SKRIPSI

PERANCANGAN DAN IMPLEMENTASI APLIKASI PENGELOLAAN PELANGGAN B2B DENGAN INTEGRASI EMAIL TEMPLATE BUILDER BERBASIS WEB



Disusun Oleh:

Nama : Muhammad Rayhan Ayman Mangku
NIM : 22523093

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2026

HALAMAN PENGESAHAN DOSEN PEMBIMBING

PERANCANGAN DAN IMPLEMENTASI APLIKASI PENGELOLAAN PELANGGAN B2B DENGAN INTEGRASI EMAIL TEMPLATE BUILDER BERBASIS WEB

TUGAS AKHIR JALUR MAGANG

Disusun Oleh:

Nama : Muhammad Rayhan Ayman Mangku
NIM : 22523093

Yogyakarta, 26 Juni 2026
Pembimbing,



Moh. Idris, S.Kom., M.Kom.

HALAMAN PENGESAHAN DOSEN PENGUJI

PERANCANGAN DAN IMPLEMENTASI APLIKASI PENGELOLAAN PELANGGAN B2B DENGAN INTEGRASI EMAIL TEMPLATE BUILDER BERBASIS WEB

TUGAS AKHIR JALUR MAGANG

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 27 Juli 2026

Tim Penguji

Moh. Idris, S.Kom., M.Kom. _____

Anggota 1

Sri Mulyati, S.Kom., M.Kom. _____

Anggota 2

Taufiq Hidayat, S.T., M.C.S., Ph.D. _____

Mengetahui,
Ketua Program Studi Informatika – Program Sarjana
Fakultas Teknologi Industri
Universitas Islam Indonesia

(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Muhammad Rayhan Ayman Mangku

NIM : 22523093

Tugas akhir dengan judul:

PERANCANGAN DAN IMPLEMENTASI APLIKASI PENGELOLAAN PELANGGAN B2B DENGAN INTEGRASI EMAIL TEMPLATE BUILDER BERBASIS WEB

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 29 Juni 2026



Muhammad Rayhan Ayman Mangku

HALAMAN PERSEMBAHAN

Karya ini saya persembahkan khususnya untuk kedua orang tua saya, yang tidak hanya memberikan kehidupan, tetapi juga dukungan bahwa segala sesuatu yang dihadapi dapat terselesaikan dengan baik apabila didasari oleh kerja keras dan usaha.

HALAMAN MOTO

Apa yang saya dapatkan hari ini tidak lain merupakan hasil dari kerja keras dan usaha yang telah dilakukan sebelumnya.

(Muhammad Rayhan Ayman Mangku)

KATA PENGANTAR

Assalamualaikum Wr. Wb.

Alhamdulillah penulis haturkan kepada Allah SWT, yang telah melimpahkan rahmat dan taufik serta hidayah-Nya sehingga penulis dapat menyelesaikan Laporan Akhir Jalur Magang di PT Varnion Technology Semesta. Magang adalah salah satu jalur kelulusan dari Program Studi Informatika di Universitas Islam Indonesia yang dapat ditempuh demi mendapatkan gelar sarjana. Magang dilaksanakan selama enam bulan dengan bekerja secara langsung di teknologi industri dan informasi. Selain itu, penulis memilih jalur magang agar mendapatkan ilmu, pengalaman bekerja, serta keahlian yang dibutuhkan di industri saat ini.

Sebagai penulis, saya mendapat dukungan dan bantuan dari berbagai pihak, oleh karena itu penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT atas segala rahmat dan hidayahnya sehingga tugas akhir ini dapat terselesaikan dengan baik.
2. Dosen pembimbing magang, Bapak Hanson Prihantoro Putro, S.T., M.T., yang telah membimbing penulis dalam proses magang.
3. Dosen pembimbing tugas akhir, Bapak Moh. Idris, S.Kom., M.Kom., yang telah membimbing penulis dalam seluruh proses penyusunan tugas akhir.
4. Kedua orang tua yang telah memberikan dukungan dan semangat yang luar biasa kepada penulis di setiap proses yang dijalani sampai dalam proses penyusunan tugas akhir ini.
5. Teman-teman seperjuangan yang selalu memberikan masukan dan bantuan serta dukungannya kepada penulis dalam proses magang hingga penyusunan tugas akhir.
6. Rekan kerja programmer dari divisi R&D PT Varnion Technology Semesta yang senantiasa memberi masukan dan saran.
7. Serta kepada *Product Lead* PT Varnion Technology Semesta, Bapak Akwari, S.E., M.M., selaku mentor dan atasan penulis selama magang.

Wassalamualaikum Wr. Wb.

