## BAB 3 PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Koordinasi

Selama kerja magang di PT Rembon Karya Digital, penugasan difokuskan pada divisi Backend dan Frontend Development, yang berada di bawah arahan Backend Leader dan Frontend Leader. Proyek utama yang dikerjakan mencakup pengembangan sistem *email blast* berbasis Laravel, yang meliputi perancangan fitur pengelolaan data pelanggan serta integrasi sistem dengan fitur *email blast* guna mendukung efisiensi dalam penyampaian informasi kepada pelanggan.

Seluruh aktivitas kerja berada di bawah bimbingan Bapak Fabio Viandino sebagai supervisor dan Bapak Aang Wiadi selaku Direktur PT Rembon Karya Digital. Koordinasi dilaksanakan secara daring melalui platform Trello, yang dimanfaatkan untuk memantau perkembangan tugas serta progres kerja. Setiap pembaruan dilakukan dengan memperbarui status tugas pada papan kerja yang terdiri atas kategori *To Do, In Progress, Review*, dan *Done*, dan akan memperoleh umpan balik langsung dari Bapak Aang. Jika diperlukan, pertemuan secara langsung di kantor diselenggarakan, khususnya untuk keperluan pembagian tugas baru atau pengurusan administrasi tertentu. Sementara itu, arahan teknis dan supervisi rutin diberikan oleh Bapak Fabio, yang juga bertanggung jawab terhadap validasi administratif selama berlangsungnya program kerja.

#### 3.2 Tugas yang Dilakukan

Selama masa pelaksanaan kerja magang di PT Rembon Karya Digital, terdapat sejumlah tanggung jawab dan tugas yang diberikan dalam lingkup pengembangan sistem internal berbasis Laravel. Adapun tugas-tugas yang dilaksanakan, antara lain:

- 1. Membangun sistem manajemen data pelanggan.
- 2. Mengembangkan fitur pengelolaan gelar/title pelanggan.
- 3. Mengimplementasikan fitur *import* data pelanggan dari file Excel ke *database*, yang kemudian ditampilkan di halaman data pelanggan
- 4. Mengintegrasikan data pelanggan dengan sistem email blast.

5. Melakukan pengujian, dokumentasi, dan perbaikan fitur.

## 3.3 Uraian Pelaksanaan Magang

Selama pelaksanaan kerja magang di PT Rembon Karya Digital, beberapa tugas dikerjakan dalam posisi sebagai *Backend dan Frontend Development Intern*. Kegiatan yang dilaksanakan setiap minggu dapat ditemukan pada Tabel 3.1.

Minggu Ke-	Pelaksanaan					
1–3	Melakukan analisis kebutuhan desain Queue Management System					
	serta pembuatan desain pertama, kedua, dan ketiga untuk Queue					
	Management System.					
4	Melakukan analisis terhadap sistem <i>Email Blast</i> perusahaan serta mempelajari beberapa <i>tools</i> baru yang digunakan.					
5	5 Membuat <i>migrations</i> dan <i>model</i> untuk fitur data pelanggan.					
6–8 Mengembangkan dan melanjutkan <i>Customer Controller</i> , mencakup fungsi <i>index</i> , <i>store</i> , <i>edit</i> , <i>update</i> , <i>destroy</i> .						
9 Memperbaiki beberapa validasi unggah file Excel, validasi <i>duplica</i> <i>email</i> , serta integrasi halaman data pelanggan ke halaman <i>Email Bla</i>						
10	10 Menguji fitur unggah file Excel untuk memastikan data masuk ke basis data dan tampil pada halaman data pelanggan.					
11	Melakukan pengujian skenario untuk integrasi pemilihan data pelanggan dalam halaman <i>Email Blast</i> .					
12	Melakukan perbaikan tampilan pada integrasi data pelanggan di halaman <i>Email Blast</i> .					
13 Melakukan <i>clean up code</i> , menambahkan komentar, serta membu dokumentasi untuk laporan supervisor dan direktur.						
14	14 Melakukan penyesuaian dengan menambahkan dan menghapus beberapa pesan <i>feedback</i> sesuai kebutuhan.					
15	Menambahkan beberapa fitur minor dan validasi tambahan.					
16–19	Melakukan pengujian akhir, pengecekan keseluruhan sistem, dan mengunggah hasil akhir ke repositori GitHub perusahaan.					

Tabel 3.1. Pekerjaan Selama Pelaksanaan Kerja Magang

#### **Proses Pengembangan Sistem** 3.4



Gambar 3.1. Flowchart Keseluruhan Sistem Email Blast

Gambar 3.1 menampilkan diagram alur (flowchart) yang memberikan gambaran menyeluruh mengenai proses utama dalam sistem yang digunakan dan dikembangkan selama kegiatan magang. Flowchart ini menunjukkan bagaimana pengguna berinteraksi dengan sistem, dimulai dari proses login hingga navigasi menuju berbagai menu seperti data pengguna, blast email, daftar email, template email, dan data pelanggan. Setiap menu mengarahkan pengguna pada fitur-fitur fungsional tertentu, antara lain pengelolaan data menggunakan metode CRUD, personalisasi isi email melalui penggunaan placeholder, serta pengaturan dan pemilihan penerima email. Dalam alur ini juga ditunjukkan modul-modul utama yang menjadi fokus pengembangan, khususnya pada bagian *Blast Email* dan Data Pelanggan.

Pembahasan lebih rinci terhadap modul-modul yang tertera pada diagram alur akan dipaparkan pada subbab berikut, dengan fokus pada modul yang berkaitan langsung dengan pengembangan yang telah dilakukan, yaitu pada bagian *Blast Email* dan Data Pelanggan.

#### 3.4.1 Manajemen Data Pelanggan

#### A Analisis Kebutuhan

Dalam proses pengembangan sistem manajemen data pelanggan, diperlukan sebuah halaman khusus yang menampilkan daftar pelanggan dalam format tabel yang interaktif dan informatif. Tabel tersebut harus dilengkapi dengan fitur *pagination* yang menyediakan pilihan jumlah entri per halaman, yaitu 10, 25, 50, atau 100 data. Selain itu, fitur pencarian menjadi komponen esensial, yang memungkinkan pengguna melakukan pencarian berdasarkan gelar, nama lengkap, alamat email, atau tanggal pendaftaran (*created date*).

Setiap baris data dalam tabel harus dilengkapi dengan tombol aksi seperti *Create*, Detail, *Edit*, dan *Delete* untuk mempermudah pengelolaan data secara langsung dari halaman tersebut. Sistem juga diharapkan dapat menampilkan informasi mengenai jumlah data yang sedang ditampilkan, misalnya "Showing 1 to 10 of 100 entries," serta menyediakan navigasi *Previous* dan *Next* di bagian bawah tabel untuk memudahkan perpindahan antar halaman.

Dari sisi tampilan, halaman ini harus menyesuaikan dengan desain antarmuka *frontend* lainnya untuk menjaga konsistensi visual dan pengalaman pengguna. Keberadaan halaman ini yang terintegrasi langsung dengan tabel customers pada basis data menjadi aspek penting dalam mendukung kemudahan pengelolaan data pelanggan, pemantauan informasi, serta integrasi lanjutan dengan sistem *email blast* yang telah digunakan perusahaan.

## M U L T I M E D I A N U S A N T A R A

#### **B** Perancangan Basis Data

Nama Kolom	Tipe Data	NULL	Deskripsi
id	bigint(20) unsigned	Tidak	Sebagai primary key, bernilai
			unik dan auto increment untuk
			mengidentifikasi setiap pelanggan.
title_id	bigint(20) unsigned	Ya	Merujuk pada relasi ke tabel lain
			yang menyimpan informasi gelar
			pelanggan.
name	varchar(255)	Tidak	Menyimpan nama lengkap
			pelanggan.
email	varchar(255)	Tidak	Menyimpan alamat email
			pelanggan.
deleted_at	timestamp	Ya	Digunakan untuk fitur soft delete,
			menandai waktu saat data dihapus.
created_at	timestamp	Ya	Menyimpan waktu pembuatan data
			pelanggan, diisi otomatis oleh
			sistem.
updated_at	timestamp	Ya	Mencatat waktu pembaruan terakhir
			pada data pelanggan.

Tabel 3.2. Struktur Tabel customers pada Sistem Email Blast

Desain struktur tabel customers yang disajikan pada Tabel 3.2, berperan sebagai elemen utama dalam struktur basis data untuk fitur manajemen data pelanggan pada sistem *Email Blast*. Tabel ini merekam informasi penting dari setiap pelanggan, seperti gelar (melalui relasi title\_id), nama lengkap, dan alamat email sebagai sasaran pengiriman pesan. Kolom id yang bertipe bigint unsigned digunakan sebagai *primary key* dan diatur untuk bertambah otomatis (*auto-increment*), untuk menangani volume data dalam jumlah besar secara efisien.

Selain menyimpan data utama, tabel ini juga menyediakan kolom created\_at, updated\_at, dan deleted\_at untuk mencatat waktu pembuatan, pembaruan, dan penghapusan data dengan metode *soft delete*, sehingga riwayat data tetap terdokumentasi dengan baik. Desain struktur tabel disusun untuk mendukung operasi *CRUD* (*Create*, *Read*, *Update*, *Delete*) secara optimal serta memungkinkan integrasi langsung dengan fitur pencarian, penyaringan data, dan *import* data pelanggan dari file Excel.

## C Flowchart Proses CRUD Data Pelanggan



Gambar 3.2. Flowchart Proses CRUD Data Pelanggan

Flowchart pada Gambar 3.2 menjelaskan secara menyeluruh alur kerja fitur manajemen data pelanggan dalam sistem *Email Blast*. Proses dimulai ketika pengguna berhasil melakukan login dan mengakses halaman khusus data pelanggan, yang menampilkan informasi dalam bentuk indeks tabel. Sistem secara otomatis mengirimkan permintaan *GET* ke basis data customers untuk mengambil dan menampilkan data yang tersedia. Jika data belum tersedia, pengguna dapat menambahkan entri baru melalui tombol *Create* dengan mengisi informasi berupa gelar, nama, dan alamat email. Setelah melalui proses validasi, data akan disimpan ke basis data menggunakan metode *POST*.

