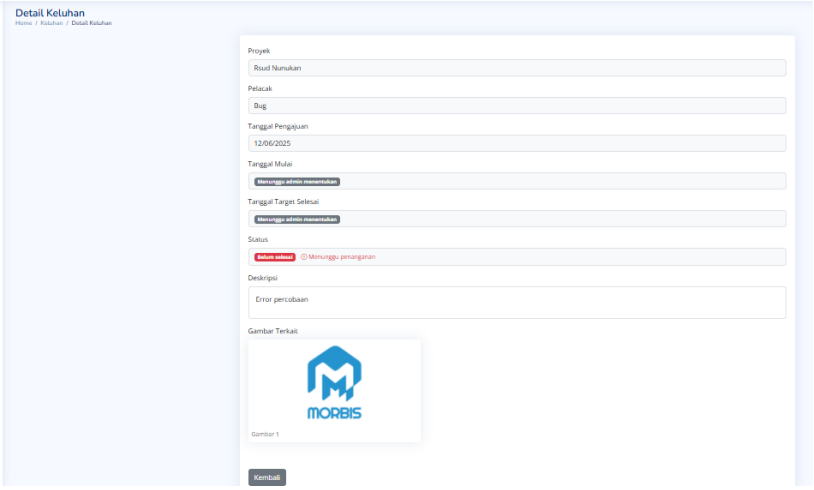
Anda dapat menggunakan **tombol Edit (ikon pensil)** > masuk ke ***form* pengisian keluhan** untuk mengubah deskripsi atau menambahkan lampiran > **Simpan perubahan** jika sudah terisi dengan benar.

Namun, jika *status* keluhan telah di*proses* oleh admin, keluhan yang terdaftar akan dipindahkan ke halaman projek. Berikut gambar fitur edit keluhan dapat dilihat pada gambar 3.6 dibawah ini:



Gambar 3.5 *Fitur Edit Keluhan*

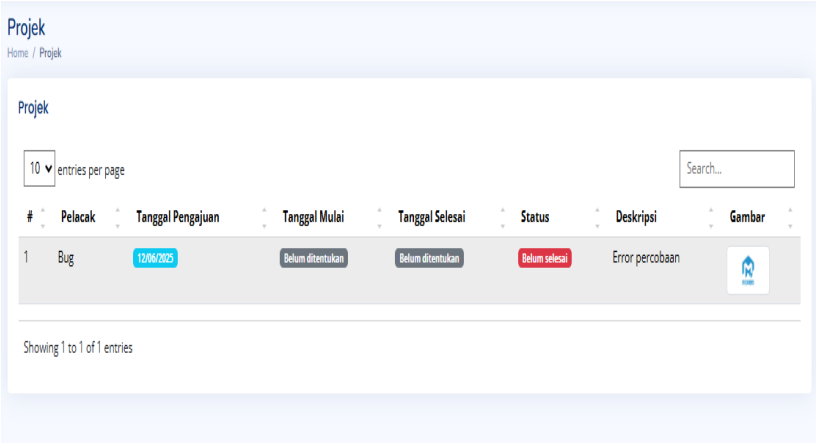
Pada tombol **aksi *Detail* (ikon mata)** memungkinkan *client* melihat isi keluhan secara keseluruhan, untuk melakukannya *client* dapat meninjau keluhan dengan: masuk ke **keluhan** > klik ***Detail* (ikon mata)** pada table **aksi** > klik **kembali** untuk mengakhiri fitur. Berikut gambar fitur detail keluhan dapat dilihat pada gambar 3.7 dibawah ini:



Gambar 3.6 Fitur detail keluhan

c. Halaman Projek *User*

Halaman ini mengelola keluhan *client* dan menampilkan daftar proyek yang telah dibuat. Jenis keluhan, tanggal pengajuan, *status* penanganan, dan deskripsi proyek ditampilkan dalam tabel untuk setiap proyek. Berikut gambar fitur halaman projek dapat dilihat pada gambar 3.8 dibawah ini:

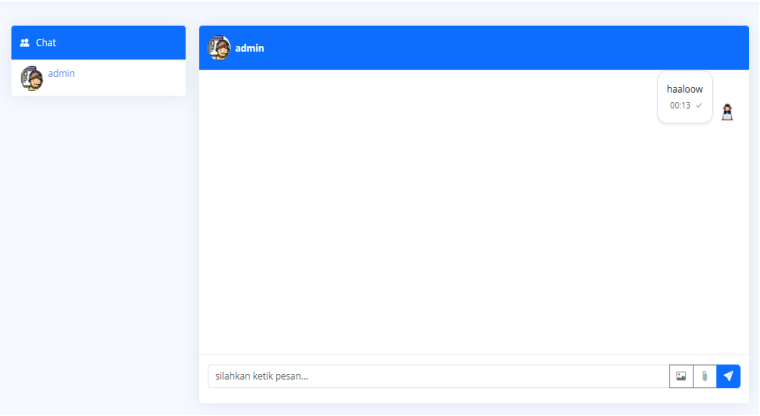


Gambar 3.7 Halaman Projek

Dengan adanya tabel projek memungkinkan *client* agar bisa memantau status terkini, durasi penanganan, dan detail proyek tanpa harus menghubungi admin langsung.

d. Halaman *Chat User*

User hanya bisa menampilkan percakapan dengan admin sebagai komunikasi langsung dengan admin untuk menyamoaikan pertanyaan atau informasi tambahan terkait keluhan yang telah diajukan. Pengguna cukup mengeklik nama admin untuk memulai percakapan. Berikut gambar fitur *chat User* dapat dilihat pada gambar 3.9 dibawah ini:



Gambar 3.8 *fitur* chat *User*

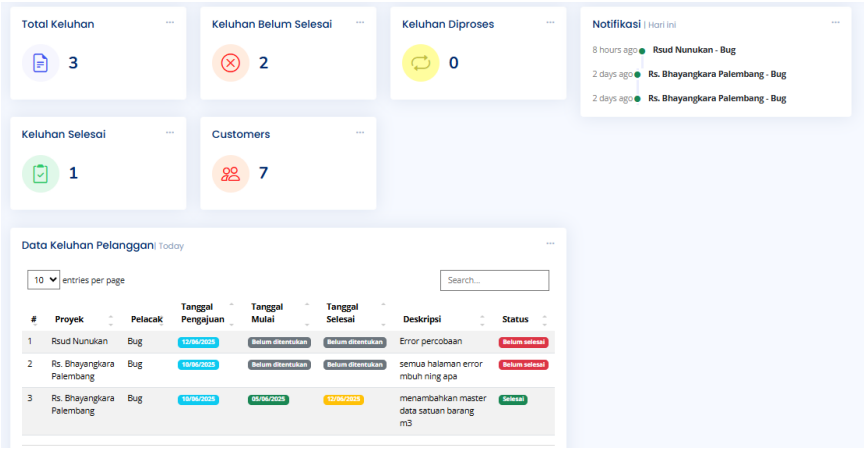
3.2.2. Halaman Admin

Sebagian menu halaman admin memiliki kesamaan dengan halaman *User/client*, namun pada halaman *User* terdapat beberapa menu tambahan untuk admin yaitu menu kelola data keseluruhan *client*.

a. Dashboard Admin

Halaman *Dashboard* Admin berfungsi sebagai pusat informasi utama yang memberikan gambaran umum terkait aktivitas dan data

keluhan dari seluruh *client*. Tampilan ini membantu admin dalam memantau jumlah laporan masuk, status penanganan, serta notifikasi terbaru secara *real-time*. Juga menampilkan *notifikasi* daftar aktivitas terkini, seperti keluhan baru yang masuk atau *status* perubahan keluhan. Waktu aktivitas juga dicantumkan (misalnya "2 *days ago*") untuk memudahkan pemantauan waktu respon. Berikut gambar fitur halaman *dashboard* admin dapat dilihat pada gambar 3.10 dibawah ini:



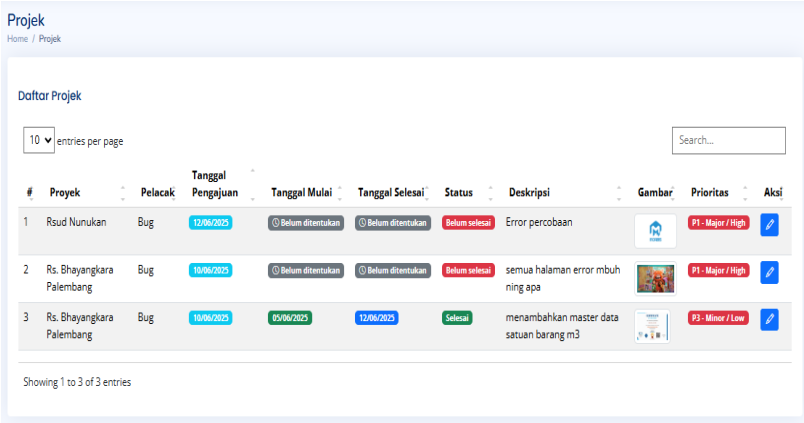
Gambar 3.9 Halaman Dashboard Admin

Dengan tampilan data yang informatif, mudah dibaca, dan responsif terhadap aktivitas terbaru, dashboard ini dibuat untuk membantu admin mengelola proses layanan secara efisien.

b. Halaman Projek Admin

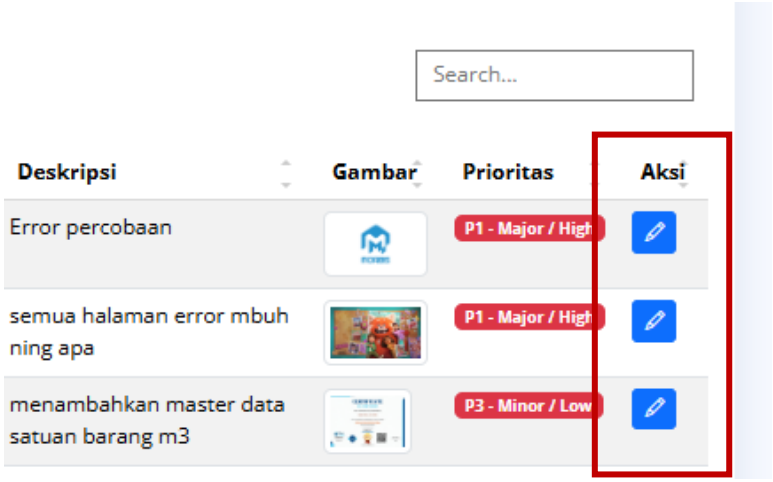
Halaman ini menampilkan daftar proyek yang terbentuk dari keluhan pengguna. Admin dapat melihat detail proyek, menetapkan prioritas, serta mengatur tanggal mulai dan selesai pengerjaan. Setiap proyek ditampilkan dalam tabel berisi informasi penting seperti nama proyek, jenis keluhan (pelacak), tanggal pengajuan, status, deskripsi,

dan lampiran gambar. Status proyek ditandai dengan warna berbeda: **merah** (Belum selesai), **kuning** (Diproses), dan **hijau** (Selesai), sehingga memudahkan pemantauan progres. Berikut gambar fitur halaman projek admin dapat dilihat pada gambar 3.11 dibawah ini:

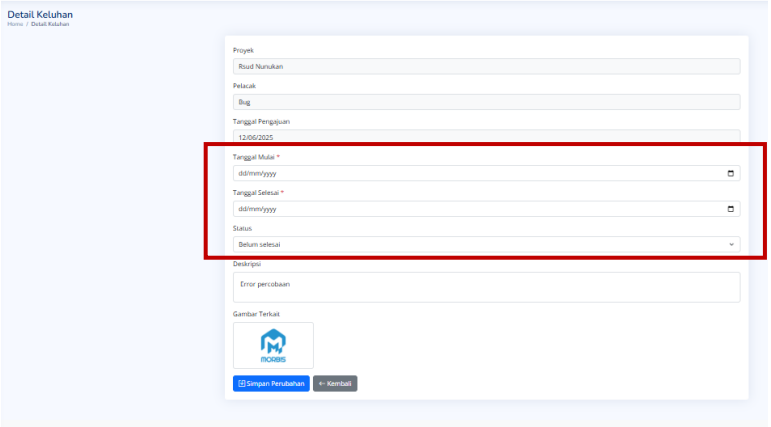


Gambar 3.10 Halaman Projek Admin

Admin bertugas untuk mengelola semua proyek. Status proyek, ditentukan langsung oleh admin melalui fitur **Edit**, begitu pula dengan pengisian tanggal mulai dan tanggal selesai. untuk melakukan pengelolaan admin pilih aksi **Edit (ikon pensil)** > pilih **Status Proyek** dapat dipilih seperti (Belum selesai, Diproses, atau Selesai) > isikan **Tanggal mulai & Tanggal selesai** sesuai yang telah ditentukan / disepakati > jika sudah **Simpan Perubahan**. Berikut gambar fitur aksi edit dapat dilihat pada gambar 3.12 dibawah ini:



Gambar 3.11 fitur aksi edit



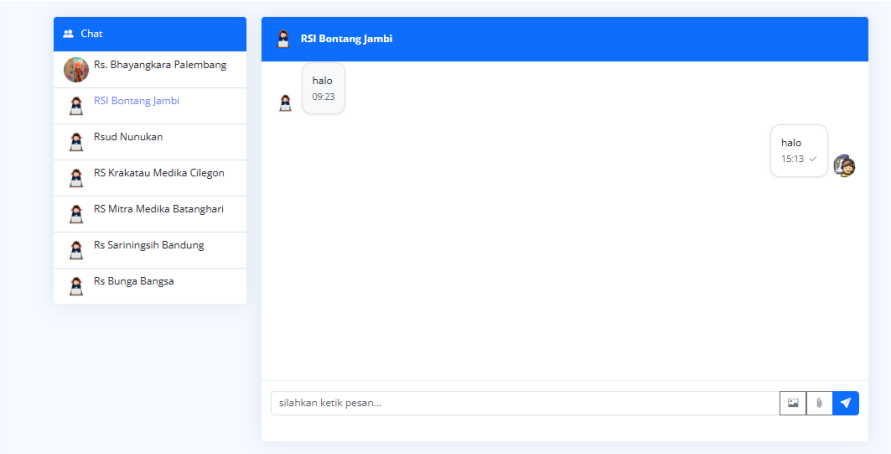
Gambar 3.12 Edit Keluhan admin

Tombol **aksi** seperti **Edit** tersedia untuk setiap proyek yang masuk dari *client*. Dengan fitur ini, admin dapat mengontrol progres proyek secara menyeluruh, mulai dari m enetapkan jadwal hingga memantau penyelesaian keluhan secara efisien.

c. Halaman *Chat* admin

Fitur *chat* memungkinkan admin berkomunikasi langsung dengan seluruh *client*. **Di sisi kiri**, admin dapat memilih nama *client* dari daftar kontak. **Di sisi kanan**, ditampilkan isi percakapan lengkap dengan

waktu pengiriman. Admin dapat membalas pesan serta mengirimkan *file* atau gambar melalui kolom input di bagian bawah. Fitur ini dirancang agar komunikasi lebih cepat, personal, dan terdokumentasi secara sistematis dalam aplikasi. Berikut gambar fitur *chat* admin dapat dilihat pada gambar 3.14 dibawah ini:

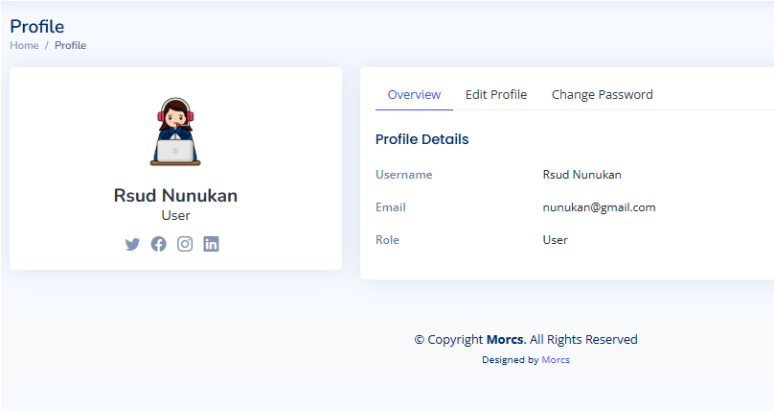


Gambar 3.13 Fitur *Chat* Admin

3.2.3. Fitur Umum (Semua *role*)

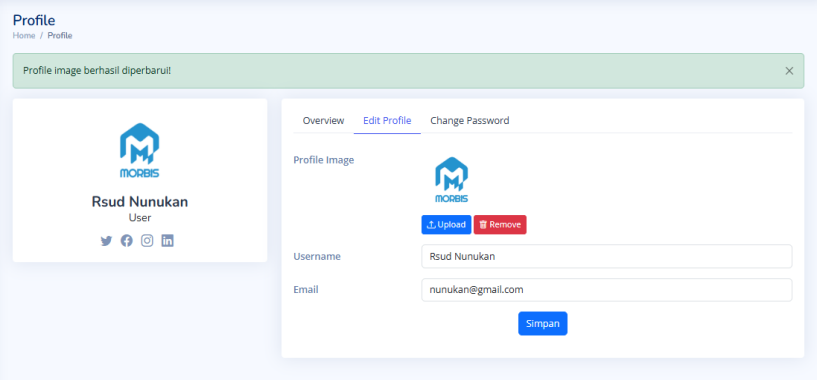
a. Profile

Baik admin maupun *User* dapat mengakses halaman "*Profile*" untuk melihat dan memperbarui data pribadi, termasuk *Username*, email, gambar *profile*, dan *password*. Halaman ini juga menyediakan opsi untuk memperkuat keamanan akun dengan mengganti sandi secara berkala. Maksud dari fitur ini adalah menjaga data tetap akurat dan aman bagi semua pengguna sistem. Berikut gambar halaman *overview profile* dapat dilihat pada gambar 3.15 dibawah ini:



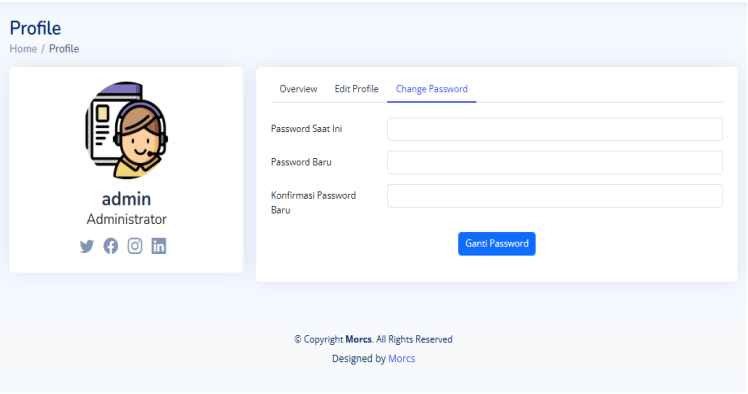
Gambar 3.14 Halaman *Overview Profile*

Pada menu edit *profile*, admin/*User* dapat memperbarui dengan mudah, caranya: masuk ke ***Profile*** > klik bagian **Edit *Profile*** > klik **simpan** untuk menyimpan perubahan *profile*. Jika sudah terubah maka akan muncul ***pop up*** berhasil di perbarui dibagian atas. Berikut gambar halaman edit *profile* dapat dilihat pada gambar 3.16 dibawah ini:



Gambar 3.15 Halaman Edit *Profile*

Bagian pro*file* juga terdapat change password. Admin/ *User* dapat mengubah password dengan cara masuk ke ***profile*** > klik bagian ***change password*** > lalu isikan ***form change password*** > klik **ganti password** jika sudah terubah. Berikut gambar halaman *change password* dapat dilihat pada gambar 3.17 dibawah ini:



Gambar 3 16 Change Password

DOKUMEN TEKNIKAL

APLIKASI CUSTOMER SERVICE MANAJEMENT BERBASIS WEBSITE

OLEH :
SISKA AULIA UTAMI
DEGA SURONO WIBOWO,S.T., M.KOM.
MUHAMMAD FIKRI HIDAYATTULLAH, S.T., M.KOM.

Profil

Morcs merupakan Aplikasi *customer service* berbasis *Website* yang dapat membantu PT Medika Digital Nusantara dalam menangani keluhan pelanggan dengan lebih cepat, transparan, dan efisien. Aplikasi ini untuk mempercepat komunikasi antara admin dan klien dan memastikan bahwa setiap keluhan dicatat dengan baik dan proyek ditindaklanjuti secara *real-time*. Dengan adanya sistem ini, pelanggan dapat lebih mudah mengajukan, melacak, dan mendapatkan solusi atas keluhan mereka,

Latar Belakang

Di era digital yang semakin maju, pengelolaan layanan pelanggan menjadi komponen penting bagi bisnis untuk meningkatkan kepuasan dan loyalitas pelanggan. Interaksi antara bisnis dan pelanggan saat ini tidak lagi terbatas pada transaksi, tetapi telah berkembang menjadi hubungan yang kompleks yang membutuhkan kepedulian, kecepatan, dan transparansi dalam layanan. Hal ini terjadi di lingkungan bisnis global saat ini. Meningkatnya akses digital mendorong masyarakat untuk menjadi lebih ekspresif dalam menyampaikan pengalaman pelanggan mereka melalui kanal online, seperti platform layanan pelanggan berbasis *Web* dan media sosial [1]. Di Indonesia, pergeseran pola konsumsi digital menunjukkan perubahan besar dalam harapan masyarakat terhadap sistem pelayanan yang lebih responsif. Namun, di tengah transformasi digital, beberapa perusahaan masih belum dapat membangun sistem penanganan keluhan pelanggan yang sistematis dan mudah diakses.