Yogyakarta, 29 Juni 2026



Muhammad Rayhan Ayman Mangku

SARI

Penelitian ini membahas terkait proses perancangan dan pengembangan aplikasi pengelolaan data pelanggan B2B yang bernama My Varnion. Pengembangan aplikasi ini didasari untuk efisiensi proses operasional bisnis perusahaan serta sebagai sistem terpusat dalam pengelolaan data layanan dari pelanggan B2B yang berlangganan terhadap produk-produk internet Varnion. Kegiatan magang ini dilakukan di PT Varnion Technology Semesta, sebuah perusahaan penyedia layanan internet (*internet service provider*) di Indonesia. Selama magang, penulis sempat berperan sebagai *frontend developer,* kemudian beralih sebagai *fullstack developer* dalam mengembangkan proyek yang diberikan oleh *Product Lead*.

Manajemen proyek dilakukan dengan menggunakan pendekatan *Agile* yang mengadaptasi beberapa elemen dari kerangka kerja Scrum untuk mengakomodasi kebutuhan fleksibilitas yang tinggi pada divisi R&D. Dalam proyek ini, penulis bertugas dalam merancang dan implementasi antarmuka, logika bisnis, dan integrasi dengan layanan email Resend.

Hasil pengembangan proyek ini menunjukkan bahwa aplikasi pengelolaan pelanggan B2B (My Varnion) telah berhasil dirancang dan diimplementasikan. Sistem yang dibangun sesuai dengan dokumentasi fitur dan mampu memfasilitasi kebutuhan staf internal dalam mengelola data pelanggan, serta merancang fitur *email template builder.* Integrasi antara lapisan antarmuka, logika bisnis, dan basis data serta layanan pihak ketiga, API Resend telah diuji dan berjalan dengan baik. Oleh karena itu, dapat meningkatkan efisiensi operasional perusahaan.

Kata kunci: Pengelolaan Pelanggan B2B, Fullstack Developer, Scrum, Email Template Builder, Resend, Docker, React.js, Express.js, My Varnion.

GLOSARIUM

Agile	Filosofi dalam pengembangan proyek yang bersifat iteratif, digunakan dalam proses pengembangan perangkat lunak dengan membagi tugas besar ke dalam kumpulan tugas kecil.
Frontend Developer	Peran dalam membangun perangkat lunak yang mengerjakan sisi antarmuka aplikasi.
Backend Developer	Peran dalam membangun perangkat lunak yang mengerjakan sisi logika bisnis aplikasi.
Email Template Builder	Fitur yang memungkinkan pengguna dapat mengelola template email dan dapat dikirimkan kepada penerima.
End to end	Proses yang terintegrasi secara menyeluruh, melalui seluruh tahapan dari awal hingga akhir.
Business to business	Model bisnis di mana sebuah perusahaan menjual produk atau jasa kepada perusahaan lain, bukan ke pembeli perorangan.
Third Party	Layanan pihak ketiga yang digunakan untuk terintegrasi dengan sistem yang dikembangkan.
Deployment	Proses menyebarkan aplikasi atau layanan agar dapat diakses secara global melalui internet oleh pengguna.
API	Jembatan atau penghubung yang memungkinkan dua aplikasi atau sistem dapat terhubung untuk saling berkomunikasi dan bertukar data.
Agentic Service	Layanan digital otomatis yang digerakkan oleh kecerdasan buatan untuk membantu dalam menyelesaikan tugas-tugas tanpa perlu banyak bantuan manusia.
Endpoint	Alamat digital atau URL spesifik untuk mengakses data atau fungsi tertentu pada server dari aplikasi.
Key-value	Cara untuk menyimpan data secara berpasangan yang terdiri dari kunci unik dan isi datanya.
Environment Variables	Pengaturan rahasia yang disimpan diluar dari kode aplikasi, biasanya disimpan dalam bentuk file tersendiri dan tersimpan di dalam server.

Dashboard	Halaman visual yang menampilkan ringkasan data dan informasi penting dalam satu tempat.
Single Page Applications	Jenis website yang hanya memuat sekali halaman web tanpa melakukan penyegaran ulang pada konten di dalamnya ketika salah satu menu di klik.
Brute Force	Metode peretasan yang dilakukan dengan menebak kata sandi secara berulang.
REST API	Format khusus dari API yang wajib menggunakan jaringan internet dan protokol HTTP untuk bertukar data.
Subscription	Model bisnis dengan metode pembayaran secara berkala (bulanan atau tahunan) untuk mendapatkan atau mengakses suatu produk.
Prototype	Visualisasi awal sebagai gambaran dari sebuah produk yang akan dikembangkan.
AI-assisted development	Proses yang terintegrasi dengan alat bantu berbasis AI untuk mendukung efisiensi waktu dalam pengembangan.