Untuk melihat rincian pelanggan, pengguna dapat menggunakan tombol Detail, yang akan mengambil data terkait dan menampilkannya secara lengkap. Dari halaman detail tersebut, tersedia pula opsi untuk memperbarui informasi pelanggan melalui tombol *Edit*. Jika perubahan telah dilakukan dan berhasil divalidasi, sistem akan memperbarui data di basis data dengan metode *POST Update*. Selain itu, terdapat fitur penghapusan data melalui tombol *Delete*, yang akan menampilkan kotak konfirmasi sebelum melanjutkan proses *soft delete* dengan mencatat waktu penghapusan di kolom deleted\_at pada basis data. Setelah menyelesaikan setiap proses untuk penambahan, pembacaan, penyuntingan, maupun penghapusan, sistem akan secara otomatis menampilkan kembali halaman indeks pelanggan agar pengguna dapat langsung melihat hasil perubahan.

#### **D** Implementasi

Halaman manajemen data pelanggan memiliki peran penting karena menjadi pusat kendali seluruh data pelanggan yang nantinya digunakan dalam proses pengiriman email massal *email blast*). Tampilan utama dari fitur ini berupa halaman *index* yang menampilkan data pelanggan dalam format tabel interaktif. Pengguna dapat melakukan pencarian berdasarkan kolom gelar, nama, alamat email, dan tanggal pembuatan data. Seluruh kolom tersebut juga dapat diurutkan secara menaik (*ascending*) maupun menurun (*descending*). Selain itu, tersedia opsi untuk menyesuaikan jumlah data yang ditampilkan per halaman, yaitu 10, 25, 50, atau 100 entri, serta navigasi berupa tombol *Previous* dan *Next* untuk berpindah halaman. Informasi mengenai rentang data yang sedang ditampilkan juga turut disediakan. Seluruh tampilan halaman ini telah dirancang agar selaras dengan antarmuka pengguna lainnya guna menjaga konsistensi visual dalam sistem.

Devel Blact	Remb	on Karya Digit	al					9
DASHBOARD	Dashboo	ard / Data Pelar	iggan					
<ul> <li>Bashboard</li> <li>Data Pengguna</li> <li>Blast Email</li> </ul>	Data F	Pelanggan						
🖽 List Blast	Show	10 v entries	\$			Search:		
III Template	#	GELAR	NAMA	÷ EMAIL	CREATED DATE		OPSI	
💄 Data Pelanggan	1	Bapak	oyen	hi@gmail.com	11 June 2025	Deta	Edit Delete	
	2	Saudara	Bambang	hii@gmail.com	11 June 2025	Deta	Edit Delete	
	3	lbu	intern	angelina5@student.umn.ac.id	10 June 2025	Deta	Edit Delete	
	Showin	g 1 to 3 of 3 entri	95				Previous 1	Next
_						Copyright ©	2025, Rembon K	arya Di

Gambar 3.3. Tampilan Indexing Halaman Data Pelanggan

Pada halaman indeks seperti pada Gambar 3.3, terdapat tombol *Create* yang akan mengarahkan pengguna ke halaman formulir penambahan pelanggan baru. Formulir ini terdiri dari beberapa isian, yaitu gelar (dalam bentuk *dropdown* yang mengambil data dari tabel titles), nama lengkap, dan alamat email pelanggan. Setelah pengguna mengisi seluruh data pada form dan mengirimkannya, sistem akan menjalankan proses validasi untuk memastikan bahwa setiap input memenuhi ketentuan yang telah ditentukan. Validasi ini meliputi keharusan pengisian nama dan email, serta pemeriksaan terhadap format email agar sesuai standar. Selain itu, sistem juga akan memastikan bahwa alamat email belum pernah digunakan sebelumnya, kecuali yang sudah terhapus secara lunak (*soft delete*).

Apabila semua data lolos validasi, maka informasi tersebut akan disimpan ke dalam tabel customers melalui pemanggilan fungsi store() pada CustomerController. Berikut ini merupakan potongan kode yang digunakan pada sisi controller:

```
public function store(Request $request)
 {
      $request -> validate([
3
          'title_id' => 'nullable|exists:titles,id',
4
          'name' => 'required|string|max:255',
          'email' => [
6
              'required',
7
              'email',
8
              Rule::unique('customers', 'email') ->whereNull('
9
     deleted_at'),
         ],
10
11
      ]);
```

```
12
13 Customer::create([
14 'title_id' => $request->title_id,
15 'name' => $request->name,
16 'email' => $request->email,
17 ]);
18
19 return redirect()->route('admin.customers.index')->with('
success', 'Data pelanggan berhasil ditambahkan.');
20 }
```

Kode 3.1: Fungsi Store pada CustomerController

Ilustrasi tampilan halaman form tambah data pelanggan dapat dilihat pada Gambar 3.4.

Tambah Data Pelanggan		
GELAR		NAMA PELANGGAN *
Pilih Gelar (Opsional)	~	Masukkan Nama Pelanggan
ALAMAT EMAIL *		
Masukkan Alamat Email		
		🛱 Simpan Data Pelanggan

Gambar 3.4. Tampilan Halaman Form Create Data Pelanggan

Setelah data pelanggan berhasil ditambahkan, sistem menyediakan fitur lanjutan untuk menampilkan informasi lengkap dari masing-masing entri. Fitur detail bertujuan memberikan akses detail terhadap data pelanggan yang sudah tersimpan, sehingga pengguna dapat melakukan pemeriksaan atau verifikasi sebelum melakukan tindakan lebih lanjut. Fitur tampilan detail pelanggan berfungsi untuk menyajikan informasi lengkap dari satu entri data pelanggan secara terstruktur. Fitur ini dapat diakses melalui halaman *indexing* data pelanggan, di mana pengguna cukup menekan tombol "Detail" pada salah satu baris data. Setelah itu, sistem akan mengeksekusi permintaan untuk mengambil data pelanggan berdasarkan *primary key* berupa id.

Data yang diperoleh kemudian ditampilkan dalam format tabel deskriptif, mencakup informasi seperti gelar, nama pelanggan, nama lengkap (gabungan antara gelar dan nama), alamat email pelanggan, tanggal data dibuat, serta tanggal data diperbarui. Penyajian ini memudahkan pengguna dalam membaca dan memverifikasi data tanpa perlu berpindah ke halaman pengeditan. Berikut merupakan potongan kode pada controller yang menangani proses pengambilan dan pengiriman data ke tampilan halaman detail:

```
public function show(Customer $customer)
{
    $customer->load('title');
    return view('admin.customers.show', compact('customer'));
}
```

Kode 3.2: Fungsi Show Untuk Menampilkan Detail Pelanggan

Ilustrasi implementasi tampilan detail data pelanggan dapat dilihat pada Gambar 3.5.

Detail Data Pelanggan	
Gelar	Bapak
Nama Pelanggan	oyen
Nama Lengkap	Bapak oyen
Alamat Email	hi@gmail.com
Tanggal Dibuat	11 June 2025 12:45:35
Tanggal Diperbarui	11 June 2025 13:04:11
Edit Kembali	

Gambar 3.5. Tampilan Halaman Detail Data Pelanggan

Setelah pengguna melihat informasi lengkap dari pelanggan tertentu melalui halaman *Detail*, sistem juga menyediakan opsi untuk memperbarui data melalui fitur *Edit*. Fitur ini memungkinkan pengguna untuk mengubah informasi pelanggan yang sudah tersimpan sebelumnya. Akses terhadap fitur ini dilakukan melalui tombol *Edit* yang tersedia di halaman indeks data pelanggan pada Gambar 3.3. Setelah tombol tersebut ditekan, sistem akan menampilkan formulir yang telah terisi dengan data yang sudah ada. Hal ini mempermudah pengguna dalam melakukan penyesuaian tanpa perlu menginput ulang seluruh informasi.

Sebelum data yang diperbarui disimpan ke dalam basis data, sistem akan menjalankan proses validasi untuk memastikan keakuratan dan kelengkapan data yang diinput. Apabila seluruh data telah valid, proses pembaruan akan dilakukan menggunakan method update() pada file CustomerController. Potongan kode berikut menunjukkan bagaimana sistem menangani proses pembaruan data pelanggan:

public function update(Request \$request, Customer \$customer)

```
2
      $request -> validate([
          'title_id' => 'nullable|exists:titles,id',
           'name' => 'required|string|max:255',
5
          'email' => [
6
               'required',
               'email',
8
               Rule::unique('customers', 'email')
9
                   ->ignore($customer->id)
10
                   ->whereNull('deleted_at'),
11
          ],
      ]);
14
      $customer->update([
15
          'title_id' => $request->title_id,
16
          'name' => $request->name,
17
           'email' => $request->email,
18
19
      ]);
20
      return redirect()->route('admin.customers.index')->with('
21
     success', 'Data pelanggan berhasil diperbarui.');
22
```

Kode 3.3: Fungsi Edit untuk Memperbarui Data Pelanggan

Ilustrasi implementasi tampilan *edit* data pelanggan dapat dilihat pada Gambar 3.9.

Edit Data Pelanggan								
GELAR		NAMA PEL	ANGGAN *					
Bapak	~	oyen						
ALAMAT EMAIL *								
hi@gmail.com								

Gambar 3.6. Tampilan Halaman Form Edit Data Pelanggan

Setelah admin diberikan akses untuk menambahkan, melihat, dan mengubah data pelanggan, sistem juga menyediakan fitur untuk menghapus data yang tidak lagi diperlukan. Fitur penghapusan ini dirancang menggunakan metode *soft delete*, yang berarti data tidak dihapus secara permanen dari basis data. Sebaliknya, data akan tetap disimpan dan hanya ditandai sebagai telah dihapus melalui pengisian nilai waktu pada kolom deleted\_at. Pendekatan ini memberikan fleksibilitas

bagi sistem untuk menjaga integritas data dan memungkinkan pemulihan data di kemudian hari apabila diperlukan.