PT. Medika Digital Nusantara, sebuah perusahaan teknologi kesehatan yang melayani puluhan klien dari berbagai lembaga kesehatan, secara khusus menyaksikan kondisi ini. Proses menangani keluhan klien masih dilakukan sebagian besar secara manual dengan menggunakan komunikasi langsung atau *telegram*, tidak ada sistem komprehensif yang dapat mengatur alur keluhan secara berjenjang dan terdokumentasi. Dalam keadaan seperti ini, ada risiko kehilangan informasi penting, keterlambatan *respons*, dan ketidaksesuaian antara tindak lanjut perusahaan dan ekspektasi pelanggan. Kondisi ini menunjukkan konflik antara keterbatasan struktur teknologi internal yang belum sepenuhnya terotomatisasi dan kebutuhan untuk sistem layanan yang efisien.

Kondisi di atas menimbulkan kebingungan akademik dan praktis. Dalam kenyataannya, tidak mengelola keluhan dengan baik dapat menyebabkan ketidakpercayaan, yang pada gilirannya dapat mengurangi kesetiaan klien [2]. Fenomena ini menimbulkan pertanyaan konseptual dari perspektif akademik tentang bagaimana organisasi teknologi kesehatan menggunakan mekanisme digital untuk memediasi hubungan kuasa antara penyedia layanan dan penerima layanan.

Realitas sosial yang terus berubah, di mana pelanggan sekarang membutuhkan keterlibatan dan kecepatan dalam interaksi digital, menjadikan urgensi penelitian ini. PT. Medika Digital Nusantara mungkin mengalami ketimpangan informasi dan disonansi antara ekspektasi dan kapasitas organisasi karena belum adanya sistem formal yang berbasis *Web*site untuk menangani keluhan [3]. Lebih jauh, krisis representasi dalam proses transformasi digital layanan pelanggan ditunjukkan oleh

kurangnya dokumentasi atau representasi pengalaman pengguna terhadap sistem keluhan yang sedang dibangun.

Metode kritis melihat sistem digital bukan hanya sebagai alat teknis; itu adalah konstruksi sosial yang dibentuk dan dibentuk oleh hubungan kekuasaan, prinsip, dan standar organisasi [4]. Dalam hal ini, membangun sistem pengelolaan keluhan bukan sekadar membuat aplikasi; itu juga tentang bagaimana perusahaan mendengarkan, menanggapi, dan bertanggung jawab terhadap kliennya. Oleh karena itu, sangat penting untuk memahami bagaimana pengguna meresepsi fitur seperti "kirim keluhan", "respon CS", dan "lihat *status*" dan bagaimana fitur ini memediasi hubungan sosial antara perusahaan dan klien.

Dengan mempertimbangkan aspek teknis dan sosial dari sistem layanan pelanggan, penelitian ini menempati posisi yang strategis dalam pengembangan ilmu komunikasi organisasi, sosiologi teknologi, serta studi interaksi digital. Studi ini menawarkan kontribusi berupa pemahaman kontekstual atas praktik pelayanan digital yang berkembang di sektor teknologi kesehatan di Indonesia, serta refleksi kritis atas praktik mediasi keluhan yang selama ini mungkin terpinggirkan atau direduksi ke dalam angka-angka statistik tanpa makna sosial.

Dengan mempertimbangkan keseluruhan fenomena, ketegangan praksis, dan kekosongan kajian sebagaimana diuraikan, maka penelitian ini diarahkan untuk mengeksplorasi bagaimana rancangan dan implementasi sistem manajemen keluhan berbasis *Web*site dibangun dan dijalankan di PT. Medika Digital Nusantara. Fokus penelitian ini tidak hanya pada bagaimana sistem bekerja secara

teknis, tetapi bagaimana para pengguna memaknai, menggunakan, dan memengaruhi sistem tersebut dalam praktik sehari-hari.

Berdasarkan berbagai temuan penelitian sebelumnya, dapat disimpulkan bahwa penerapan sistem layanan pelanggan berbasis *Web*site merupakan langkah strategis bagi PT Medika Digital Nusantara dalam meningkatkan resolusi keluhan pelanggan. Oleh karena itu, penelitian ini akan berfokus pada perancangan dan implementasi sistem tersebut dengan mempertimbangkan aspek teknologi, keamanan, dan efisiensi layanan.

Spesifikasi Teknik:

Spesifikasi Teknik seperti:

- *Manual Book*
- *Source code*

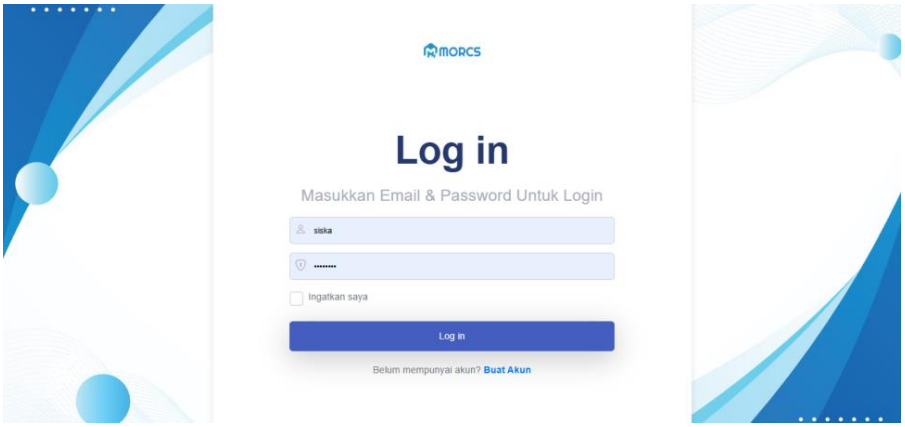
Berikut merupakan urutan spesifikasi untuk pembuatan aplikasi berbasis *Web*site:*Framework Laravel*

1. *API*
2. *Php*
3. *Laragon*

Berikut spesifik modul aplikasi:

1. **Halaman Pengguna**
   a. Halaman *Login*

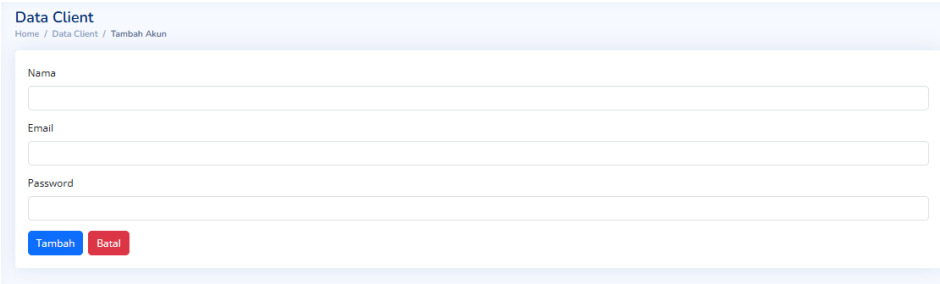
- Setiap *User* baik *User* admin atau bisa harus mengisikan *form login* yang berisikan *Username* dan *password* kemudian *button*.
- untuk *client* yang baru bekerjasama akan dibuatkan akses *login* oleh admin.
- Pertanyaan untuk melakukan register hanya untuk admin yang baru menggunakan aplikasi.



Gambar 1.1 *Halaman Login*

b. Halaman Data *Client*

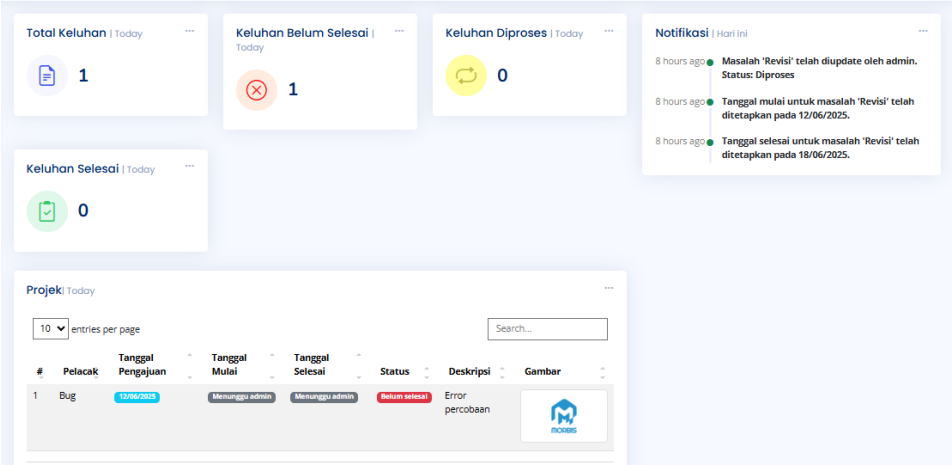
- Digunakan untuk admin menambahkan data *client* baru yang akan mengunakan aplikasi.
- Sebelumnya admin masuk dulu menggunakan *login* lalu ke halaman data *client* dan admin mengisikan *form* data *client.*
- Terdapat *button* tambah di akhir *form.*



Gambar 1.2 Halaman Data *Client*

c. Halaman *Dashboard*

- Menampilan berbagai fitur yang ada di aplikasi.
- menampilkan ringkasan data penting dalam bentuk *statistik*, *notifikasi*, dan tabel keluhan terbaru, yang membantu admin memantau aktivitas sistem secara *real-time*.

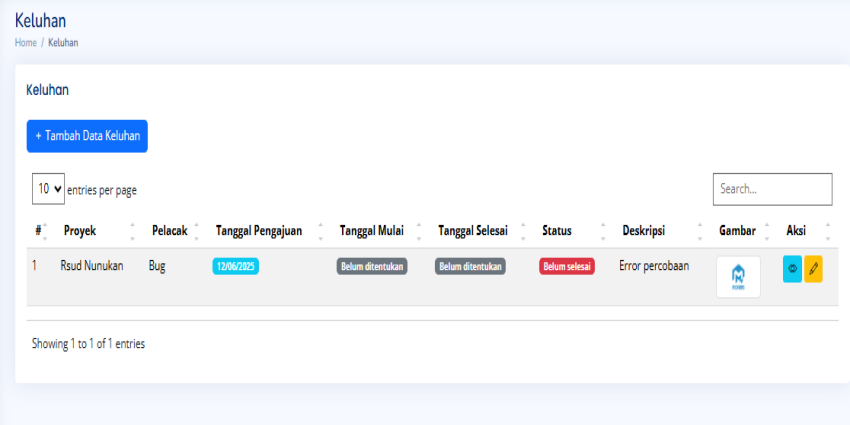


Gambar 1.3 Halaman Dashboard

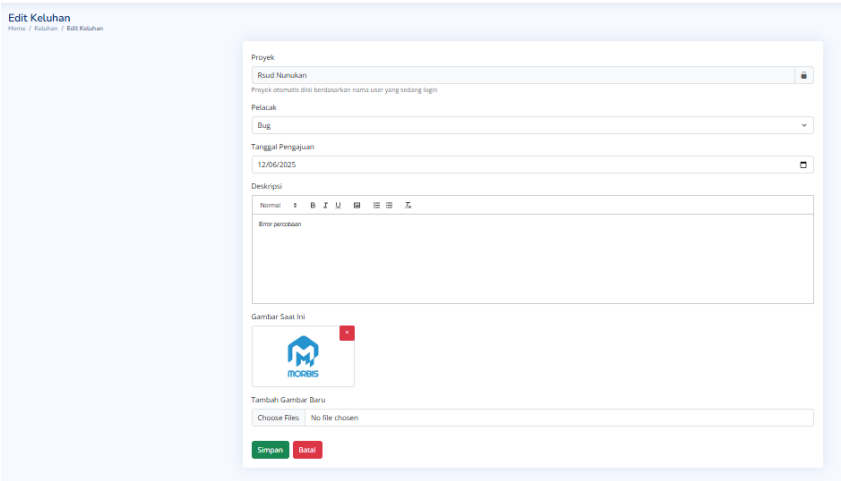
d. Halaman Keluhan

- Melaporkan masalah keluhan pada admin.