DAFTAR ISI

HALAMAN PENGESAHAN DOSEN PEMBIMBING ii
HALAMAN PENGESAHAN DOSEN PENGUJI iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR iv
HALAMAN PERSEMBAHAN v
HALAMAN MOTO vi
KATA PENGANTAR vii
SARI viii
GLOSARIUM ix
DAFTAR ISI xi
DAFTAR TABEL xiii
DAFTAR GAMBAR xiv
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Ruang Lingkup 3
1.3 Tujuan 4
1.4 Manfaat 4
1.5 Sistematika Penulisan 5
BAB II LANDASAN TEORI DAN TINJAUAN PUSTAKA 7
2.1 Customer Relationship Management Business to Business 7
2.2 Email dan Email API 7
2.3 *Fullstack Developer* 8
2.4 Arsitektur Aplikasi Web 9
2.5 React.js 9
2.6 Express.js 10
2.7 PostgreSQL 10
2.8 Redis 11
2.9 Docker 11
2.10 Scrum 12
2.11 Tinjauan Pustaka 12
BAB III PELAKSANAAN MAGANG 16
3.1 Manajemen Proyek 16
3.2 Proyek My Varnion 16
3.3 *Tools* dalam proses Pengembangan My Varnion 18
3.4 Pengembangan Proyek My Varnion 21
3.4.1 Perancangan Sistem 21

3.4.2 Pemodelan Basis Data....26
3.4.3 Implementasi Antarmuka Aplikasi Pengelolaan Pelanggan B2B....30
3.4.4 Implementasi Fitur *Email Template Builder* & Integrasi API Resend....47
3.4.5 Hasil Pengujian Sistem....55
3.5 Kondisi Akhir dan Keberlanjutan Sistem....59
BAB IV REFLEKSI PELAKSANAAN MAGANG....60
4.1 Relevansi Akademik....60
4.2 Pembelajaran Magang....62
BAB V PENUTUP....64
5.1 Kesimpulan....64
5.2 Saran....64
DAFTAR PUSTAKA....66
LAMPIRAN....70

DAFTAR TABEL

Tabel 2.1 Penelitian Sejenis 12
Tabel 2.2 Perbandingan Penelitian Sebelumnya 14
Tabel 3.1 Jadwal Pelaksanaan Proyek 16
Tabel 3.2 Deskripsi *Use Case* 25
Tabel 3.3 Pemetaan Relasi Basis Data 28
Tabel 3.4 Hasil Pengujian Aplikasi 56

DAFTAR GAMBAR

Gambar 2.1 Scrum *Framework*.....12
Gambar 3.1 *Source Code* My Varnion dari *Product Lead* 17
Gambar 3.2 Tampilan Spreadsheet.....18
Gambar 3.3 Tampilan Antigravity.....19
Gambar 3.4 Tampilan Docker Desktop *Container Group* My Varnion.....20
Gambar 3.5 Tampilan Postman.....20
Gambar 3.6 Tampilan pgAdmin 21
Gambar 3.7 Arsitektur Sistem.....22
Gambar 3.8 *Use Case Diagram* 24
Gambar 3.9 Tampilan *Entity Relationship Diagram* (ERD) 27
Gambar 3.10 Antarmuka Halaman *Login* Internal 30
Gambar 3.11 Antarmuka Halaman Manajemen Pengguna Internal 31
Gambar 3.12 Antarmuka Halaman Manajemen Pelanggan B2B 32
Gambar 3.13 Antarmuka Editor Blog pada Halaman Manajemen Berita 33
Gambar 3.14 Antarmuka Formulir Tambah Acara pada Halaman Manajemen Acara 35
Gambar 3.15 Antarmuka Tambah FAQ pada Halaman Manajemen Pusat Bantuan.....36
Gambar 3.16 Antarmuka Daftar Notifikasi pada Halaman Manajemen Notifikasi.....37
Gambar 3.17 Antarmuka Halaman Manajemen Pertemuan 38
Gambar 3.18 Antarmuka Halaman Manajemen Tiket.....39
Gambar 3.19 Antarmuka Halaman Login Pelanggan B2B.....40
Gambar 3.20 Antarmuka Halaman Dashboard Pelanggan 41
Gambar 3.21 Antarmuka Halaman Pengaturan Pelanggan.....42
Gambar 3.22 Antarmuka Halaman Tiket Pelanggan 43
Gambar 3.23 Antarmuka Halaman Pengajuan Pertemuan Pelanggan.....45
Gambar 3.24 Antarmuka Halaman Berita & Acara Pelanggan 46
Gambar 3.25 Antarmuka Halaman FAQ Pelanggan.....47
Gambar 3.26 Antarmuka Editor Halaman *Email Template*.....49
Gambar 3.27 Antarmuka Pengisian Variabel Dinamis pada Halaman *Use Template*.....50
Gambar 3.28 Tampilan Hasil *Template Email* yang Terkirim pada Gmail.....50
Gambar 3.29 Tampilan *Dashboard* Resend.....51
Gambar 3.30 Konfigurasi Fungsi Pengiriman Email.....52
Gambar 3.31 Konfigurasi Rute *Endpoint* API *campaign.route.ts* 53