Proses dimulai ketika pengguna menekan tombol *Delete* pada halaman indeks data pelanggan yang dapat dilihat pada Gambar 3.3. Sistem kemudian menampilkan jendela konfirmasi (*modal*) untuk memastikan bahwa tindakan penghapusan dilakukan secara sadar. Jika pengguna menyetujui konfirmasi tersebut, maka sistem akan mengeksekusi fungsi delete() pada file CustomerController. Potongan kode berikut menunjukkan implementasi fungsi penghapusan data:

```
public function destroy(Customer $customer)
{
    $customer->delete();
    return redirect()->route('admin.customers.index')->with('
    success', 'Data pelanggan berhasil dihapus.');
}
```

Kode 3.4: Fungsi Delete untuk Menghapus Data Pelanggan

Ilustrasi implementasi tampilan ketika pengguna menekan tombol hapus dan konfirmasi penghapusan ditampilkan dapat dilihat pada Gambar 3.7.



Gambar 3.7. Tampilan Konfirmasi Penghapusan Data Pelanggan

## E Pengujian

NT	<b>GI</b> :	T ( D'I 'I	<b>x 1</b> 1	TT 11
INO.	Skenario	Input yang Diberikan		Hasil yang
	Pengujian		Pengujian	Diharapkan
1	Mengakses	Login sebagai admin	Klik menu	Menampilkan daftar
	halaman data		pelanggan	pelanggan jika tersedia,
	pelanggan			jika kosong muncul
				pesan "Tidak ada data
				yang tersedia pada tabel
				ini".
2	Menambahkan	Gelar: Saudara, Nama:	Klik tombol	Data baru ditampilkan
	data pelanggan	Bambang, Email:	Simpan	di halaman data
	baru (valid)	hii@gmail.com		pelanggan dan
				tersimpan di tabel
				customers.
3	Menambahkan	Nama: Hello, Email:	Klik tombol	Data berhasil
	data pelanggan	helloooo@gmail.com	"Simpan" tanpa	ditambahkan tanpa gelar
	baru (tanpa		memilih gelar	(kolom title_id
	gelar)			bernilai NULL).
4	Menambahkan	Gelar: Saudara, Nama:	Klik tombol	Muncul pesan
	data pelanggan	Oyen, Email: hii.com	"Simpan"	peringatan bahwa
	dengan format			format email harus valid
	email tidak valid			(mengandung @).
5	Menambahkan	Gelar: Saudari, Nama:	Klik tombol	Muncul pesan
	data pelanggan	Oyen, Email:	"Simpan"	peringatan bahwa email
	dengan email	hii@gmail.com		sudah digunakan dan
	yang sudah	IVER	<b>ΓΙΤΑ</b>	tidak boleh duplikat.
	digunakan			•
6	Menampilkan	Tekan tombol "Detail"	Sistem	Informasi lengkap
	detail pelanggan	pada salah satu entri	mengambil data	pelanggan ditampilkan
	NU	pelanggan	berdasarkan ID	dengan benar.
			pelanggan	
		Lanjutan di halam	an berikutnya	

Tabel 3.3. Skenario Pengujian Fitur Manajemen Data Pelanggan

No.	Skenario	Input yang Diberikan	Langkah	Hasil yang
	Pengujian		Pengujian	Diharapkan
7	Mengubah data pelanggan	Nama: Bambang $\rightarrow$ Bambang Santoso,	Tekan tombol "Update" pada	Data diperbarui dan pada halaman detail
		Email: hii@gmail.com $\rightarrow$ hi@gmail.com, Gelar: Saudara $\rightarrow$ Bapak	halaman <i>edit</i>	ditampilkan tanggal pembaruan.
8	Menghapus data pelanggan ( <i>soft</i> <i>delete</i> )	Tekan tombol "Hapus"	Konfirmasi "Ya" pada modal	Data hilang dari halaman pelanggan namun masih ada di <i>database</i> (kolom deleted_at terisi).
9	Fitur pencarian pada halaman pelanggan	Kata kunci pencarian "santoso", "bapak", "hi@gmail.com", "12 Juni"	Ketik kata kunci satu per satu pada kolom pencarian	Data yang sesuai dengan kolom gelar, nama, email, atau tanggal dibuat akan ditampilkan.
10	Filter jumlah entri yang ditampilkan	Pilih "10", "25", "50", atau "100" dari <i>dropdown</i>	Pilih opsi filter entri pada halaman indeks	Jumlah baris data yang ditampilkan sesuai dengan opsi yang dipilih.
11	Navigasi halaman <i>indexing</i>	Klik tombol " <i>Next</i> " atau " <i>Previous</i> "	Lakukan navigasi antar halaman data	Sistem menampilkan data pelanggan sesuai dengan halaman yang dipilih.

#### Lanjutan dari Tabel 3.3

Berdasarkan hasil pengujian yang disajikan pada Tabel 3.3, seluruh fungsionalitas pada halaman manajemen data pelanggan telah berfungsi dengan baik dan sesuai dengan ekspektasi.

## 3.4.2 Pengelolaan Gelar/*Title* Pelanggan

#### A Analisis Kebutuhan

Dalam sistem manajemen data pelanggan, diperlukan fitur untuk mengelola gelar atau *title* guna mendukung personalisasi dalam penulisan isi email. Gelar seperti "Mr",

ΓΑ

"Mrs", "Bapak", "Ibu", "Saudara", dan "Saudari" dikelola secara terpisah agar dapat digunakan secara konsisten dan sesuai dalam menyapa pelanggan. Halaman pengelolaan gelar ini memungkinkan admin untuk menambahkan, memperbarui, atau menonaktifkan gelar sesuai kebutuhan komunikasi.

Fitur ini penting karena sistem *email blast* yang dikembangkan ditujukan untuk pengiriman email massal dengan format yang tetap profesional dan personal. Dengan adanya pengelolaan gelar, sistem dapat menyusun salam pembuka yang sesuai dengan identitas penerima, sehingga menciptakan kesan formal sekaligus membangun kedekatan dalam komunikasi antara perusahaan dan pelanggan.

#### **B** Perancangan Basis Data

Nama Kolom	Tipe Data	NULL	Deskripsi
id	bigint(20) unsigned	Tidak	Sebagai primary key dengan nilai
			unik dan <i>auto increment</i> untuk
			mengidentifikasi setiap gelar.
name	varchar(50)	Tidak	Menyimpan singkatan gelar
			pelanggan (contoh: Mr, Mrs,
			Bapak, Ibu) dan bersifat unik.
description	varchar(100)	Ya	Penjelasan lebih lengkap dari
			singkatan gelar yang disimpan di
			kolom name.
is_active	tinyint(1)	Tidak	Menyatakan status gelar, 1 untuk
			aktif dan 0 untuk tidak aktif.
created_at	timestamp	Ya	Menyimpan informasi waktu saat
			data dibuat pertama kali.
updated_at	timestamp	Ya	Mencatat waktu terakhir kali data
	NIVE	RS	diperbarui.

Tabel 3.4. Struktur Tabel titles pada Sistem Email Blast

Tabel *titles* yang dapat dilihat pada Tabel 3.4 disusun sebagai referensi penyimpanan data terkait gelar atau sapaan pelanggan yang nantinya akan digunakan dalam sistem, khususnya dalam proses penyusunan isi email. Struktur tabel ini terdiri atas beberapa atribut penting. Kolom id berfungsi sebagai *primary key* dengan sifat unik dan *auto increment*, sehingga setiap entri memiliki identifikasi yang berbeda secara otomatis. Selanjutnya, kolom name digunakan untuk menyimpan singkatan gelar seperti "Mr", "Mrs", "Saudara", "Saudari", "Bapak", atau "Ibu". Nilai pada kolom ini harus bersifat unik dan tidak

boleh bernilai kosong (NOT NULL), karena menjadi identifikasi utama dalam penggunaan sapaan. Kolom description disediakan secara opsional untuk memberikan deskripsi atau penjelasan tambahan mengenai singkatan gelar yang tercantum pada kolom name.

Untuk mengatur status penggunaan gelar dalam sistem, disediakan kolom is\_active, di mana nilai 1 menunjukkan status aktif dan nilai 0 menunjukkan tidak aktif. Hal ini memberikan keleluasaan dalam mengelola daftar gelar yang dapat ditampilkan kepada pengguna. Terakhir, kolom created\_at dan updated\_at digunakan untuk mencatat waktu pembuatan dan waktu terakhir pembaruan data, masing-masing. Dengan desain struktur seperti ini, data gelar pelanggan dapat dikelola secara efisien, fleksibel, serta mendukung dinamika penggunaan di dalam sistem aplikasi.

#### C Flowchart Pengelolaan Gelar/Titles

Flowchart pada gambar 3.8 menggambarkan alur kerja dalam pengelolaan data gelar atau *title* pelanggan yang dilakukan secara manual melalui terminal. Seluruh proses dilakukan dengan mengeksekusi perintah SQL langsung tanpa melibatkan antarmuka pengguna berbasis web. Prosedur dimulai dari tahap *start*, dilanjutkan dengan autentikasi ke dalam sistem basis data bernama email\_blast. Setelah koneksi berhasil, pengguna memperoleh akses ke tabel titles, yaitu tabel yang menyimpan data gelar seperti "Mr", "Mrs", "Saudara", "Saudari", "Bapak", dan "Ibu".

Pengguna kemudian dapat memilih tindakan yang ingin dilakukan terhadap data tersebut, seperti menambah entri baru (*create*), mengubah data (*edit*), mengaktifkan kembali gelar (*activate*), atau menonaktifkan gelar (*deactivate*). Seluruh tindakan ini dilakukan dengan perintah SQL sesuai kebutuhan dan disertai validasi input untuk menjaga konsistensi data. Setelah eksekusi perintah selesai, hasilnya dapat diverifikasi melalui perintah SELECT untuk memastikan bahwa perubahan telah tercatat dengan benar.