- **Mempunyai tombol "Tambah Data Keluhan"** di bagian atas halaman untuk membuat keluhan baru berupa *form input* keluhan.
- Pusat *monitoring* keluhan yang telah diajukan oleh pengguna berupa tabel data keluhan.
- Terdapat aksi *button* edit dan *detail* dibagian tabel data keluhan.



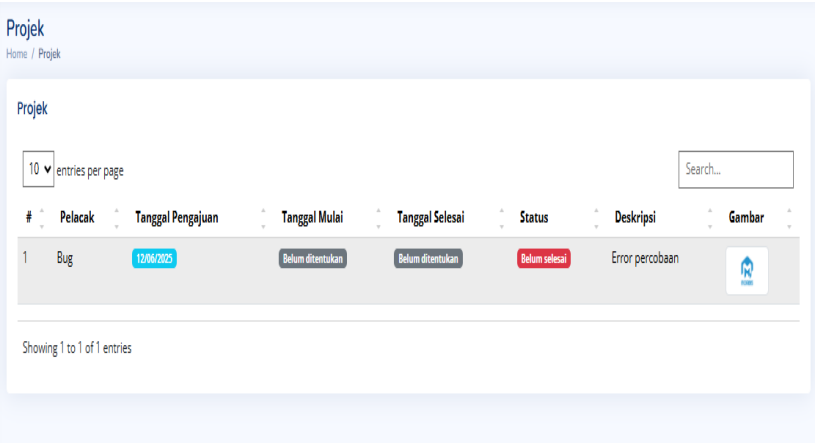
Gambar 1.4 Halaman Keluhan



Gambar 1.5 form tambah keluhan

e. Halaman Projek

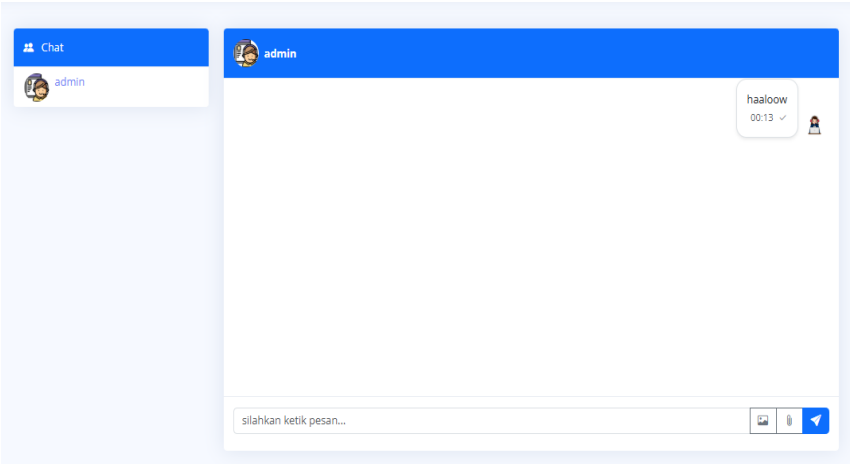
- Menampilkan daftar proyek yang telah diajukan oleh pengguna.
- Memantau *status* terkini dan durasi penanganan.
- Terdapat *search box* di pojok kiri atas.



Gambar 1.6 Halaman Projek

f. Halaman *Chat*

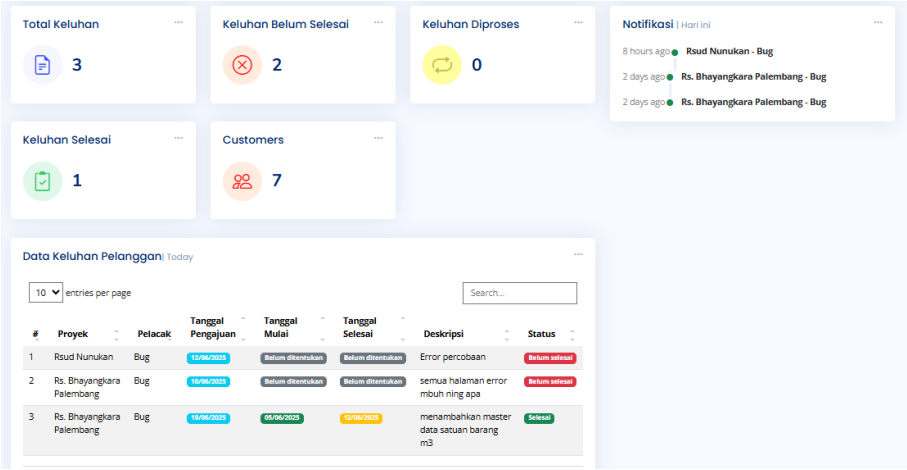
- Komunikasi langsung antara **pengguna** dan **admin** dalam aplikasi CSM.
- Dapat digunakan untuk menyampaikan *detail* tambahan dari keluhan.
- Dilengkapi dengan *ikon upload* dan tombol kirim



Gambar 1.7 Halaman *Chat*

g. Halaman *Dashboard* Admin

- Menampilkan data secara *real-time* dalam *info box*.
- Sebagai pusat pemantauan aktivitas pengguna serta *status* keluhan secara keseluruhan dalam aplikasi.
- Terdapat *search box* di pojok kiri atas.

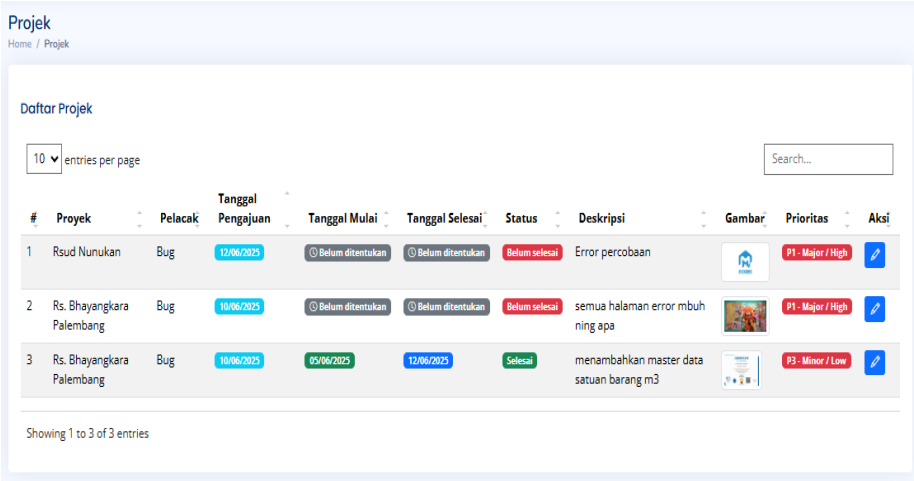


Gambar 1.8 Halaman *Dashboard* Admin

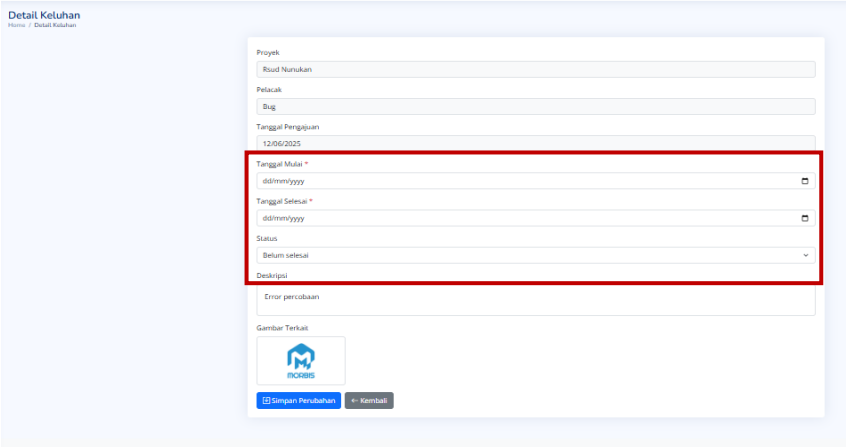
h. Halaman Projek Admin

- Memantau dan mengelola proyek yang terbentuk dari keluhan yang diajukan oleh *client*.

- Terdapat tombol edit yang didalamnya terdapat *form* untuk mengelola status pemantauan waktu pengerjaan proyek.