Gambar 3.32 Konfigurasi Variabel Lingkungan *.env.docker* 54
Gambar 3.33 Halaman *Settings* portal internal *my-varnion-admin* 55

BAB I
PENDAHULUAN

1.1 Latar Belakang

PT Varnion Technology Semesta merupakan perusahaan penyedia layanan internet (Internet Service Provider/ISP) dan solusi teknologi informasi yang berbasis di Indonesia. PT Varnion Technology Semesta yang dikenal sebagai Varnion didirikan pada tahun 2007 oleh Alexander Budiman, seorang figur yang telah berkiprah di dunia internet sejak tahun 1996. Varnion lahir dengan visi besar "*Technology for Everyone*", Varnion menghadirkan beragam layanan yang mengutamakan keandalan koneksi, keamanan jaringan, serta integrasi yang mulus dengan sistem internal pelanggan korporasi. Selain itu, Varnion telah bekerja sama dengan lebih dari 400 mitra yang tersebar di berbagai wilayah di Indonesia, seperti perhotelan dan organisasi.

Selama proses magang, penulis mengerjakan tiga proyek yang berkaitan dengan internal perusahaan, yaitu Pixku, Arclab, dan My Varnion. Setiap proyek yang dikerjakan memiliki durasi pengembangan dan latar belakang yang berbeda. Pixku merupakan proyek internal dan berfungsi sebagai alat pembuat tampilan web (*page builder*) dalam mengelola konten pada website utama PT Varnion Technology Semesta, proyek ini merupakan proyek pertama dan merupakan proyek pembaruan (*revamp)* yang dikerjakan selama kurang lebih tiga bulan oleh penulis. Setelah proyek Pixku dikerjakan, penulis mendapat tugas dalam mengerjakan proyek Arclab, sebelum mengawali dalam mengerjakan proyek ini, penulis dan *Product Lead* melakukan pengarahan (*briefing*) terkait fitur dan konsep dari proyek Arclab, proyek ini merupakan proyek internal perusahaan yang berfungsi sebagai manajemen pengiriman email kepada klien dan pengelolaan *template email* yang direncanakan akan digunakan untuk internal perusahaan serta dijual kepada klien dalam bentuk langganan (*subscription*).

Selanjutnya, pada konteks hubungan *business-to-business* (B2B), pengelolaan pelanggan korporasi umumnya lebih rumit dibandingkan pelanggan individu. Prosesnya melibatkan pemangku kepentingan, siklus kontrak yang lebih panjang, serta kebutuhan dokumentasi yang lebih detail. Komunikasi dengan pelanggan korporasi biasanya dilakukan melalui banyak cara, seperti pertemuan langsung, sistem tiket, pesan instan, dan email. Jika tidak dikelola secara terpusat, informasi yang tersebar di berbagai media berpotensi menimbulkan data tidak konsisten dan kesalahpahaman. Kondisi tersebut juga menyulitkan

perusahaan dalam memantau status layanan, melihat riwayat komunikasi, dan memastikan setiap permintaan pelanggan direspons dengan tepat waktu.

Dalam rangka memenuhi kebutuhan tersebut, Varnion mengembangkan sistem pengelolaan pelanggan B2B berbasis web yang bernama My Varnion sebagai langkah awal untuk pengembangan *agentic service* yang mendukung fitur *chat* berbasis tiket dan solusi terkait akun pelanggan. Sistem ini dirancang untuk memusatkan pengelolaan layanan, profil pelanggan, serta mendukung komunikasi operasional antara perusahaan dan pelanggan.

Sebelum aplikasi My Varnion dikembangkan, proses pengelolaan pelanggan B2B di lingkungan Varnion masih banyak mengandalkan pencatatan manual, penggunaan dokumen yang tersebar di berbagai platform, serta pengiriman email secara terpisah oleh masing-masing staf. Oleh karena itu, pola kerja seperti ini menyulitkan proses operasional perusahaan dan dinilai kurang efektif dalam mengelola data pelanggan.

Di tengah kondisi tersebut, email merupakan salah satu metode komunikasi utama yang digunakan untuk menyampaikan berbagai informasi penting kepada pelanggan, seperti pengumuman pemeliharaan jaringan, pembaruan layanan, maupun informasi promosi terkait produk-produk perusahaan. Namun, ketika pengelolaan email dilakukan secara manual tanpa dukungan sistem terintegrasi, penyusunan konten, pengaturan tampilan, dan pengiriman ke banyak penerima menjadi kurang efisien dan rawan kesalahan. Oleh karena itu, diperlukan sebuah aplikasi pengelolaan pelanggan B2B yang tidak hanya menyimpan dan menampilkan data pelanggan, tetapi juga menyediakan fitur *email template builder* yang memungkinkan pengguna internal dapat menyusun dan menyesuaikan tampilan email secara fleksibel melalui antarmuka berbasis web sehingga email yang dikirim tetap konsisten ketika diterima oleh pelanggan. Di sisi lain, dengan memanfaatkan sistem pengelolaan pelanggan dan otomatisasi email yang terhubung langsung dengan basis data pelanggan dapat membantu perusahaan menyampaikan informasi dengan lebih konsisten.