# UNIVERSITAS MULTIMEDIA NUSANTARA



Gambar 3.8. Flowchart Pengelolaan Gelar/Titles

#### 1. Menambah Data (Create)

Untuk menambahkan entri gelar baru, pengguna menggunakan perintah berikut:

INSERT INTO titles (name, description, is\_active, created\_at, updated\_at) VALUES ('...', '...', 1, NOW(), NOW());

Validasi perlu dilakukan untuk memastikan bahwa nilai name tidak kosong dan tidak

22 Pengembangan Modul Manajemen..., Angelina, Universitas Multimedia Nusantara duplikat. Apabila validasi gagal, maka input harus diperbaiki dan proses diulang.

2. Mengubah Data (Edit)

Untuk melakukan perubahan terhadap data yang sudah ada, digunakan perintah seperti:

UPDATE titles
SET name = '...', description = '...', updated\_at = NOW()
WHERE id = ...;

Input harus divalidasi agar sesuai format dan tidak menyebabkan konflik data. Jika validasi gagal, proses perlu diulang.

3. Mengaktifkan Gelar (Activate)

Untuk mengaktifkan kembali suatu gelar yang sebelumnya dinonaktifkan, pengguna menjalankan perintah:

```
UPDATE titles
SET is_active = 1, updated_at = NOW()
WHERE id = ...;
```

#### 4. Menonaktifkan Gelar (Deactivate)

Sementara itu, untuk menonaktifkan gelar tertentu, perintah yang digunakan adalah:

UPDATE titles
SET is\_active = 0, updated\_at = NOW()
WHERE id = ...;

Setelah setiap aksi dijalankan, hasil perubahan dapat diverifikasi melalui pesan keluaran di terminal atau dengan menjalankan perintah:

SELECT \* FROM titles; LTIMEDIA

Jika hasil query menunjukkan perubahan sesuai dengan input, maka proses dianggap berhasil dan dilanjutkan ke tahap akhir (*end*).

#### **D** Implementasi

Fitur pengelolaan data gelar pelanggan diimplementasikan secara langsung melalui terminal dengan menggunakan perintah SQL pada basis data email\_blast. Seluruh proses dilakukan melalui baris perintah atau *command line interface*. Informasi gelar pelanggan yang tersimpan memiliki peran penting dalam penyusunan konten pada sistem pengiriman *email blast*, sehingga akurasi dan kelengkapan data menjadi aspek yang sangat diperhatikan dalam pengelolaannya. Langkah awal dalam proses pengelolaan data gelar pelanggan adalah menampilkan seluruh isi tabel titles untuk memperoleh gambaran data yang tersimpan sebelum dilakukan perubahan. Hal ini dilakukan dengan menjalankan perintah:

```
SELECT * FROM titles;
```

#### Kode 3.5: Perintah SQL Untuk Menampilkan Tabel

Perintah tersebut akan menampilkan seluruh entri gelar yang telah tersimpan dalam sistem, termasuk informasi seperti nama gelar, deskripsi, status keaktifan, serta waktu pembuatan dan pembaruan data. Hasil dari query ini ditunjukkan pada Gambar 3.9.

+   id		description	is_active	created_at	+ updated_at
1	Mr   Mrs	Mister (Tuan) Missus (Nyonya)	1	2025-06-10 11:37:00 2025-06-10 11:37:00	2025-06-10 11:37:00     2025-06-10 11:37:00
	Bapak	Bapak	1	2025-06-10 11:37:00	2025-06-10 11:37:00
	Saudara Saudara	Saudara	1	2025-06-10 11:37:00	2025-06-10 11:37:00
+		).001 sec)	+		-++

Gambar 3.9. Tampilan Data Awal Tabel titles

Melalui tampilan tersebut, pengguna dapat memastikan kondisi data awal sebelum melakukan aksi seperti penambahan, pengubahan, atau pengaturan status aktif/nonaktif gelar. Tampilan ini juga menjadi acuan penting untuk menghindari duplikasi data serta memverifikasi integritas entri yang ada.

Setelah memastikan kondisi awal data yang tersimpan dalam tabel titles, langkah berikutnya adalah melakukan pengelolaan data sesuai kebutuhan sistem. Salah satu proses yang umum dilakukan adalah penambahan data gelar baru ke dalam tabel, yang akan digunakan sebagai referensi dalam penyusunan konten email. Penambahan entri gelar baru ke dalam tabel titles dilakukan dengan mengeksekusi perintah SQL berikut:

I INSERT INTO titles (name, description, is\_active, created\_at, updated\_at)

#### 2 VALUES ('Ny', 'Nyonya', 1, NOW(), NOW());

Kode 3.6: Perintah SQL Untuk Menambahkan Data Gelar Baru ke Tabel

Perintah ini akan menyimpan data gelar "Ny" (Nyonya) sebagai entri aktif dalam sistem. Fungsi NOW() digunakan untuk mencatat waktu saat data dibuat dan diperbarui. Setelah perintah dijalankan, pengguna dapat memverifikasi keberhasilan proses dengan menampilkan ulang isi tabel menggunakan query SELECT \* FROM titles; untuk memastikan bahwa entri baru telah muncul.

+   +	id		description	is_active	created_at	++   updated_at
	1 2 3 4 5 6 9	Mr     Mrs     Bapak     Ibu     Saudara     Saudari     Ny	Mister (Tuan) Missus (Nyonya) Bapak Ibu Saudara Saudari Nyonya		2025-06-10 11:37:00 2025-06-10 11:37:00 2025-06-10 11:37:00 2025-06-10 11:37:00 2025-06-10 11:37:00 2025-06-10 11:37:00 2025-06-10 11:37:00 2025-06-14 08:18:11	2025-06-10       11:37:00         2025-06-10       11:37:00         2025-06-10       11:37:00         2025-06-10       11:37:00         2025-06-10       11:37:00         2025-06-10       11:37:00         2025-06-10       11:37:00         2025-06-10       11:37:00         2025-06-10       11:37:00         2025-06-10       11:37:00         2025-06-10       11:37:00
+ 7	rows	++ s in set (0	).000 sec)	+		++

Gambar 3.10. Hasil Eksekusi Perintah Penambahan Data Gelar pada Terminal

Gambar 3.10 memperlihatkan hasil eksekusi langsung di terminal saat menambahkan gelar baru, dalam hal ini gelar "Ny" (Nyonya). Proses ini memastikan bahwa data baru tersimpan dalam sistem secara aktif dan dapat langsung digunakan dalam modul pengiriman email. Validasi terhadap keunikan name dan kelengkapan data menjadi penting untuk menjaga konsistensi serta mencegah duplikasi entri dalam basis data.

Edit Data Pelanggan		
GELAR		NAMA PELANGGAN *
Pilih Gelar (Opsional)	~	hello
Pilih Gelar (Opsional)		
Mr		
Mrs		
Bapak		🛱 Update Data Pelanggan
Ibu		
Saudara		
Saudari		
Ny		

Gambar 3.11. Tampilan Dropdown Website Setelah Penambahan Gelar "Ny" (Nyonya)

Gambar 3.11 menunjukkan bahwa setelah gelar "Ny" berhasil ditambahkan melalui perintah SQL, sistem secara otomatis memperbarui daftar *dropdown* gelar pelanggan pada antarmuka web.

Setelah melakukan penambahan data, pengelolaan selanjutnya yang mungkin dibutuhkan adalah melakukan pembaruan atau koreksi terhadap gelar yang telah tersimpan

dalam tabel. Proses ini berguna apabila terdapat kesalahan penulisan, perubahan standar penamaan, atau kebutuhan penyesuaian lainnya dalam sistem. Untuk mengubah data gelar berdasarkan ID tertentu, digunakan perintah SQL berikut:

```
1 UPDATE titles
2 SET name = 'Dr', description = 'Doktor', updated_at = NOW()
3 WHERE id = 9;
```



Query tersebut akan memperbarui kolom name dan description pada baris yang memiliki id = 9, serta mencatat waktu pembaruan menggunakan fungsi NOW(). Proses pengubahan ini harus dilakukan secara hati-hati agar tidak terjadi konflik ID, kesalahan data, atau duplikasi nama gelar yang sudah ada dalam tabel.

++   id	name	description	is_active	created_at	++   updated_at				
	Mr Mrs	Mister (Tuan)	1	2025-06-10 11:37:00	2025-06-10 11:37:00				
	Bapak	Bapak	1	2025-06-10 11:37:00	2025-06-10 11:37:00				
4	Ibu Saudara	Ibu Saudara	1	2025-06-10 11:37:00 2025-06-10 11:37:00	2025-06-10 11:37:00				
6     9	Saudari Dr	Saudari Doktor	1 1	2025-06-10 11:37:00 2025-06-14 08:18:11	2025-06-10 11:37:00   2025-06-14 09:40:18				
++ 7 rows	7 rows in set (0.001 sec)								

Gambar 3.12. Hasil Eksekusi Perintah Pengubahan Data Gelar pada Terminal

Gambar 3.12 menampilkan hasil pengubahan gelar pada terminal, di mana gelar "Ny" (Nyonya) diperbarui menjadi "Dr" dengan deskripsi "Doktor".

ELAR		NAMA PELANGGAN *
Pilih Gelar (Opsional)	~	hello
Pilih Gelar (Opsional)		
Mr		
Mrs		
Bapak		🛱 Update Data Pelanggan
Ibu		
Saudara		
Saudari		
Dr		

Gambar 3.13. Tampilan *Dropdown* Website Setelah Pengubahan Gelar Menjadi "Dr" (Doktor)

Setelah dilakukan perubahan data gelar dari "Ny" menjadi "Dr" melalui terminal, seperti diperlihatkan pada Gambar 3.13, *dropdown* pada halaman web langsung menampilkan pembaruan tersebut.