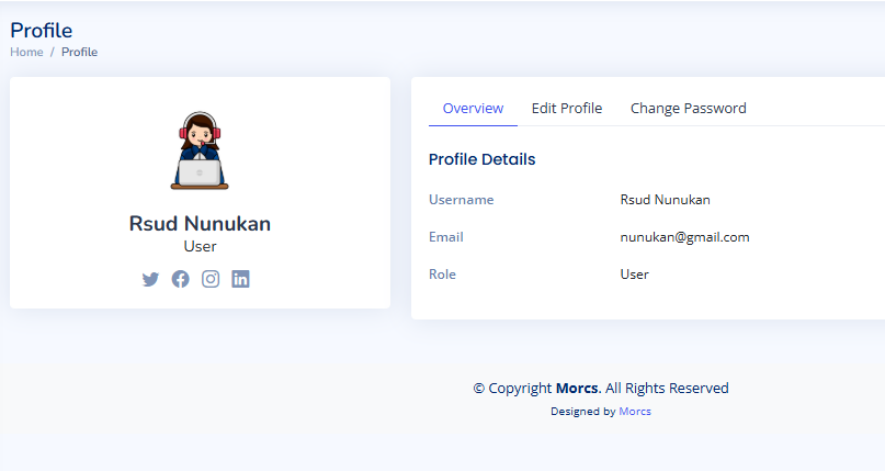
Gambar 1.9 Halaman ProJek Admin



Gambar 1. 10 Edit Keluhan admin

i. Halaman *Profile*

- Menampilkan data diri *User* / admin
- Menampilkan *icon* edit *profile* dan *change Password*
- Menampilkan foto *profile User* / admin



Gambar 1.11 Halaman *Profile*

2. *Source Code*

a. *Source code UI (User Interface)*

1) *Login.blade.php*

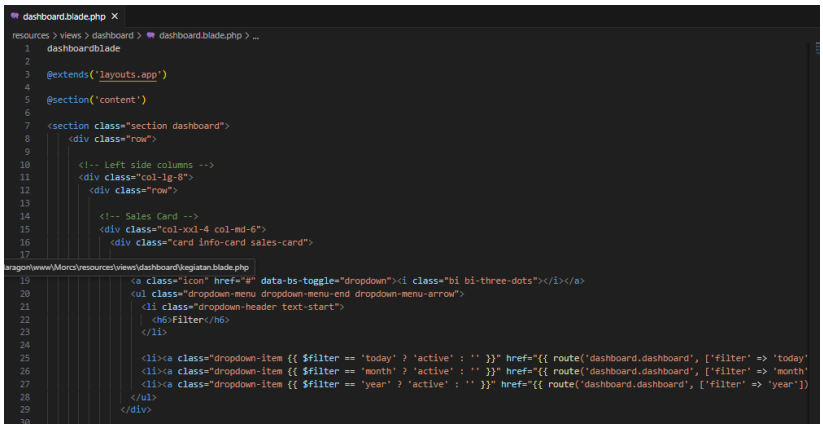
- *Source code* ini digunakan untuk menampilkan halaman *form login*, yang berisikan *Email*, *password* dan *button login*.

```
resources > views > auth > login.blade.php > html > body > div#auth > div#auth-left > div.text-center
  2   <html lang="en">
115   <body>
116       <div id="auth">
117           <div id="auth-left">
118               <div class="auth-logo">
119                   <a href="/"><img src="assets/img/logo3.jpg" alt="Logo"></a>
120               </div>
121               <h2 class="auth-title">Log in</h2>
122               <p class="auth-subtitle">Masukkan Email & Password Untuk Login</p>
123
124               <form method="post" action="{{ route('login') }}">
125                   @csrf
126                   @if (session()->has('loginError'))
127                   <div class="alert alert-danger alert-dismissible fade show" role="alert">
128                       {{ session('loginError') }}
129                       <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
130                   </div>
131                   @endif
132
133                   <div class="form-group position-relative has-icon-left">
134                       <input type="text" id="email" name="email"
135                           class="form-control @error('email') is-invalid @enderror"
136                           placeholder="Email">
137                       <div class="form-control-icon">
138                           <i class="bi bi-person"></i>
139                       </div>
140                       @error('email')
141                       <small class="text-danger">{{ $message }}</small>
142                       @enderror
143                   </div>
```

Gambar 1.12 *Source code Login*

2) Dashboard.blade.php

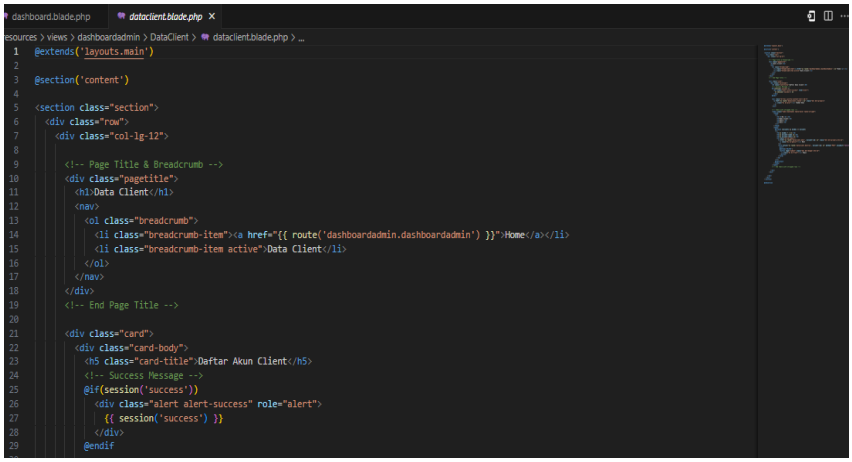
- *Source code* ini merupakan halaman *User* CSM yang terdapat pada *folder* **resources > views > dashboard.**
- Merupakan *Source code* untuk menampilkan halaman *Dashboard* aplikasi CSM. *Source code* ini berada di dalam *folder* dengan nama ***Views***.
- Menampilkan ringkasan data penting dalam bentuk *statistik*, notifikasi, dan tabel keluhan terbaru.



Gambar 1.13 *Source code* Halaman *Dashboard*

3) *Dataclient.blade.php*

- *Source code* ini merupakan halaman *User* CSM yang terdapat pada *folder* ***resources > views > dashboardadmin > DataClient.***
- *Source code* menampilkan dafatar akun *client* di halaman data *client* di aplikasi CSM.
- Melihat daftar akun *client* yang sudah didaftarkan.

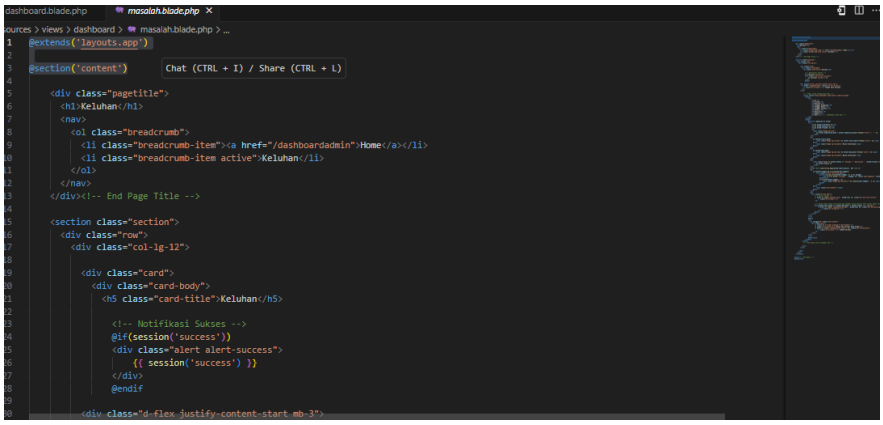


```
dashboard.blade.php    dataclient.blade.php ×
esources > views > dashboardadmin > DataClient > dataclient.blade.php > ...
1  @extends('layouts.main')
2
3  @section('content')
4
5  <section class="section">
6    <div class="row">
7      <div class="col-lg-12">
8
9        <!-- Page Title & Breadcrumb -->
10       <div class="pagetitle">
11         <h1>Data Client</h1>
12         <nav>
13           <ol class="breadcrumb">
14             <li class="breadcrumb-item"><a href="{{ route('dashboardadmin.dashboardadmin') }}">Home</a></li>
15             <li class="breadcrumb-item active">Data Client</li>
16           </ol>
17         </nav>
18       </div>
19       <!-- End Page Title -->
20
21       <div class="card">
22         <div class="card-body">
23           <h5 class="card-title">Daftar Akun Client</h5>
24           <!-- Success Message -->
25           @if(session('success'))
26             <div class="alert alert-success" role="alert">
27               {{ session('success') }}
28             </div>
29           @endif
```

Gambar 1.14 *Source code* Data *Client*

4) *Masalah.blade.php*

- *Source code* ini merupakan halaman *User* CSM yang terdapat pada *folder* ***resources* > *views* > *dashboard*.**
- Halaman ini berisi daftar keluhan dari *client*, serta menampilkan notifikasi jika terjadi aksi yang sukses.
- Menampilkan daftar **keluhan pengguna** kepada **admin**.

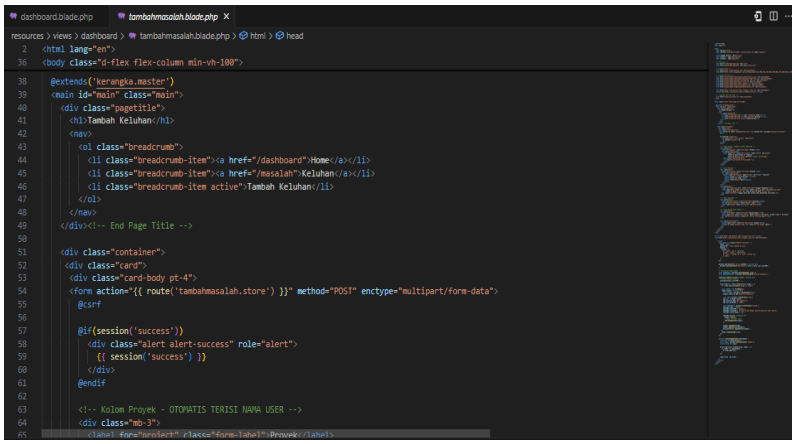


```
dashboard.blade.php    masalah.blade.php ×
ources > views > dashboard > masalah.blade.php > ...
1  @extends('layouts.app')
2
3  @section('content')        Chat (CTRL + I) / Share (CTRL + L)
4
5      <div class="pagetitle">
6        <h1>Keluhan</h1>
7        <nav>
8          <ol class="breadcrumb">
9            <li class="breadcrumb-item"><a href="/dashboardadmin">Home</a></li>
10           <li class="breadcrumb-item active">Keluhan</li>
11         </ol>
12       </nav>
13     </div><!-- End Page Title -->
14
15     <section class="section">
16       <div class="row">
17         <div class="col-lg-12">
18
19           <div class="card">
20             <div class="card-body">
21               <h5 class="card-title">Keluhan</h5>
22
23                 <!-- Notifikasi Sukses -->
24                 @if(session('success'))
25                 <div class="alert alert-success">
26                     {{ session('success') }}
27                 </div>
28                 @endif
29
30           <div class="d-flex justify-content-start mb-3">
```

Gambar 1.15 *Source code* Halaman Keluhan

5) tambahmasalah.blade.php

- *Source code* ini merupakan halaman *User* CSM yang terdapat pada *folder* ***resources > views > dashboard.***
- *Source code* ini untuk mengisi form keluhan dan mengirimkannya ke sistem.
- *Form* ini merupakan bagian dari proses input data baru ke dalam *database* keluhan yang akan diproses oleh admin.