Selama melaksanakan kegiatan magang, penulis berkontribusi secara langsung dalam pengembangan aplikasi My Varnion sebagai aplikasi pengelolaan pelanggan B2B berbasis web. Dalam proses tersebut, penulis terlibat mulai dari perancangan antarmuka hingga penyusunan struktur basis data yang digunakan oleh aplikasi. Pada sisi antarmuka, penulis menggunakan React.js sebagai kerangka untuk membangun tampilan yang interaktif dan mudah dikembangkan kembali di masa mendatang. Sementara itu, pada sisi server, penulis menggunakan Express.js sebagai kerangka kerja untuk mengelola logika bisnis, mengatur rute API, dan menghubungkan aplikasi dengan basis data.

1.2 Ruang Lingkup

Ruang lingkup dan batasan aktivitas selama kegiatan magang penulis mencakup tiga proyek yang dikerjakan termasuk pengembangan sistem pengelolaan pelanggan *business-to-business* (B2B) yang bernama My Varnion. Ketiga proyek ini dibangun dengan tujuan untuk meningkatkan efisiensi operasional pada PT Varnion Technology Semesta.

Lingkup proyek serta batasan aktivitas yang dilakukan penulis dalam mengerjakan tiga proyek adalah sebagai berikut:

a. Proyek Pixku

   Pengembangan *revamp* proyek Pixku dilakukan mulai dari bulan September sampai November 2025. Pada proyek ini, penulis tidak melakukan penambahan API *key* baru atau kredensial lainnya pada proses konfigurasi, melainkan menggunakan sumber daya yang sudah ada dari versi sebelumnya agar sistem dapat dikembangkan baik dari sisi antarmuka maupun logika bisnis serta basis data. Oleh karena itu, pengembangan hanya berfokus pada implementasi antarmuka versi terbaru dan perbaikan sisi *backend* menyesuaikan dengan kebutuhan pembaruan sistem.

b. Proyek Arclab

   Pengembangan proyek Arclab dilakukan pada akhir bulan November sampai Januari 2026. Pada proyek ini, penulis tidak melakukan pendaftaran akun di layanan Resend untuk mendapatkan data kredensial karena tugas tersebut dikerjakan oleh *Product Lead*. Selain itu, penulis tidak melakukan pengujian pengiriman email kepada klien secara rill karena data pelanggan asli belum terdaftar pada sistem, sehingga masih menggunakan data tiruan saat proses pengembangan.

c. Proyek My Varnion

   Pengembangan proyek My Varnion dilakukan pada bulan Januari sampai Maret 2026. Pada proyek ini, penulis tidak melakukan pendaftaran akun di layanan Resend demi mendapatkan data kredensial seperti API *key*, melainkan menggunakan *environtment* yang sudah digunakan pada proyek Arclab. Penulis mengerjakan seluruh lapisan sesuai dengan tiga lapisan utama aplikasi pada proyek ini. Selain itu, penulis tidak melakukan pengujian pengiriman email dan fitur tiket kepada klien secara rill karena data pelanggan asli belum terdaftar pada sistem, sehingga masih menggunakan data tiruan saat proses pengembangan.

1.3 Tujuan

Tujuan dari pelaksanaan magang dan penyusunan laporan tugas akhir ini adalah untuk mendokumentasikan kontribusi penulis dalam pengembangan platform internal PT Varnion Technology Semesta melalui tiga proyek utama yang telah dikerjakan, diantaranya sebagai berikut:

a. Proyek Pixku

   Proyek ini dikembangkan untuk membuat *page builder* terhadap website utama Varnion, sehingga memungkinkan untuk mengelola tampilan dan isi konten melalui satu platform terpusat.

b. Proyek Arclab

   Proyek ini didasari sebagai tahap awal membentuk media atau platform *email template builder,* sehingga memungkinkan untuk mengelola *email marketing* dan transaksional untuk ditujukan kepada pelanggan produk-produk Varnion.

c. Proyek My Varnion

   Proyek ini dikembangkan sebagai langkah awal pengembangan *agentic service* yang mendukung fitur *chat* berbasis tiket dan solusi terkait akun pelanggan. Selain itu, proyek ini juga mengembangkan fitur *email template builder* seperti yang dimiliki proyek Arclab.