Selain menambahkan dan memperbarui data, pengelolaan gelar pelanggan juga mencakup pengaturan status keaktifan. Apabila suatu gelar tidak lagi diperlukan dalam proses pengiriman email, maka dapat dinonaktifkan menggunakan perintah berikut:

```
1 UPDATE titles
2 SET is_active = 0, updated_at = NOW()
3 WHERE id = 9;
```

Kode 3.8: Perintah SQL Untuk Menonaktifkan Data Gelar Dalam Tabel titles

Perintah ini akan mengubah nilai is\_active menjadi 0, menandakan bahwa gelar tersebut tidak lagi aktif. Data gelar tetap tersimpan dalam tabel titles, namun tidak akan digunakan dalam proses *email blast* karena statusnya nonaktif.

+   id	+   name	+	+   is_active	+   created_at	++   updated_at
+   1   2   3   4   5   6   9	+   Mr   Mrs   Bapak   Ibu   Saudar   Saudar   Dr	Mister (Tuan)   Missus (Nyonya)   Bapak   Ibu na   Saudara i   Saudari   Doktor	1   1   1   1   1   1   0	2025-06-10 11:37:00 2025-06-10 11:37:00 2025-06-10 11:37:00 2025-06-10 11:37:00 2025-06-10 11:37:00 2025-06-10 11:37:00 2025-06-14 08:18:11	+ 2025-06-10 11:37:00   2025-06-10 11:37:00   2025-06-10 11:37:00   2025-06-10 11:37:00   2025-06-10 11:37:00   2025-06-10 11:37:00   2025-06-14 09:53:59
7 row	'+ s in set	(0.001 sec)	+		++

Gambar 3.14. Eksekusi Perintah Menonaktifkan Data Gelar pada Terminal

Gambar 3.14 memperlihatkan proses menonaktifkan gelar "Dr" melalui terminal. Pengaturan ini memungkinkan sistem tetap menyimpan riwayat data tanpa menghapusnya secara permanen, memberikan fleksibilitas jika sewaktu-waktu gelar tersebut dibutuhkan kembali.

Edit Data Pelanggan													
GELAR	NAM	A PELA	NGGAN	•									
Pilih Gelar (Opsional) ~	h	ello											
Pilih Gelar (Opsional)													
Mr													
Mrs													
Bapak			BU	Upda	ite Dat	a Pelo	anggo	an					
Ibu													_
Saudara													
Saudari													

Gambar 3.15. Tampilan Dropdown Website Setelah Menonaktifkan Gelar "Dr"

Pada Gambar 3.15, gelar "Dr" yang sebelumnya ditampilkan di dalam daftar, kini tidak lagi muncul setelah statusnya diubah menjadi nonaktif.

Sementara itu, jika sebuah gelar yang sebelumnya dinonaktifkan ingin digunakan kembali, maka statusnya dapat diubah menjadi aktif dengan perintah berikut:

```
1 UPDATE titles
2 SET is_active = 1, updated_at = NOW()
3 WHERE id = 9;
```

Kode 3.9: Perintah SQL Untuk Mengaktifkan Kembali Data Gelar Dalam Tabel titles

Perintah ini akan memperbarui nilai is\_active menjadi 1, menandakan bahwa gelar tersebut telah diaktifkan dan dapat digunakan kembali oleh sistem.

+   i	.d	name	   description	is_active	created_at	updated_at
+             +	1 2 3 4 5 6 9	Mr Mrs Bapak Ibu Saudara Saudari Dr	Mister (Tuan)   Missus (Nyonya)   Bapak   Ibu   Saudara   Saudari   Doktor	1     1     1     1     1     1	2025-06-10 11:37:00 2025-06-10 11:37:00 2025-06-10 11:37:00 2025-06-10 11:37:00 2025-06-10 11:37:00 2025-06-10 11:37:00 2025-06-10 11:37:00 2025-06-14 08:18:11	2025-06-10 11:37:00   2025-06-10 11:37:00   2025-06-10 11:37:00   2025-06-10 11:37:00   2025-06-10 11:37:00   2025-06-10 11:37:00   2025-06-10 11:37:00   2025-06-14 10:02:56
7 r	ows	s in set (6	0.001 sec)			

Gambar 3.16. Hasil Eksekusi Perintah Mengaktifkan Kembali Data Gelar pada Terminal

Gambar 3.16 menunjukkan hasil eksekusi perintah aktivasi gelar melalui terminal. Dengan mekanisme ini, administrator sistem memiliki kontrol penuh atas daftar gelar yang tampil secara dinamis, tanpa harus kehilangan data historis yang pernah digunakan sebelumnya.

dit Data Pelanggan		
ELAR		NAMA PELANGGAN *
Pilih Gelar (Opsional)	~	hello
Pilih Gelar (Opsional)		
Mr		
Mrs		
Bapak		🛱 Update Data Pelanggan
Ibu		
Saudara		
Saudari		
Dr		

Gambar 3.17. Tampilan Dropdown Website Setelah Mengaktifkan Kembali Gelar "Dr"

Gambar 3.17 memperlihatkan bahwa setelah status gelar "Dr" diubah kembali menjadi aktif, opsi tersebut kembali muncul dalam daftar *dropdown* di antarmuka pengguna.

## E Pengujian

No.	Skenario	Input yang Diberikan	Langkah	Hasil yang
	Pengujian		Pengujian	Diharapkan
1	Menampilkan	Query SQL	Jalankan perintah	Data gelar pelanggan
	seluruh data	menampilkan data	SELECT * FROM	ditampilkan sesuai isi
	gelar		titles; di	tabel titles
	5		terminal	
2	Menambahkan	name = "Ny",	Jalankan perintah	Data gelar baru berhasil
	gelar baru	description =	INSERT INTO	tersimpan dan muncul
		"Nyonya"	titles ()	dalam hasil query
			dengan data baru	
3	Menambahkan	name = "Ny",	Jalankan perintah	Gagal menyimpan data,
	gelar dengan	description =	INSERT dengan	muncul pesan kesalahan
	nama duplikat	"Nyonya"	nilai name yang	karena data duplikat
			sudah ada	
4	Mengedit	id = 9, description	Jalankan UPDATE	Deskripsi pada ID
	deskripsi gelar	= "Tuan"	titles SET	tersebut diperbarui
			description =	sesuai input
			WHERE id	
			=	
5	Menonaktifkan	id = 9	Jalankan UPDATE	Status is_active
	gelar		titles SET	berubah menjadi 0
			is_active = 0	(nonaktif)
			WHERE id =	
6	Mengaktifkan	id = 9	Jalankan UPDATE	Status is_active
	gelar kembali	IVER	titles SET	berubah menjadi 1 (aktif
			is_active = 1	kembali)
	MU		WHERE $id = \dots$	A
7	Mengedit nama	name = ""	Jalankan UPDATE	Gagal atau muncul
	gelar menjadi	JAN	titles SET	peringatan input
	kosong		name = ''	
			WHERE id =	

Tabel 3.5. Skenario Pengujian Fitur Pengelolaan Data Gelar Pelanggan

Berdasarkan hasil pengujian yang tertera pada Tabel 3.5 yang telah dilakukan terhadap fitur pengelolaan data gelar pada sistem, dapat disimpulkan bahwa seluruh skenario yang diuji berjalan sesuai ekspektasi, mulai dari penambahan entri baru, pembaruan informasi gelar, perubahan status keaktifan, hingga validasi terhadap input duplikat dan kosong. Setiap instruksi SQL yang dijalankan menunjukkan hasil yang akurat dan sesuai dengan kondisi yang diharapkan, baik dalam hal penyimpanan data pada tabel titles maupun dalam proses verifikasi melalui tampilan data. Oleh karena itu, fitur ini dapat dinyatakan telah berfungsi dengan baik dan mampu mendukung kebutuhan operasional sistem secara efektif.

#### 3.4.3 Integrasi Import Data Pelanggan melalui Excel

#### A Analisis Kebutuhan

Fitur import data pelanggan melalui file Excel telah tersedia dalam sistem yang dikembangkan. Namun, untuk memastikan data hasil unggahan dapat langsung dimanfaatkan dalam pengelolaan pelanggan, dibutuhkan proses integrasi lebih lanjut. Integrasi ini mencakup penyimpanan otomatis data ke dalam tabel customers di basis data, data yang telah tersimpan perlu ditampilkan secara otomatis pada halaman indeks pelanggan. Dengan demikian, admin dapat segera mengakses dan mengelola informasi yang baru ditambahkan. Implementasi ini menjadi krusial demi menjamin konsistensi data, meningkatkan efisiensi operasional, serta mempercepat proses kerja administratif yang berlangsung dalam sistem.

#### **B** Perancangan Basis Data

Data yang diunggah melalui proses *import* akan langsung disimpan ke dalam tabel customers, yang telah dirancang sebelumnya untuk memuat atribut-atribut penting seperti name, email, dan title\_id. Relasi antar tabel telah diatur melalui penggunaan *foreign key*, di mana title\_id pada tabel customers mengacu pada kolom id pada tabel titles. Mekanisme ini memungkinkan sistem untuk mengelola dan menampilkan gelar pelanggan secara akurat berdasarkan data relasional yang tersedia. Karena struktur tabel sudah sesuai dengan kebutuhan fitur, tidak diperlukan perancangan ulang pada skema basis data. Struktur lengkap tabel customers dapat dilihat pada Tabel 3.2.



#### C Flowchart Integrasi Import Data Pelanggan melalui Excel

Gambar 3.18. Flowchart Integrasi Import Data Pelanggan melalui Excel

Fitur integrasi *import* data Excel dalam sistem dikembangkan untuk mempermudah proses administrasi data pelanggan serta mendukung alur kerja pengiriman *email blast*.