Gambar 1.16 *Source code* Halaman tambah Keluhan

6) Projek.blade.php

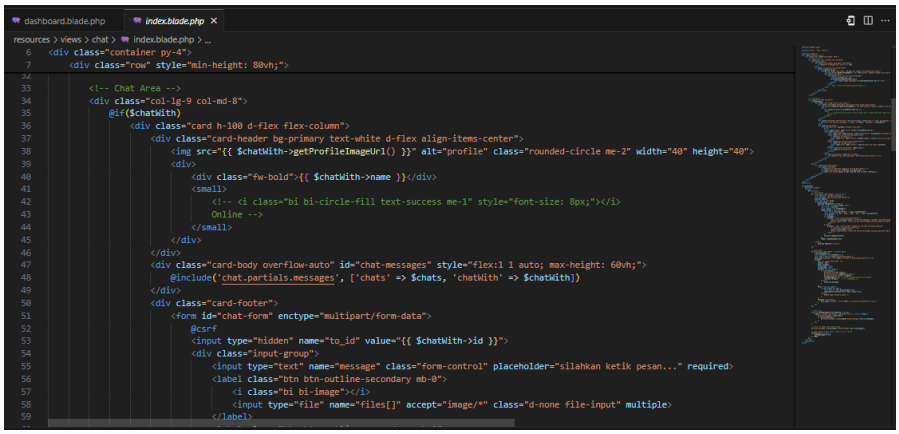
- *Source code* ini merupakan halaman *User* CSM yang terdapat pada *folder* ***resources > views > dashboard.***
- *Source code* ini merupakan halaman **tampilan daftar proyek untuk *User*** dalam aplikasi ***Customer service Manajement (CSM)***

```
dashboard.blade.php    projek.blade.php ×
resources > views > dashboard > projek.blade.php > ...
1  @extends('layouts.app')
2
3  @section('content')
4
5
6    <div class="pagetitle">
7      <h1>Projek</h1>
8      <nav>
9        <ol class="breadcrumb">
10         <li class="breadcrumb-item"><a href="/dashboard">Home</a></li>
11         <li class="breadcrumb-item active">Projek</li>
12       </ol>
13     </nav>
14   </div><!-- End Page Title -->
15
16   <section class="section">
17     <div class="row">
18       <div class="col-lg-12">
19
20         <div class="card">
21           <div class="card-body">
22             <h5 class="card-title">Projek</h5>
23
24             <!-- Table with stripped rows -->
25             <table class="table datatable table-hover table-striped">
26               <thead>
27                 <tr>
28                   <th><b>#</b></th>
29                   <th>Pelacak</th>
30                   <th>Tanggal Pengajuan</th>
```

Gambar 1. 17 *Source code* Halaman projek

7) Index.blade.php

- *Sourcece code* ini merupakan halaman *User* CSM yang terdapat pada *folder* ***resources > views > Chat.***
- *Source code* ini **menampilkan halaman percakapan** antara *User* dan admin dalam aplikasi *Customer service Manajement* (CSM).
- Mengelola komunikasi antara pengguna dengan admin.

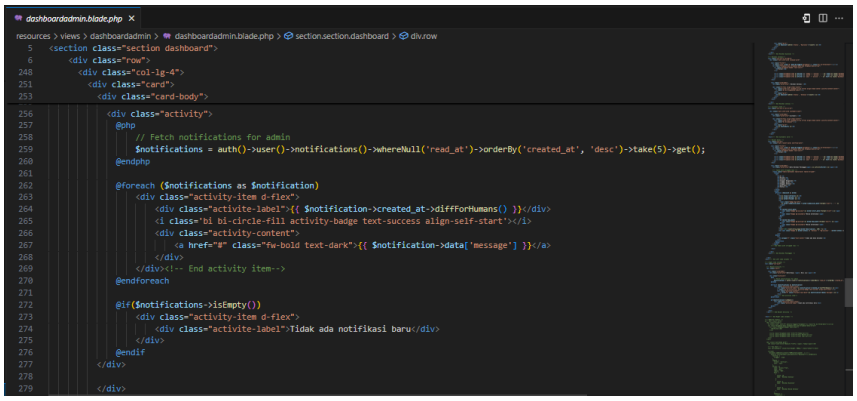


```
dashboard.blade.php    index.blade.php ×
resources > views > chat > index.blade.php > ...
6   <div class="container py-4">
7       <div class="row" style="min-height: 80vh;">
33          <!-- Chat Area -->
34          <div class="col-lg-9 col-md-8">
35              @if($chatWith)
36                  <div class="card h-100 d-flex flex-column">
37                      <div class="card-header bg-primary text-white d-flex align-items-center">
38                          <img src="{{ $chatWith->getProfileImageUrl() }}" alt="profile" class="rounded-circle me-2" width="40" height="40">
39                          <div>
40                              <div class="fw-bold">{{ $chatWith->name }}</div>
41                              <small>
42                                  <!-- <i class="bi bi-circle-fill text-success me-1" style="font-size: 8px;"></i>
43                                  Online -->
44                              </small>
45                          </div>
46                      </div>
47                      <div class="card-body overflow-auto" id="chat-messages" style="flex:1 1 auto; max-height: 60vh;">
48                          @include('chat.partials.messages', ['chats' => $chats, 'chatWith' => $chatWith])
49                      </div>
50                      <div class="card-footer">
51                          <form id="chat-form" enctype="multipart/form-data">
52                              @csrf
53                              <input type="hidden" name="to_id" value="{{ $chatWith->id }}">
54                              <div class="input-group">
55                                  <input type="text" name="message" class="form-control" placeholder="silahkan ketik pesan..." required>
56                                  <label class="btn btn-outline-secondary mb-0">
57                                      <i class="bi bi-image"></i>
58                                      <input type="file" name="files[]" accept="image/*" class="d-none file-input" multiple>
59                                  </label>
```

Gambar 1. 18 *Source code* Halaman *Chat*

8) *dashboardadmin.blade.php*

- *Source code* ini merupakan halaman *User* CSM yang terdapat pada *folder* ***resources > views > dashboardadmin***
- *Source code* ini untuk memantau seluruh aktivitas terbaru, termasuk **notifikasi** yang berkaitan dengan keluhan dari pengguna.
- Menampilkan informasi secara ringkas, *real-time*, dan *visual* di halaman utama.



Gambar 1. 19 *Source code* Halaman *Dashboard* Admin

9) projekadmin.blade.php

- *Source code* ini merupakan halaman *User* CSM yang terdapat pada *folder* ***resources > views > dashboardadmin***
- *Source code* ini untuk melihat dan mengelola seluruh daftar proyek yang ditindaklanjuti dari keluhan yang diajukan oleh *client*.
- Menampilkan data keluhan dalam bentuk **tabel interaktif.**

```
projekadmin.blade.php ×
resources > views > dashboardadmin > projekadmin.blade.php > ...
5   <section class="section">
6     <div class="row">
7       <div class="col-lg-12">
20
21        <div class="card">
22          <div class="card-body pt-4">
23            <h5 class="card-title">Daftar Projek</h5>
24
25            <!-- Tabel untuk menampilkan data -->
26            <table class="table datatable table-hover table-striped">
27              <thead>
28                <tr>
29                  <th>#</th>
30                  <th>Proyek</th>
31                  <th>Pelacak</th>
32                  <th>Tanggal Pengajuan</th>
33                  <th>Tanggal Mulai</th>
34                  <th>Tanggal Selesai</th>
35                  <th>Status</th> <!-- Tambahkan kolom status -->
36                  <th>Deskripsi</th>
37                  <th>Gambar</th>
38                  <th>Prioritas</th>
39                  <th>Aksi</th>
40                </tr>
41              </thead>
42              <tbody>
43                @forelse ($masalah as $item)
44                  <tr>
45                    <td>{{ $loop->iteration }}</td>
46                    <td>{{ $item->project }}</td>
```

Gambar 1.20 *Source code* Halaman Projek Admin

10) detailadmin.blade.php

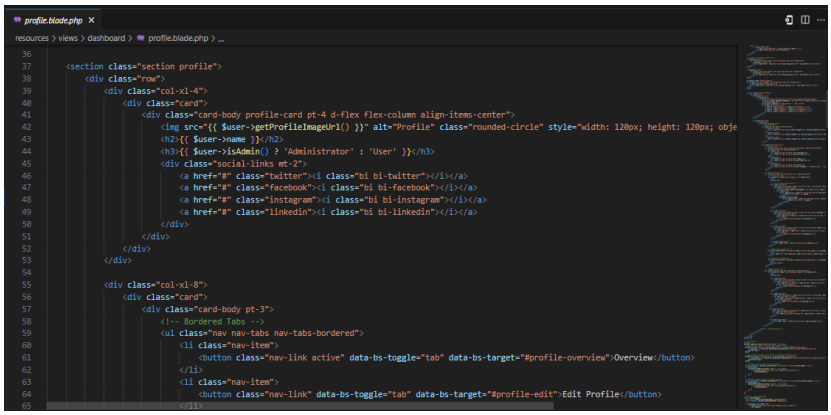
- *Source code* ini merupakan halaman *User* CSM yang terdapat pada *folder* ***resources > views > dashboardadmin.***
- *Source code* ini untuk **mengedit data keluhan** yang ditampilkan di halaman admin.
- **Memperbarui informasi proyek,** khususnya **tanggal mulai, tanggal selesai,** dan **status keluhan.**

```
detailadmin.blade.php ×
resources > views > dashboardadmin > detailadmin.blade.php > ...
15      <div class="container">
16        <div class="card">
17          <div class="card-body pt-4">
19            <form action="{{ route('admin.masalah.update', $masalah->id) }}" method="POST">
46
47            <!-- START DATE (Editable by Admin) -->
48            <div class="mb-3">
49              <label for="start_date" class="form-label">Tanggal Mulai <span class="text-danger">*</span></label>
50              <input type="date" class="form-control @error('start_date') is-invalid @enderror"
51                     name="start_date" id="start_date"
52                     value="{{ old('start_date', $masalah->start_date?->format('Y-m-d')) }}" required min="{{ date('Y-m-d') }}">
53              @error('start_date')
54                  <div class="invalid-feedback">{{ $message }}</div>
55              @enderror
56            </div>
57            <!-- DUE DATE (Editable by Admin) -->
58            <div class="mb-3">
59                <label for="due_date" class="form-label">Tanggal Selesai <span class="text-danger">*</span></label>
60                <input type="date" class="form-control @error('due_date') is-invalid @enderror"
61                       name="due_date" id="due_date"
62                       value="{{ old('due_date', $masalah->due_date?->format('Y-m-d')) }}" required min="{{ date('Y-m-d') }}">
63              @error('due_date')
64                  <div class="invalid-feedback">{{ $message }}</div>
65              @enderror
66            </div>
67
68            <!-- STATUS -->
69            <div class="mb-3">
70              <label for="status" class="form-label">Status</label>
71              <select class="form-select" name="status" id="status">
```

Gambar 1. 21 *Source code detail* Projek

11) *profile.blade.php*

- *Source code* ini merupakan halaman *User* CSM yang terdapat pada *folder* ***resources > views > dashboard.***
- *Source code* ini menampilkan informasi pribadi dari pengguna yang sedang *login*.
- Menampilkan untuk semua jenis peran pengguna, baik **admin** maupun ***client/User*.**