1.4 Manfaat

Berdasarkan tujuan yang telah dirumuskan, manfaat yang diharapkan dari pelaksanaan magang dan penulisan tugas akhir ini adalah sebagai berikut:

a. Optimalisasi Pengelolaan Konten (Proyek Pixku):

   Meningkatkan efisiensi tim internal dalam melakukan pembaruan konten pada website utama perusahaan melalui sistem *page builder* yang lebih responsif dan fleksibel sehingga meminimalkan ketergantungan pada proses *coding* manual untuk sekedar mengubah konten.

b. Profesionalisme Komunikasi Bisnis (Proyek Arclab):

   Memberikan kemudahan bagi perusahaan dalam mengelola korespondensi klien secara terpusat dan konsisten. Selain itu, kedepannya diharapkan untuk dikembangkan dengan menambahkan modul atau fitur langganan (*subscription*) yang dapat memberikan peluang bagi perusahaan untuk melakukan monetisasi layanan manajemen email.

c. Efisiensi Operasional B2B (Proyek My Varnion):

   Mempercepat proses penanganan data pelanggan korporasi dan memastikan penyampaian informasi penting (seperti notifikasi pemeliharaan jaringan) tersampaikan kepada klien melalui integrasi API Resend.

d. Keandalan dan Skalabilitas Infrastruktur:

   Memberikan dokumentasi teknis terkait lingkungan pengembangan menggunakan Docker Compose, yang dapat memudahkan tim *R&D* perusahaan dalam melakukan pemeliharaan sistem serta memilimalkan risiko perbedaan konfigurasi antar lingkungan kerja.

e. Pengembangan Kompetensi:

   Meningkatkan keahlian dan menambah pengetahuan penulis dalam mengimplementasikan arsitektur *full-stack,* mulai dari manajemen basis data PostgreSQL hingga integrasi layanan pihak ketiga (*third-party*) dalam skala industri nyata.

1.5 Sistematika Penulisan

Penyajian laporan skripsi ini menggunakan sistematika penulisan sebagai berikut:

1. **Bagian Utama**

   Bagian utama terbagi dari bab dan sub bab sebagai berikut:

   1. **BAB I PENDAHULUAN**

      Bab ini berisi mengenai latar belakang penulisan, laporan tugas akhir, ruang lingkup, tujuan, manfaat, dan sistematika dari penulisan laporan tugas akhir.

   2. **BAB II LANDASAN TEORI DAN TINJAUAN PUSTAKA**

      Bab ini membahas mengenai landasan teori yang berkaitan dengan alat dan metode yang digunakan selama proses pengembangan.

   3. **BAB III PELAKSANAAN MAGANG**

      Bab ini membahas mengenai aktivitas penulis selama mengikuti kegiatan magang, penjelasan terkait proyek yang dikerjakan selama magang, pembahasan aplikasi My Varnion yang dikerjakan, serta metode yang digunakan selama proses pengembangan di PT Varnion Technology Semesta.

4. **BAB IV REFLEKSI PELAKSANAAN MAGANG**

   Bab ini membahas terhadap pembelajaran yang didapatkan selama menjalankan kegiatan magang dan tantangan yang dihadapi, serta keterampilan yang berhasil dikembangkan.

5. **BAB V PENUTUP**

   Bab ini berisi kesimpulan dan saran terhadap apa yang akan dilakukan untuk pengembangan selanjutnya.

BAB II
LANDASAN TEORI DAN TINJAUAN PUSTAKA

2.1 Customer Relationship Management Business to Business

Customer Relationship Management (CRM) merupakan strategi efektif yang berfokus pada pengelolaan hubungan jangka panjang dengan pelanggan melalui pemanfaatan data dan teknologi, dengan tujuan meningkatkan kualitas layanan, kepuasan, dan loyalitas pelanggan. Strategi ini diwujudkan melalui perancangan sistem yang dapat mengumpulkan, menyimpan, dan mengintegrasikan data pelanggan dalam satu platform secara terpusat (Suwandi et al., 2023). Dalam konteks *business-to-business* (B2B), pelanggan yang dikelola adalah organisasi, lembaga, atau perusahaan dengan hubungan kontrak kerja sama jangka panjang sehingga penerapan CRM B2B lebih menekankan pada pengelolaan akun pelanggan perusahaan dan pemenuhan kebutuhan setiap pelanggan agar kualitas layanan meningkat, hubungan bisnis semakin kuat, dan loyalitas pelanggan dapat terjaga (Quzwen et al., 2021).