Setelah pengguna berhasil melakukan *login*, pengguna dapat mengakses halaman *Blast Email*. Pada halaman ini, pengguna dapat mengisi data email yang akan dikirim, kemudian melanjutkan dengan mengunggah data pelanggan dalam format file Excel.

file Excel yang diunggah memuat informasi pelanggan seperti nama, alamat email, dan gelar (opsional). Data tersebut langsung diproses dan disimpan ke dalam tabel customers. Setelah penyimpanan berhasil, sistem akan menampilkan data pelanggan tersebut secara otomatis pada halaman indeks data pelanggan. Sebelum proses pengiriman email berlangsung, sistem akan melakukan validasi terhadap alamat email pada setiap entri. Jika ditemukan data dengan format email yang tidak sesuai, maka pengiriman akan dibatalkan untuk entri tersebut. Namun, apabila alamat email dinyatakan valid, maka pesan akan dikirimkan secara otomatis. Proses ini dirancang untuk memastikan bahwa hanya data pelanggan yang benar-benar valid yang digunakan dalam pengiriman email.

#### D Implementasi

Untuk mempermudah proses pengelolaan data pelanggan dalam pengiriman *email blast*, sistem menyediakan fitur *import* data melalui file Excel. Fitur ini memungkinkan admin untuk mengunggah data pelanggan sekaligus, yang kemudian disimpan langsung ke dalam basis data dan ditampilkan secara otomatis pada halaman data pelanggan.

IMPORT EMAILS (EXCEL): •									
Choose File	a File No file chosen								
Format yang didukung: Excel (xlsx, xls) atau CSV Format kolom: Email   Nama   Gelar (opsional)									

Gambar 3.19. Fitur Import Excel pada Halaman Blast Email

Gambar 3.28 menunjukkan antarmuka pengguna tempat file Excel diunggah. Administrator dapat menekan tombol *Choose File* untuk memilih dokumen yang berisi data pelanggan yang ingin dikirimkan email secara massal.

	А	В	С
1	angelina5@student.umn.ac.id	Intern	saudari
2	<u>halo.com</u>	Halo	Kakak

Gambar 3.20. Contoh Isi file Excel yang Diunggah

Gambar 3.20 menampilkan struktur data dalam file Excel yang diunggah. Setiap baris merepresentasikan satu entri pelanggan dengan informasi berupa alamat email, nama, dan gelar.

Data F	Pelanggan				
Create				Coursely.	
Snow	10 v entries			Search:	
# ‡	GELAR 🍦 NAMA	🔶 EMAIL	CREATED DATE	÷ 0P	PSI ≑
1	Saudari Intern	angelina5@student.umn.ac.id	14 June 2025	Detail	dit Delete
2	- Halo	halo.com	14 June 2025	Detail	dit Delete

Gambar 3.21. Tampilan Hasil Import Data Melalui Excel

Gambar 3.21 memperlihatkan hasil akhir dari proses *import*, di mana data pelanggan yang berasal dari file Excel berhasil ditampilkan pada halaman daftar pelanggan. Setelah proses ini selesai, data telah tersimpan secara otomatis ke dalam tabel customers dan siap digunakan dalam proses pengiriman *email blast*. Untuk memastikan bahwa data benar-benar tersimpan dengan baik, Gambar 3.22 berikut menampilkan tampilan isi basis data yang telah diperbarui setelah proses *import* selesai. Hal ini menunjukkan bahwa data dari Excel telah terhubung dengan baik ke struktur *database* yang ada.

++		+	+		+
id   title_id	name	email	deleted_at	created_at	updated_at
9   6     10   NULL	Intern Halo	angelina5@student.umn.ac.id   halo.com	NULL NULL	2025-06-14 23:13:04 2025-06-14 23:13:04	2025-06-14 23:13:04   2025-06-14 23:13:04
2 rows in set (0.	.007 sec)				

Gambar 3.22. Tampilan Tabel customers Setelah Proses Import dari file Excel

Pada sisi *backend*, proses integrasi dilakukan melalui class EmailsImport. file Excel dibaca secara baris demi baris, lalu setiap data dikaitkan langsung dengan model Customer. Potongan kode berikut merupakan bagian utama yang bertugas menyimpan data hasil *import* ke dalam tabel customers.

```
1 $customer = Customer::firstOrCreate(
2 ['email' => $email],
3 [
4 'name' => $name,
5 'title_id' => $titleId,
6 ]
7 );
8
9 if (! $customer->title_id && $titleId) {
9 $customer->update(['title_id' => $titleId]);
11 }
```



Potongan Kode 3.10 menunjukkan bagaimana sistem memeriksa apakah email pelanggan sudah ada sebelumnya. Jika belum, maka data baru akan disimpan. Apabila title\_id belum tercatat tetapi tersedia pada file, maka nilai tersebut akan diperbarui secara otomatis. Proses ini menghubungkan data yang diunggah dari Excel dengan tabel customers, yang strukturnya dapat dilihat pada Tabel 3.2.

## E Pengujian

No.	Skenario	Input yang	Langkah	Hasil yang
	Pengujian	Diberikan	Pengujian	Diharapkan
1	Import data	file Excel berisi	Klik tombol	Data tersimpan
	pelanggan dari file	kolom nama, email,	"Import", pilih file	ke database
	Excel valid	dan gelar	Excel valid, lalu	
			klik tombol	
			"Kirim"	
2	Menampilkan data	File Excel dengan 5	Lakukan import,	Semua data dari
	pelanggan hasil	data pelanggan	lalu navigasi ke	file Excel tampil
	<i>import</i> di halaman		halaman data	pada halaman
	pelanggan		pelanggan	data pelanggan
3	Memperbarui gelar	file Excel dengan	Import file Excel	Data pelanggan
	pelanggan jika	email pelanggan yang	yang memuat	diperbarui pada
	email sudah ada,	sudah ada, namun ada	email lama namun	kolom gelar
	tetapi gelar belum	tambahan kolom	kini dengan kolom	sesuai isi file
	diisi	gelar	gelar terisi	Excel
4	Mengabaikan baris	file Excel berisi data	Import file seperti	Baris kosong
	kosong dalam file	pelanggan dan	biasa	diabaikan dan
	Excel	beberapa baris	<b>Δ Τ Ι </b>	tidak
		kosong di tengah atau		menyebabkan
		bawah	<b>FD</b>	error
		Lanjutan di halaman	berikutnya	
	NU	SAN	TAR	Α

Tabel 3.6. Skenario Pengujian Fitur Import Data Pelanggan Melalui Excel

#### Lanjutan dari Tabel 3.6

No.	Skenario	Input yang	Langkah	Hasil yang
	Pengujian	Diberikan	Pengujian	Diharapkan
5	Menolak file Excel	file yang bukan Excel	Klik tombol	Sistem
	yang tidak sesuai	(misal .txt)	"Import", pilih file	menampilkan
	format atau rusak		tidak valid, lalu	pesan error dan
			kirim	tidak
				melanjutkan
				proses
6	Menangani data	file Excel berisi dua	Import file ke	Hanya satu data
	ganda (email	baris dengan email	sistem	yang disimpan
	duplikat) dalam	yang sama		
	file Excel			
7	Import file Excel	file Excel hanya	Lakukan <i>import</i>	Data tetap
	tanpa kolom gelar	berisi kolom nama	file	tersimpan, kolom
		dan email, tanpa gelar		gelar bernilai
				NULL atau
				kosong
8	Import file Excel	file Excel yang sama	Unggah kembali	Tidak terjadi
	yang sebelumnya	pernah diunggah	file yang sama	duplikasi, data
	pernah diunggah	sebelumnya		diperbarui atau
				dilewati sesuai
				logika sistem

Berdasarkan hasil pengujian pada Tabel 3.6 yang dilakukan terhadap fitur integrasi *import* data pelanggan melalui file Excel, seluruh skenario pengujian telah menunjukkan hasil sesuai dengan ekspektasi. Setiap tahapan pengujian, mulai dari proses unggah file, penyimpanan data ke dalam basis data customers, hingga penampilan data secara otomatis pada halaman daftar pelanggan, berjalan lancar tanpa kendala.

#### 3.4.4 Integrasi Fitur dengan Sistem Email Blast

#### A Analisis Kebutuhan

Dalam proses integrasi sistem *Email Blast*, salah satu kebutuhan utama yang diidentifikasi adalah penyatuan data gelar (*title*) dengan atribut pelanggan lainnya, yaitu nama dan alamat email. Tujuan dari integrasi ini adalah agar data pelanggan yang telah

Pengembangan Modul Manajemen..., Angelina, Universitas Multimedia Nusantara

tersimpan dapat dimanfaatkan secara optimal untuk keperluan pengiriman email massal yang bersifat personal. Dengan struktur data yang terintegrasi, sistem mampu menampilkan informasi pelanggan secara lengkap di halaman daftar pelanggan serta memungkinkan pemanfaatan data tersebut dalam modul pengiriman email.

Kebutuhan berikutnya berkaitan dengan tampilan antarmuka pada halaman *Blast Email*, di mana pengguna perlu diberikan dua metode dalam memilih penerima email yaitu mengunggah dengan file Excel yang berisi daftar pelanggan dan dengan memilih langsung dari data pelanggan yang sudah tersimpan di sistem. Penyediaan dua opsi ini bertujuan untuk memberikan fleksibilitas dalam operasional pengiriman email, sesuai dengan kebiasaan kerja dan preferensi pengguna.

Selain pemilihan data penerima, sistem juga dirancang untuk mendukung kebutuhan personalisasi isi email. Untuk itu, dibutuhkan kemampuan dalam melakukan pemanggilan data pelanggan, khususnya gelar dan nama ke dalam badan pesan email. Hal ini memungkinkan sistem menyusun sapaan personal seperti "Bapak Andi" atau "Ibu Dian" secara otomatis, sehingga meningkatkan kesan eksklusif bagi penerima email. Agar proses ini berjalan konsisten dan akurat, integrasi data dilakukan melalui relasi antartabel, yaitu titles, customers, dan emails, yang menjadi dasar dalam menyusun informasi yang digunakan di berbagai bagian sistem.