Gambar 1. 22 *Source code* Halaman *Profile*

b. *Configurasi Database*

1) *.env*

- Untuk menghubungkan ke *database*. Dengan nama *database* dari aplikasi ini adalah **morcs**.
- Menggunakan *connection MySQL*

```
.env
21
22  DB_CONNECTION=mysql
23  DB_HOST=127.0.0.1
24  DB_PORT=3306
25  DB_DATABASE=morcs
26  DB_USERNAME=root
27  DB_PASSWORD=
28
29  SESSION_DRIVER=database
30  SESSION_LIFETIME=120
31  SESSION_ENCRYPT=false
32  SESSION_PATH=/
33  SESSION_DOMAIN=null
34
35  BROADCAST_CONNECTION=log
36  FILESYSTEM_DISK=local
37  QUEUE_CONNECTION=database
38
39  CACHE_STORE=database
40  CACHE_PREFIX=
41
42  MEMCACHED_HOST=127.0.0.1
43
```

Gambar 1.23 *Convigurasi Database*

c. *Class Models*

1) *Chat.php*

- Menggunakan *relasi belongsTo* atau timbal balik.
- Pada *function from* menunjukkan bahwa sebuah *chat* **dimiliki oleh pengguna yang mengirim pesan** *(from_id → Users.id).*
- Pada *function to* menunjukkan bahwa sebuah *chat* **ditujukan kepada pengguna lain** *(to_id → Users.id)***.**

```php
namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Chat extends Model
{
    protected $fillable = ['from_id', 'to_id', 'message', 'files'];

    protected $casts = [
        'is_read' => 'boolean',
        'read_at' => 'datetime',
    ];


    public function from()
    {
        return $this->belongsTo(User::class, 'from_id');
    }

    public function to()
    {
        return $this->belongsTo(User::class, 'to_id');
```

Gambar 1.24 *Source code Chat.php*

2) Masalah.php

- Model ini digunakan untuk **data keluhan atau masalah** yang dilaporkan oleh pengguna dalam aplikasi CSM.
- Pada *function User* akan menampilkan data pengguna sesuai dengan *id User*.
- Menggunakan relasi *belongsTo* atau timbal balik.

```php
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Masalah extends Model
{
    use HasFactory;

    // Nama tabel jika berbeda dari konvensi 'masalahs'
    protected $table = 'tambahmasalah';


    // Kolom yang boleh diisi massal

    protected $fillable = [
        'project',
        'tracker',
        'submission_date',
        'start_date',
        'due_date',
        'description',
        'images', // tambahkan ini untuk menyimpan array path image
        'status',
        'user_id',
        'prioritas',
    ];

    // Cast kolom 'images' sebagai array (disimpan dalam bentuk JSON di database)
    protected $casts = [
    'images' => 'array',
    'submission_date' => 'date',
    'start_date' => 'date',
    'due_date' => 'date',
    ];

    public function user()
    {
        return $this->belongsTo(User::class);
    }
```

Gambar 1.25 *Source code* Masalah.php

3) *Notification.php*

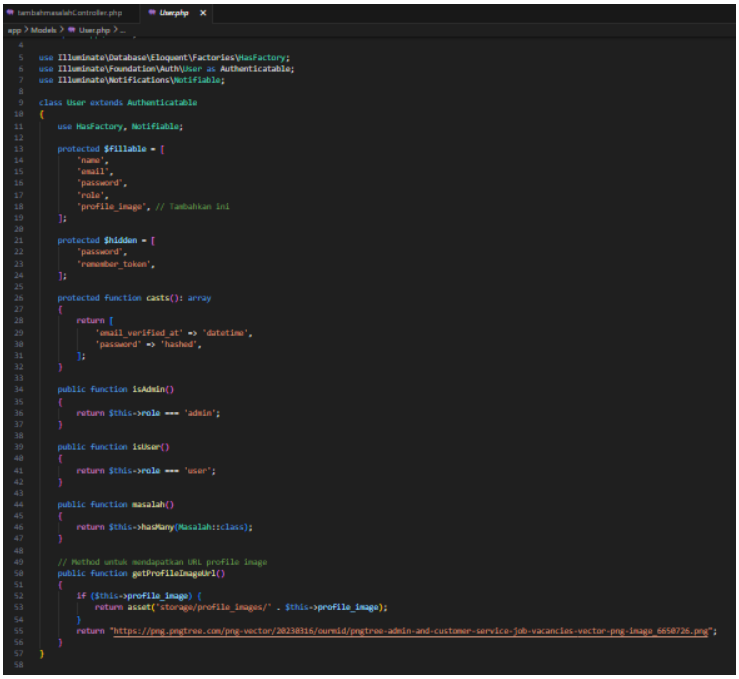
- *function notifiable* digunakan untuk **mengakses data pemilik *notifikasi.***
- Menggunakan relasi *morphTo.*

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Notifications\Notifiable;

class Notification extends Model
{
    use HasFactory, Notifiable;

    // Kolom yang dapat diisi
    protected $fillable = [
        'notifiable_id', // ID dari entitas yang menerima notifikasi
        'notifiable_type', // Tipe dari entitas yang menerima notifikasi
        'type', // Tipe notifikasi
        'data', // Data notifikasi dalam format JSON
        'read_at', // Waktu dibaca
    ];

    // Jika Anda menggunakan polymorphic relationships, Anda bisa menambahkan ini
    protected $table = 'notifications';

    // Jika Anda ingin mengakses notifiable model
    public function notifiable()
    {
        return $this->morphTo();
    }
}
```

Gambar 1.26 *Souce Code Notification.php*

4) *User.php*

- Pada *function isAdmin* mengembalikan *true* jika *User* memiliki peran sebagai admin.
- Pada *function isUser* mengembalikan *true* jika *User* memiliki peran sebagai *User* biasa/*client*.
- Pada *function* masalah menggunakan relasi *one to many (hasMany)* yang berarti satu *User* bisa memiliki banyak keluhan.
- Pada *function getProfileImageUrl* menampilkan ***URL* gambar *profile*** jika *User* telah mengunggah gambar.



```php
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;

class User extends Authenticatable
{
    use HasFactory, Notifiable;

    protected $fillable = [
        'name',
        'email',
        'password',
        'role',
        'profile_image', // Tambahkan ini
    ];

    protected $hidden = [
        'password',
        'remember_token',
    ];

    protected function casts(): array
    {
        return [
            'email_verified_at' => 'datetime',
            'password' => 'hashed',
        ];
    }

    public function isAdmin()
    {
        return $this->role === 'admin';
    }

    public function isUser()
    {
        return $this->role === 'user';
    }

    public function masalah()
    {
        return $this->hasMany(Masalah::class);
    }

    // Method untuk mendapatkan URL profile image
    public function getProfileImageUrl()
    {
        if ($this->profile_image) {
            return asset('storage/profile_images/' . $this->profile_image);
        }
        return "https://png.pngtree.com/png-vector/20230316/ourmid/pngtree-admin-and-customer-service-job-vacancies-vector-png-image_6658726.png";
    }
}
```

Gambar 1.27 *Source code User.php*

d. *Class Controller*

1) *AuthController.php*

- *Source code* ini merupakan halaman CSM yang terdapat pada *folder* ***app > Http > Controller.***
- Terdapat *function showlogin* untuk menampilkan **halaman *login*** dengan mengembalikan *view*.
- Terdapat *function login* untuk menangani proses *login* saat *form* dikirim.
- *Source code* pada halaman *AuthController.php* digunakan untuk menangani proses ***login* pengguna.**

```
tambahmasalahController.php    AuthController.php ×
app > Http > Controllers > AuthController.php > AuthController > login
6   use Illuminate\Support\Facades\Auth;
7   use App\Models\User;
8   use Illuminate\Support\Facades\Hash;
9
10  class AuthController extends Controller
11  {
12      public function showLogin()
13      {
14          return view('auth.login');
15      }
16
17      public function login(Request $request)
18      {
19          // Validasi data
20          $request->validate([
21              'email' => 'required|string|max:255',
22              'password' => 'required|string|min:8',
23          ]);
24
25          // Cari user berdasarkan NIM
26          $user = User::where('email', $request->email)->first();
27
28          if ($user && Hash::check($request->password, $user->password)) {
```

Gambar 1.28 *Source code AuthController.php*

2) *ChatController.php*

- *Source code* ini merupakan halaman CSM yang terdapat pada *folder* ***app > Http > Controller.***
- Terdapat *function fetch* untuk menampilkan **daftar semua pesan termasuk isi *file*** dari *User* dan admin agar bisa ditampilkan di *frontend.*
- Terdapat *function send* untuk menunjukkan proses **menyimpan data *chat*** baru ke *database*
- *Source code* pada halaman *ChatController.php* digunakan untuk mengatur tampilan percakapan antara pengguna.

```
tambahmasalahController.php    ChatController.php 3 ×
app > Http > Controllers > ChatController.php > ChatController > fetch
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Models\Chat;
7  use App\Models\User;
8  use Auth;
9
10 class ChatController extends Controller
11 {
12     public function index($userId = null)
13     {
14         $user = Auth::user();
15
16         $users = $user->isAdmin()
17             ? User::where('id', '!=', $user->id)->get()
18             : User::where('role', 'admin')->get();
19
20         $chatWith = $userId ? User::findOrFail($userId) : null;
21
22         // Tandai sebagai terbaca
23         if ($chatWith) {
24             Chat::where('from_id', $chatWith->id)
25                 ->where('to_id', $user->id)
26                 ->where('is_read', false)
27                 ->update([
28                     'is_read' => true,
29                     'read_at' => now(),
30                 ]);
```

Gambar 1.29 *Source code ChatController.php*

3) *DashboardController.php*

- Menggunakan relasi hasMany untuk mengambil keluhan yang dibuat oleh *User* tersebut.
- *Source code* pada halaman *DahsboardtController.php* digunakan untuk menampilkan data keluhan yang dilaporkan oleh pengguna yang sedang *login*, dan menyaring data berdasarkan waktu.
- Menampilkan daftar keluhan *User* yang sedang *login.*

```php
tambahmasalahController.php    dashboardController.php 2 ×
app > Http > Controllers > dashboardController.php > dashboardController > index
use Illuminate\Http\Request;
use App\Models\masalah; // Tambahkan ini
use Carbon\Carbon;

class dashboardController extends Controller
{
    public function index(Request $request)
    {
        $user = auth()->user();

        // Get filter parameter from request
        $filter = $request->get('filter', 'today');

        // Base query
        $query = $user->masalah();

        // Apply filter
        switch($filter) {
            case 'today':
                $query->whereDate('submission_date', Carbon::today());
                break;
            case 'month':
                $query->whereMonth('submission_date', Carbon::now()->month)
                      ->whereYear('submission_date', Carbon::now()->year);
                break;
            case 'year':
                $query->whereYear('submission_date', Carbon::now()->year);
                break;
        }

        // Menggunakan relationship untuk mengambil masalah user
        $masalah = $query->orderBy('created_at', 'desc')->get();

        // Ambil 5 notifikasi terbaru yang belum dibaca untuk pengguna yang sedang login
        $notifications = auth()->user()->notifications()->whereNull('read_at')->orderBy('created_at', 'desc')->take(5)->get();

        // Kirim data masalah dan notifikasi ke view
        return view('dashboard.dashboard', compact('masalah', 'notifications', 'filter'));
```

Gambar 1.30 *Source code DahsboardtController.php*

4) *dataclientController.php*

- Terdapat CRUD untuk admin menambahkan data *client*, mengedit, mengirim, mengupdate, menghapus data*cient*.
- terdapat 6 function CRUD seperti *function index*, *function create*, *function store*, *function* edit, *function update*, *function destroy* / hapus.
- *Source code* pada halaman *dataclientController.php* untuk mengelola data *client* yang akan dimasukan.