Untuk mendukung hal tersebut, dalam pengembangan aplikasi My Varnion, CRM B2B dimanfaatkan sebagai wadah untuk mendukung koordinasi antar divisi internal penyedia layanan dalam melayani perusahaan pelanggan. Sistem ini dirancang untuk memfasilitasi beberapa fungsi utama, seperti pencatatan komunikasi, pengelolaan data layanan pelanggan yang sedang berjalan, serta menyediakan portal pelanggan berbasis web yang dapat diakses langsung oleh perusahaan mitra. Melalui portal ini, pelanggan dapat menerima notifikasi terkait informasi terbaru layanan, menjadwalkan atau melihat agenda pertemuan dengan pihak Varnion, membuat dan memantau tiket dukungan (support) ketika terjadi kendala pada layanan, mengakses konten berupa *Frequently Asked Questions* (FAQ), serta memperoleh informasi terbaru melalui fitur berita dan acara yang disediakan. Dengan fitur-fitur tersebut, diharapkan dapat meningkatkan keterlibatan pelanggan, mempercepat proses penanganan masalah oleh internal, dan memberikan pengalaman layanan yang lebih aktif.

2.2 Email dan Email API

Email transaksional merupakan pesan yang dikirim sebagai respon atas tindakan tertentu yang dilakukan pelanggan, dengan fokus untuk menyampaikan informasi penting terkait sebuah tindakan, seperti informasi transaksi pembelian dan penagihan (Sangeetha, 2016). Pada aplikasi My Varnion, konsep email transaksional digunakan untuk mengirimkan

email notifikasi yang berisi informasi terkait layanan. Email transaksional pada sistem ini dikelola dalam bentuk *email template*, yang merupakan kerangka pesan yang memuat struktur dan tampilan tetap seperti subjek, *header*, isi utama konten yang berisi informasi, dan *footer* yang berisi keterangan tambahan. Penggunaan *email template* memungkinkan konten atau kerangka email hanya disusun sekali dan dapat digunakan kembali dengan menyisipkan data dinamis sehingga semua struktur tampilan email pada setiap *template* terlihat seragam sementara informasi spesifik pelanggan pada konten dari *template* dapat disesuaikan dengan resiko kesalahan manual yang rendah.

Email API merupakan antarmuka pemrograman aplikasi yang menghubungkan aplikasi dengan penyedia layanan email sehingga aplikasi dapat mengirim email secara terprogram melalui pemanggilan *endpoint* API tanpa harus berinteraksi langsung dan mengelola server *SMTP* sendiri (Kristyna, 2024). Salah satu penyedia layanan tersebut adalah Resend, merupakan sebuah platform pengiriman email modern yang dirancang khusus untuk pengembang agar dapat mengintegrasikan fungsi email ke dalam aplikasi dengan lebih mudah. Dalam sistem pada aplikasi My Varnion, Resend digunakan sebagai layanan pengiriman email notifikasi, melalui REST API. Melalui integrasi tersebut, pengguna internal pada aplikasi My Varnion dapat menyusun dan mengirim email yang telah dibuat dengan menyertakan konten HTML, subjek, dan daftar penerima secara terstruktur langsung dari sisi server tanpa perlu konfigurasi dan mengelola infrastruktur email secara mandiri (Resend, n.d.).

2.3 Fullstack Developer

Fullstack developer merupakan seorang yang mengerjakan dua posisi sekaligus pada proses pengembangan aplikasi, yaitu berperan dalam pengembangan sisi *frontend* dan *backend.* Pada sisi *frontend,* pengembang berfokus dalam merancang antarmuka aplikasi yang telah ditetapkan oleh tim *designer* dan mengintegrasikannya dengan server aplikasi (Fitrianto, 2023). Selain itu, pada sisi *backend,* pengembang bekerja di balik layar dan berhubungan dengan server, seperti merancang basis data dan menyediakan antarmuka layanan (API) yang dapat digunakan pada sisi *frontend* (Sarmanela et al., 2023).

Dalam proses pengembangan aplikasi My Varnion, penulis menjalankan peran sebagai *fullstack developer* dengan membagi dua fokus utama, yaitu pembuatan antarmuka dan pengembangan sisi server. Pada sisi *frontend*, penulis menggunakan React.js untuk merancang tampilan dan interaksi pengguna pada portal pelanggan maupun portal internal agar informasi layanan dan fitur CRM B2B dapat diakses dengan mudah dan saling terhubung. Pada sisi

backend, penulis mengembangkan layanan server menggunakan Express.js untuk mengelola logika bisnis, merancang struktur basis data, serta menyediakan *endpoint* API yang dapat digunakan pada sisi *frontend* untuk menghubungkan antarmuka dengan penyimpanan data dan layanan eksternal *email API* seperti platform Resend.

2.4 Arsitektur Aplikasi Web

Aplikasi berbasis web banyak digunakan sebagai media interaksi antara pelanggan dan penyedia layanan karena dapat diakses dari berbagai perangkat selama terhubung dengan koneksi internet. Arsitektur aplikasi terbagi dalam beberapa lapisan (*layer*), dan setiap lapisan memiliki fungsi serta respons yang berbeda. Lapisan dari arsitektur terdiri atas tiga lapisan utama, yaitu presentasi, logika bisnis, dan akses data. Dengan menggunakan arsitektur berlapis, aplikasi yang dikembangkan dapat lebih modular dan lebih mudah dikembangkan dimasa mendatang.