#### **B** Perancangan Basis Data

Integrasi sistem *Email Blast* didukung oleh keterkaitan antara tiga tabel utama dalam basis data, yaitu titles, customers, dan emails. Relasi antar tabel ini dirancang untuk memastikan konsistensi informasi pelanggan serta mendukung proses personalisasi dan pengiriman email dalam skala besar.

Tabel titles yang tertera pada 3.4 menyimpan informasi mengenai gelar atau sapaan formal seperti Mr", "Mrs", "Saudara", "Saudari", "Bapak", dan "Ibu". Meskipun bersifat opsional, data dalam tabel ini sangat berguna untuk membentuk sapaan personal pada isi email. Tabel ini memiliki hubungan satu ke banyak (*one-to-many*) dengan tabel customers melalui kolom title\_id, yang memungkinkan satu gelar digunakan oleh lebih dari satu pelanggan.

Tabel customers yang tertera pada 3.2 menyimpan data inti pelanggan, seperti nama dan alamat email, serta memiliki referensi ke gelar melalui kolom title\_id. Tabel ini menjadi sumber utama data pelanggan yang digunakan dalam berbagai fitur sistem, termasuk dalam proses seleksi penerima email. Ketika pengguna memilih penerima dari daftar internal, sistem akan mengambil data langsung dari tabel ini.

Selanjutnya, tabel emails merekam data yang digunakan dalam proses pengiriman email, yang mencakup informasi seperti alamat email, nama penerima, batch\_id, dan

status pengiriman. Nilai pada kolom name dapat berupa kombinasi antara gelar dan nama pelanggan, seperti "Ibu Sari", yang diperoleh berdasarkan data dari tabel customers dan titles.

Meskipun tidak terdapat hubungan *foreign key* eksplisit antara tabel emails dan customers, keterkaitan antar data tetap dibangun secara logis melalui proses impor dan pemrosesan data. Sistem akan mencocokkan alamat email yang diunggah atau dimasukkan oleh pengguna dengan entri yang ada di tabel customers. Jika ditemukan kecocokan, maka informasi tambahan seperti gelar dan nama akan diambil dan digunakan untuk melengkapi data yang dimasukkan ke dalam tabel emails.

#### C Flowchart Integrasi Fitur dengan Sistem Email Blast

Pada Gambar 3.23 memperlihatkan alur utama yang terjadi dalam sistem Email Blast, dimulai dari proses login pengguna hingga email berhasil dikirim. Setelah berhasil masuk ke sistem, pengguna akan diarahkan ke halaman *Blast Email*. Di halaman ini, terdapat dua pilihan metode pengiriman email, yaitu secara manual atau menggunakan *template* yang telah tersedia. Apabila pengguna memilih metode manual, maka pengguna perlu mengisi sejumlah informasi secara langsung, seperti kolom CC (opsional), BCC (opsional), subjek, serta isi atau deskripsi email. Sebaliknya, jika memilih menggunakan *template*, pengguna cukup memilih salah satu *template* yang tersedia dan hanya perlu menambahkan informasi pada kolom CC dan BCC jika diperlukan.

Langkah berikutnya adalah memilih penerima email. Sistem menyediakan dua sumber data yaitu pengguna dapat mengunggah file Excel yang berisi daftar pelanggan, atau memilih langsung dari data pelanggan yang sudah tersimpan di sistem. Untuk file Excel, sistem secara otomatis akan melakukan pengecekan terhadap format file. Bila file tidak sesuai format yang diharapkan, sistem akan menampilkan pesan *error*. Namun, jika file valid, nama file tersebut akan muncul pada kolom *import* sebagai tanda bahwa file berhasil diunggah.

Tahapan selanjutnya berkaitan dengan pengaturan waktu pengiriman email. Pengguna dapat memilih untuk langsung mengirimkan email atau menjadwalkannya pada waktu tertentu. Jika opsi penjadwalan dipilih, maka pengguna harus menentukan tanggal dan jam pengiriman. Ketika proses pengiriman berlangsung, sistem akan memverifikasi apakah data penerima berasal dari file Excel atau dari *database* internal. Jika datanya berasal dari file Excel dan belum tercatat di sistem, maka data tersebut akan disimpan terlebih dahulu ke dalam tabel customers. Setelah proses pengiriman selesai, pengguna akan dialihkan ke halaman List Blast yang menampilkan daftar email yang telah dikirimkan.



Gambar 3.23. Flowchart Integrasi Fitur Pada Sistem Email Blast

ΝΤΑ

#### D Implementasi

Fitur Data Pelanggan, *Import* File dari Excel, dan Pengelolaan *Title* dikembangkan sebagai bagian dari sistem Email Blast untuk mengelola dan mengirim email massal kepada pelanggan secara efisien dan profesional. Tujuan dari pengembangan ini adalah

Α

S

menyediakan sarana komunikasi yang fleksibel, personal, dan terintegrasi dengan data pelanggan yang sudah ada. Implementasi fitur ini menggabungkan aspek antarmuka pengguna yang ramah, fleksibilitas dalam pemilihan data penerima, serta kemampuan personalisasi isi email.

(B) Fyrlliat	Rembon Karya Digital			<b>Q</b>
DASHBOARD	Dashboard / Email Blast			
Dashboard     Data Penaguna	Email Blast			
🖾 Blast Email	Placeholder yang tersedia untuk bac (nane) - Nama penerima	dan email:		
🖽 List Blast	(email) - Email penerima (title) - Gelar penerima (full_ame) - Nama lengkap dengan g Contact: "the little largeme "schap medical" 2	<b>jelar</b> Ch. Ranak, John Doo <sup>*</sup>		
💄 Data Pelanggan	TYPE •	cc	BCC	
	schedule		PILIH PENERIMA *	
	Jadwalkan pengiriman		Pilih Cara Penerima	~
			🖈 Kirim	
			Copyri	aht © 2025. Rembon Karva Diait

Gambar 3.24. Tampilan Halaman Blast Email Lengkap

Gambar 3.24 menunjukkan tampilan utama halaman *Blast Email* yang menjadi pusat aktivitas pengguna dalam menyusun dan mengirim email massal. Pada halaman ini, pengguna dapat menentukan berbagai parameter pengiriman email, seperti CC, BCC, subjek email, dan isi atau deskripsi email. Sistem menyediakan dua metode untuk memilih daftar penerima email, yaitu melalui unggahan file Excel atau memilih langsung dari data pelanggan yang tersimpan di sistem. Penyediaan dua metode ini ditujukan untuk mendukung fleksibilitas operasional, sehingga pengguna dapat menyesuaikan cara kerja dengan preferensi masing-masing.

# UNIVERSITAS MULTIMEDIA NUSANTARA

Placeholder yang tersedia untu (name) - Nama penerima (emi1) - Email penerima (title) - Gelar penerima (full_nee) - Nama lengkap deng Contoh: "th. (title) [name]" akan men	t <b>badan email:</b> an gelar adi 'Yth. Bapak John Doe''			
(PE *	сс		BCC	
Manual	~	~		~
JBJECT *				
Intern PT Rembon Karya Digital				
SCRIPTION *				
↔ → Paragraph → B				
Solamat Malam (full_pame), apo	kah banar untuk (amail) danaan n	ama "Inamol" Ititloli		
selution walant frail name, abo	Kan benar antak ternany dengan n	and (numer, (mer)		

Gambar 3.25. Contoh Pengisian Isi Email dengan Placeholder

Gambar 3.25 memperlihatkan contoh pengisian isi email menggunakan placeholder seperti {name}, {title}, {full\_name}, dan {email}. Placeholder ini memungkinkan sistem untuk menyisipkan data pelanggan secara otomatis ke dalam badan email saat proses pengiriman berlangsung. Fitur ini menjadi penting dalam menciptakan komunikasi yang lebih personal. Misalnya, jika pengguna menuliskan "Halo {full\_name}," maka sistem akan menggantinya dengan "Halo Saudari Intern," apabila data pelanggan yang bersangkutan memiliki gelar "Saudari" dan nama "Intern." Fitur ini tidak hanya meningkatkan kesan eksklusif bagi penerima, tetapi juga membangun kedekatan emosional yang lebih kuat antara perusahaan dan pelanggan.



Gambar 3.26. Contoh Hasil Email Setelah Placeholder Diterapkan

Gambar 3.26 menunjukkan hasil dari email yang dikirim ke pelanggan. Sistem telah berhasil menggantikan placeholder dengan informasi sebenarnya dari *database* customers. Contoh pada Gambar 3.26 memperlihatkan email yang dimulai dengan sapaan formal seperti "Selamat Malam Saudari Intern," yang dihasilkan dari penggabungan data pada kolom title dan name melalui penggunaan *placeholder* full\_name. Selain itu, seluruh *placeholder* lain seperti name, title, dan email juga berhasil tergantikan dengan data

aktual dari basis data. Hal ini menunjukkan bahwa integrasi antar tabel titles, customers, dan emails telah berjalan dengan benar.

SCHEDULE	PILIH PENERIMA *	
Jadwalkan pengiriman	Pilih dari Data Pelanggan	~
PILIH PENERIMA DARI DATA PELANGGAN *		
Cari pelanggan	Pilih Semua	1 dari 5 dipilih
Bapak Bambang Santoso		
hi@gmail.com		
Gelar: Bapak		
Bapak Halo		
halo.com		
Gelar. Bapak		
hello		
helloooo@gmail		
Saudari Intern		
angelina5@student.umn.ac.id		
	≰l Kirim	

Gambar 3.27. Fitur Pemilihan Penerima dari Data Pelanggan

Gambar 3.27 memperlihatkan tampilan antarmuka yang digunakan untuk memilih daftar penerima dari data pelanggan yang telah tersimpan. Sistem secara otomatis menampilkan informasi pelanggan yang sudah terdaftar, termasuk gelar, nama, dan alamat email. Fitur ini dirancang untuk memudahkan proses pemilihan penerima email dari data pelanggan yang telah tersimpan di dalam sistem tanpa harus mengunggah file terlebih dahulu. Pengguna dapat memilih satu atau beberapa entri pelanggan yang telah tervalidasi, termasuk informasi nama, gelar, dan alamat email.