```php
use App\Models\User;
use App\Models\masalah; // Pastikan untuk mengimpor model masalah
use Illuminate\Http\Request;

class dataclientController extends Controller
{
    // Menampilkan semua user dengan role 'user'
    public function index()
    {
        $clients = User::where('role', 'user')->get();

        // Ambil 5 masalah terbaru untuk notifikasi
        $recentIssues = masalah::orderBy('created_at', 'desc')->take(5)->get();
        // Ambil notifikasi untuk admin
        $notifications = auth()->user()->notifications()->whereNull('read_at')->get(); // Ambil notifikasi yang belum dibaca

        // Kirim data ke view
        return view('dashboardadmin.DataClient.dataclient', compact('clients', 'recentIssues', 'notifications'));
    }

    // Tampilkan form tambah client
    public function create()
    {
        // Ambil notifikasi untuk admin
        $notifications = auth()->user()->notifications()->whereNull('read_at')->get(); // Ambil notifikasi yang belum dibaca
        return view('dashboardadmin.DataClient.tambahclient', compact('notifications'));
    }

    // Simpan data client baru
    public function store(Request $request)
    {
        $request->validate([
            'name' => 'required|string|max:255',
            'email' => 'required|string|email|max:255|unique:users',
            'password' => 'required|string|min:6',
        ]);

        User::create([
```

Gambar 1.31 *Source code dataclientController.php*

5) *tambahmasalahController.php*

- Terdapat CRUD untuk admin menambahkan data *client*, mengedit, mengirim, meng*update*, meyimpan keluhan.
- Terdapat *function index, function store, function* edit, *function update*.
- Terdapat *function getPrioritas untuk* memanggil *API* prediksi prioritas keluhan berdasarkan deskripsi.
- Mengirim data ke *endpoint* melalui ***HTTP POST*** menggunakan *GuzzleHttp Client* untuk **menentukan prioritas keluhan secara otomatis.**

```php
use Illuminate\Http\Request;
use App\Models\Masalah;
use Illuminate\Support\Facades\Auth;
use App\Models\Notification;
use App\Notifications\NewIssueNotification;
use App\Models\User;
use GuzzleHttp\Client;
use Carbon\Carbon;
class tambahmasalahController extends Controller
{
    /**
     * Tampilkan form tambah masalah
     */

    public function index()
    {
        $user = Auth::user(); // Ambil data user yang sedang login
        return view('dashboard.tambahmasalah', compact('user'));
    }


    public function getPrioritas($deskripsi)
    {
        if (empty($deskripsi)) {
            return 'Tidak ada deskripsi';
        }

        $client = new Client();
        $url = 'http://192.168.43.119:5000/prediksi-keluhan';

        try {
            $response = $client->post($url, [
                'json' => ['deskripsi' => $deskripsi]
            ]);
            $data = json_decode($response->getBody(), true);
            return $data['hasil_prediksi_prioritas'] ?? 'Tidak diketahui';
        } catch (\Exception $e) {
            return 'Error: ' . $e->getMessage();
```

Gambar 1.32 *tambahmasalahController.php*

6) *ImageUploadController.php*

- *File* disimpan secara otomatis menggunakan *Laravel Storage* dengan *disk public.*
- *Source code ImageUploadController.php* untuk **meng-*handle* proses *upload* gambar** dari pengguna ke *server*.
- *Source code* digunakan *upload* gambar dari *form* dan menyimpan *file* ke *storage*.

```php
ImageUploadController.php ×
app > Http > Controllers > ImageUploadController.php > ...
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Storage;

class ImageUploadController extends Controller
{
    public function upload(Request $request)
    {
        if ($request->hasFile('images')) {
    foreach ($request->file('images') as $file) {
        $path = $file->store('uploads', 'public');
        // Simpan path ke database jika diperlukan, contoh:
        // MasalahImage::create(['masalah_id' => $masalah->id, 'path' => $path]);
            }
        }
    }
}
```

Gambar 1.33 *Source code ImageUploadController.php*

7) *masalahController.php*

- *masalahController.php* ini menangani proses CRUD, crate, read, update, *delete* gambar, *upload file*, dan prediksi prioritas.
- Terdapat 10 *function* CRUD seperti *function index*, *function uploadImage*, *function show*, *function showAdmin*, *function getPrioritas*, *function updateAdmin*, *function* edit, *function update*, *function updateStatus*, *functiondeleteImage*.

```
app > Http > Controllers > masalahController.php > masalahController > index
namespace App\Http\Controllers;

use App\Models\masalah;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Storage;
use Illuminate\Support\Facades\Auth;
use Carbon\Carbon;
use GuzzleHttp\Client; //menentukan prioritas keluhan secara otomatis

class masalahController extends Controller
{
    public function index()
    {
        $user = auth()->user();

        // Menggunakan relationship untuk mengambil masalah user
        $masalah = $user->masalah()->where('status', 'Belum selesai')->orderBy('created_at', 'desc')->get();
        return view('dashboard.masalah', compact('masalah'));
    }

    public function uploadImage(Request $request)
    {
        if ($request->hasFile('images')) {
            $paths = [];
            foreach ($request->file('images') as $file) {
                $path = $file->store('uploads', 'public');
                $paths[] = $path;
```

Gambar 1.34 Source code masalahController.php

8) *NotificationController.php*

- *Source code NotificationController.php* digunakan untuk menangani penandaan notifikasi sebagai telah dibaca dalam sistem *Laravel Notification*.
- Terdapat *function markAllAsRead* untuk *redirect* kembali ke halaman sebelumnya dengan pesan sukses.
- Terdapat *function markAsRead* untuk kebalikannya *redirect* kembali ke halaman sebelumnya.

```
app > Http > Controllers > NotificationController.php > NotificationController > markAsRead
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class NotificationController extends Controller
{
    public function markAllAsRead()
    {
        $user = Auth::user();
        $user->notifications()->whereNull('read_at')->update(['read_at' => now()]);

        return redirect()->back()->with('success', 'Semua notifikasi telah ditandai sebagai dibaca');
    }

    public function markAsRead($id)
    {
        $notification = Auth::user()->notifications()->find($id);

        if ($notification) {
            $notification->update(['read_at' => now()]);
        }

        return redirect()->back();
    }
```

Gambar 1.35 *Source code NotificationController.php*

9) *profileadminController.php*

- *Controller* ini menangani fitur pengelolaan *profile* dalam aplikasi.
- Terdapat *function index*, *function update*, *function updateImage*, *function* deleteImage, *function* updatePassword.

```
app > Http > Controllers > profileController.php > profileController > update

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Storage;
use Illuminate\Validation\Rule;

class profileController extends Controller
{
    public function index()
    {
        $user = Auth::user();
        return view('dashboard.profile', compact('user'));
    }

    public function update(Request $request)
    {
        $user = Auth::user();

        $request->validate([
            'name' => 'required|string|max:255',
            'email' => [
                'required',
                'email',
                Rule::unique('users')->ignore($user->id),
```

Gambar 1. 36 Source Code profileadminController.php

10) *projekadminController.php*

- Digunakan sebagai halaman utama admin untuk memantau dan menindaklanjuti laporan dari pengguna.
- *Source code* untuk **mengelola tampilan daftar proyek di sisi admin.**

```
app > Http > Controllers > projekadminController.php > projekadminController > index
1   <?php
2
3   namespace App\Http\Controllers;
4
5   use Illuminate\Http\Request;
6   use Illuminate\Support\Facades\Storage;
7   use App\Models\masalah;
8
9   class projekadminController extends Controller
10  {
11      public function index()
12      {
13          // Cek role admin
14          if (auth()->user()->role !== 'admin') {
15              abort(403);
16          }
17          $masalah = masalah::orderBy('created_at', 'desc')->get();
18          // Ambil notifikasi untuk admin
19          // Ambil 5 notifikasi untuk admin yang belum dibaca
20          $notifications = auth()->user()->notifications()->whereNull('read_at')->orderBy('created_at', 'desc')->tak
21          $recentIssues = masalah::orderBy('created_at', 'desc')->take(5)->get();
22          return view('dashboardadmin.projekadmin', compact('masalah', 'recentIssues', 'notifications'));
23      }
24  }
25
```

Gambar 1. 37 *Source code projekadminController.php*

11) *registerController.php*

- Sebagai peran utama halaman proyek admin, menggabungkan fitur pemantauan keluhan, notifikasi, dan kontrol akses berbasis peran (*role*).
- Terdapat 2 *function* yaitu *function index* dan *function store*.

```
app > Http > Controllers > registerController.php > registerController > store
5   use Illuminate\Http\Request;
6   use App\Models\User;
7   class registerController extends Controller
8   {
9       public function index()
10      {
11          return view('auth.register');
12      }
13
14      public function store(Request $request)
15      {
16          $request->validate([
17
18              'name' => 'required|string|max:255',
19              'email' => 'required|string|email:dns|unique:users',
20              'password' => 'required|string|min:5'
21          ]);
22
23          User::create([
24
25              'name' => $request->name,
26              'email' => $request->email,
27              'password' => bcrypt($request->password)
28
29          ]);
30
31          return to_route('login');
```

Gambar 1.38 *Source code registerController.php*

12. PrediksiKeluhanAPI.py

- Digunakan sebagai *endpoint API* untuk memproses dan memprediksi prioritas keluhan pengguna berdasarkan deskripsi teks yang dikirimkan.
- *Source code* untuk mengambil input teks, memprosesnya melalui model *SVM* dan alat bantu (*vectorizer* dan *label encoder*), lalu mengembalikan hasil prediksi prioritas dalam format *JSON response.*

```python
from flask import Flask, request, jsonify
import joblib

app = Flask(__name__)

# Load model dan alat bantu
model = joblib.load("modelbaru/svm_model.pkl")
vectorizer = joblib.load("modelbaru/vectorizer.pkl")
encoder = joblib.load("modelbaru/label_encoder.pkl")

@app.route("/prediksi-keluhan", methods=["POST"])
def prediksi_keluhan():
    try:
        data = request.get_json()
        deskripsi = data["deskripsi"]

        # Proses deskripsi
        X = vectorizer.transform([deskripsi])
        y_pred = model.predict(X)
        label = encoder.inverse_transform(y_pred)

        return jsonify({
            "deskripsi": deskripsi,
            "hasil_prediksi_prioritas": label[0]
        })
    except Exception as e:
        return jsonify({"error": str(e)}), 400

if __name__ == "__main__":
    app.run(host='192.168.52.119', port=5000, debug=True)
```

Gambar 1.39 kode Prediksi API