IMPORT EMAILS (EX	CEL): *	
Choose File	No file chosen	
Format yang didu Format kolom: En	kung: Excel (xisx, xis) atau CSV nail   Nama   Gelar (opsional)	 

Gambar 3.28. Fitur Unggah File Excel sebagai Sumber Data Penerima

Gambar 3.28 menampilkan fitur Import Excel yang memungkinkan pengguna untuk mengunggah file berisi daftar pelanggan dari sumber eksternal. Sistem akan melakukan pengecekan terhadap struktur file untuk memastikan bahwa kolom-kolom penting seperti nama, email, dan gelar tersedia dengan format yang sesuai. Apabila format file telah memenuhi ketentuan, seluruh data pelanggan akan disimpan ke dalam basis data dan secara otomatis ditampilkan di halaman daftar penerima email.

Perlu dicatat bahwa sistem tidak secara langsung menolak entri dengan alamat email yang tidak valid. Seluruh data tetap dimasukkan ke dalam database customers, termasuk baris dengan format email yang tidak sesuai. Namun, ketika proses pengiriman email dilakukan, sistem akan mengidentifikasi alamat email yang tidak valid dan menandainya dengan status pengiriman "failed". Dengan mekanisme ini, pengguna tetap dapat mengelola seluruh data pelanggan yang telah diimpor, sekaligus diberikan kesempatan untuk melakukan koreksi informasi langsung melalui halaman data pelanggan sebelum digunakan dalam proses komunikasi selanjutnya.

## E Pengujian

No.	Skenario	Input yang	Langkah	Hasil yang	
	Pengujian	Diberikan	Pengujian	Diharapkan	
1	Akses halaman	Pengguna sudah login	Klik menu Blast	Halaman Blast	
	Blast Email		Email dari	Email terbuka	
			dashboard	dengan form	
				pengisian	
2	Mengisi form	Subjek, isi email	Isi form secara	Placeholder	
	manual email	<pre>dengan {full_name},</pre>	manual, gunakan	tergantikan	
	dengan placeholder	{email}, {title},	placeholder dalam	otomatis saat	
		{name}	isi email	email dikirim	
3	Upload file Excel	File Excel berisi	Klik tombol	File berhasil	
	valid	email, nama, gelar	"Upload Excel"	diunggah dan	
			dan pilih file	nama file muncul	
				di kolom import	
4	Pilih penerima dari	Pilih data pelanggan	Centang beberapa	Data pelanggan	
	data pelanggan	dari daftar	pelanggan dari	muncul sebagai	
			daftar di sistem	penerima email	
5	Jadwalkan	Tanggal dan jam di	Isi kolom jadwal	Email masuk ke	
	pengiriman email	masa mendatang	dan klik tombol	antrian	
			"Kirim"	pengiriman	
				dengan status	
		VERS	SITA	pending	
6	Kirim email	Form lengkap dan	Isi semua kolom	Email langsung	
	langsung tanpa	valid	dan klik kirim	terkirim dan	
	jadwal			muncul di daftar	
	NU	JAN		pengiriman	
Lanjutan di halaman berikutnya					

Tabel 3.7. Skenario Pengujian Integrasi Fitur ke Sistem Email Blast

#### Lanjutan dari Tabel 3.7

7	Input email tidak valid dari Excel	Email dengan format salah (misal:	<i>Upload</i> Excel berisi email invalid	Email gagal dikirim dan
	_	halo.com)		berstatus "failed"
8	<i>Placeholder</i> tidak	Isi email dengan	Isi form manual	Placeholder
	dikenali	{nam}	dengan	tidak terganti,
			<i>placeholder</i> salah	ditampilkan
			ketik	sebagaimana
				ditulis
9	Data dari Excel	File Excel berisi data	Upload file Excel,	Data pelanggan
	tersimpan ke	pelanggan baru	lalu cek tabel	baru tersimpan
	database		customers	otomatis ke
	pelanggan			database

Berdasarkan implementasi dan pengujian fitur *Email Blast* yang tertera pada Tabel 3.7, sistem telah berhasil memenuhi kebutuhan komunikasi email yang efisien, fleksibel, dan personal. Integrasi data pelanggan, baik dari input manual maupun hasil unggahan Excel, telah terkelola dengan baik dan ditampilkan secara akurat. Fitur personalisasi melalui *placeholder* juga terbukti mampu meningkatkan kualitas isi pesan, sehingga penerima email mendapatkan kesan bahwa email tersebut memang ditujukan secara spesifik untuk penerima.

#### 3.5 Kondisi Sebelum dan Sesudah Pengembangan

#### 3.5.1 Sebelum Pengembangan

Sebelum dilakukan proses pengembangan, sistem *Email Blast* yang digunakan di PT Rembon Karya Digital hanya memiliki fitur dasar berupa pengiriman email massal secara manual atau dengan memanfaatkan template tertentu. Sistem juga sudah menyediakan kemampuan untuk mengimpor daftar penerima melalui unggahan file Excel. Namun, data yang diunggah tidak tersimpan secara permanen dalam basis data, sehingga tidak dapat diakses kembali untuk kebutuhan pengiriman berikutnya. Selain itu, belum tersedia modul khusus yang memungkinkan pengelolaan data pelanggan secara terpusat, seperti untuk menambahkan, memperbarui, atau menghapus data pelanggan secara langsung dari antarmuka sistem. Kemampuan untuk melakukan personalisasi isi email pun masih terbatas. Data personal hanya dapat dimasukkan dari isi file Excel yang diunggah,

tanpa adanya dukungan *placeholder* yang terintegrasi atau fleksibel untuk menyusun sapaan secara otomatis.

#### 3.5.2 Sesudah Pengembangan

Melalui proses pengembangan, sistem diperbarui dengan penambahan modul manajemen data pelanggan, yang memungkinkan pengguna untuk melakukan operasi *Create*, *Read*, *Update*, dan *Delete* (CRUD) secara langsung terhadap data pelanggan melalui antarmuka aplikasi. Selain itu, pengguna kini juga dapat menambahkan informasi gelar atau sebutan pelanggan melalui sistem relasi khusus yang menghubungkan data pelanggan dengan entitas titles.

Fitur unggah file Excel turut dikembangkan sehingga data yang berhasil diimpor secara otomatis tersimpan ke dalam basis data dan dapat ditampilkan pada halaman manajemen pelanggan. Pengembangan ini juga mencakup penambahan dukungan terhadap penggunaan *placeholder* seperti {title}, {name}, {full\_name}, dan {email} dalam isi email, sehingga sistem mampu menyusun sapaan secara personal langsung dari data pelanggan yang tersimpan.

#### 3.6 Kendala dan Solusi yang Ditemukan

Selama menjalani kegiatan magang di PT Rembon Karya Digital, terdapat berbagai tantangan yang dihadapi dalam proses pengembangan sistem. Setiap kendala yang muncul memberikan kesempatan untuk belajar dan beradaptasi secara langsung di lingkungan kerja profesional. Berikut ini merupakan beberapa kendala yang ditemukan beserta solusi yang diupayakan untuk mengatasinya.

#### 3.6.1 Kendala

- 1. Tidak adanya dokumentasi pada kode sumber menjadi hambatan utama selama proses pengembangan. Keseluruhan file program tidak disertai komentar yang memadai, sehingga menyulitkan pemahaman terhadap alur logika serta fungsi dari setiap bagian kode. Akibatnya, diperlukan waktu lebih lama untuk menelusuri dan mempelajari dokumentasi sistem secara manual.
- 2. Gangguan pada server email sempat terjadi selama beberapa minggu, sehingga fungsi pengujian dan pengiriman *email blast* tidak dapat dijalankan. Sayangnya, informasi mengenai gangguan ini tidak segera disampaikan kepada mahasiswa magang, sehingga menyebabkan miskomunikasi internal dan berimbas pada perlunya penyesuaian ulang terhadap sebagian logika sistem.

- 3. Hambatan komunikasi internal juga turut menjadi tantangan. Dalam beberapa situasi, instruksi dan penugasan yang diberikan tidak disampaikan secara jelas. Akibatnya, mahasiswa magang perlu datang langsung ke kantor untuk memperoleh kejelasan terkait detail pekerjaan yang harus diselesaikan.
- 4. Perubahan posisi selama masa magang juga berdampak pada proses kerja. Pergantian posisi dalam divisi mengharuskan adanya adaptasi ulang terhadap alur kerja serta redistribusi tanggung jawab yang sempat mengganggu kelancaran pengembangan sistem.

#### 3.6.2 Solusi

- Untuk mengatasi kurangnya dokumentasi dalam kode, langkah yang diambil adalah dengan menyusun dokumentasi pribadi dalam bentuk catatan serta peta alur sistem berdasarkan pemahaman terhadap isi kode. Pendekatan ini terbukti membantu mempercepat proses pengembangan selanjutnya.
- Ketika server email mengalami kendala, langkah pertama yang dilakukan adalah mengomunikasikan permasalahan tersebut kepada pihak terkait untuk memperoleh kejelasan. Setelah mendapat informasi yang dibutuhkan, pengujian dilanjutkan secara lokal.
- 3. Untuk mengurangi resiko miskomunikasi, dilakukan upaya peningkatan komunikasi melalui pengunggahan progres kerja ke dalam *platform* Trello dan pelaporan lanjutan melalui WhatsApp. Jika masih dibutuhkan kejelasan lebih lanjut, mahasiswa magang juga bersedia hadir langsung ke kantor untuk melakukan diskusi tatap muka.
- 4. Menanggapi perubahan struktur divisi, mahasiswa magang segera melakukan penyesuaian dengan anggota tim baru, meminta penjelasan mengenai wewenang serta ekspektasi dari pihak yang bertanggung jawab, dan menyelaraskan pekerjaan yang telah dilakukan dengan arahan yang diperbarui.

# UNIVERSITAS MULTIMEDIA NUSANTARA