Lapisan presentasi (*presentation layer*) berperan dalam menampilkan antarmuka, menangani interaksi pengguna seperti memproses *input* pengguna, dan mengirim permintaan pengguna ke lapisan logika bisnis. Lapisan logika bisnis (*application layer*) merupakan inti dari sistem, seperti menangani logika bisnis, mengelola alur, dan memproses data yang diteruskan dari lapisan presentasi. Sementara itu, lapisan terakhir dari tiga lapisan utama yaitu lapisan akses data (*data access layer*), lapisan ini berperan dalam interaksi dan terhubung langsung dengan penyimpanan data karena lapisan ini berfungsi untuk menyimpan dan menyediakan data secara terstruktur. Selain itu, lapisan ini juga mendukung beberapa proses seperti pengambilan, penulisan, dan pembaruan data (Tu, 2023).

Pendekatan dengan menggunakan tiga lapisan utama pada arsitektur aplikasi sejalan dengan kebutuhan sistem berbasis web pada aplikasi My Varnion karena mendukung pengembangan yang berkelanjutan, integrasi dengan berbagai layanan salah satunya seperti layanan pengiriman email, serta akses pengguna yang terus berkembang.

2.5 React.js

Teknologi yang digunakan pada pengembangan aplikasi My Varnion terdapat beberapa pada setiap lapisan (*layer*). Pada lapisan presentasi, pengembangan untuk membangun antarmuka menggunakan *framework* React.js. React merupakan *framework* JavaScript yang memiliki beberapa keunggulan dibandingkan *framework* lain. Dengan menggunakan React, proses pengembangan pada suatu proyek dapat lebih cepat karena memiliki struktur yang

modular sehingga memungkinkan pengembangan menjadi fleksibel dan dapat dipelihara karena komponen yang dibangun diatas kerangka React dapat berdiri secara mandiri serta dirancang untuk kinerja tinggi (Komperla et al., 2022).

Dalam konteks pengembangan pada lapisan presentasi, pemanfaatan ekosistem React menjadi semakin penting karena mendukung pengembangan aplikasi *single-page-application* (SPA) yang memungkinkan konten halaman pada web diperbarui secara dinamis berdasarkan aksi pengguna terhadap suatu komponen tanpa perlu memuat ulang semua komponen sekaligus (Davis, 2023). Oleh karena itu, pendekatan ini sangat cocok dengan aplikasi My Varnion yang memiliki tampilan antarmuka berupa *dashboard*, karena dapat meningkatkan kenyamanan pengguna ketika mengakses portal pelanggan melalui navigasi yang lebih responsif dan perpindahan halaman yang terasa lebih cepat setelah pemuatan awal aplikasi.

2.6 Express.js

Pada lapisan logika bisnis, pengembangan dilakukan dengan menggunakan *framework* dari Node.js yaitu Express.js. Express.js merupakan kerangka kerja aplikasi *backend* yang dibangun menggunakan bahasa pemrograman JavaScript. Penggunaan kerangka ini memudahkan proses pengembangan karena menyediakan struktur yang sederhana, sistem *routing* yang jelas, serta ekosistem *middleware* yang luas sehingga pengembang dapat membangun layanan web dan REST API dengan lebih cepat dan terorganisir. Selain itu, Express.js juga mendukung penggunaan *middleware* untuk menangani berbagai kebutuhan, seperti validasi data, penanganan *error*, serta mekanisme keamanan dan autentikasi (Nahjan et al., 2023).

2.7 PostgreSQL

Pada lapisan akses data, pengembangan basis data menggunakan sistem manajemen basis data objek relasional PostgreSQL yang bersifat sumber terbuka (*open source*) (Saiholau, 2024). Dengan menggunakan PostgreSQL, memungkinkan konsistensi dan performa dalam proses pengambilan data terjadi secara optimal (Simarmata et al., 2025). Selain itu, PostgreSQL memiliki keunggulan dalam mengelola data jumlah besar, didukung standar SQL yang lengkap, serta memiliki keunggulan dalam optimasi *query* yang memungkinkan proses analisa atau pencarian data lebih cepat dan efisien (Sselvianto et al., 2025).

2.8 Redis

Redis merupakan basis data NoSQL yang menyimpan data berbasis *key-value* dan berjalan di memori. Memungkinkan pembatasan jumlah permintaan dari pengguna dapat diterapkan secara efisien dan proses pengambilan data terkait mekanisme tersebut dapat dilakukan dengan lebih cepat (Ridhalri, 2022). Dalam konteks penggunaannya pada proyek My Varnion, Redis memiliki peran utama sebagai *Distributed Rate Limiter Storage* yang berfungsi untuk mengelola dan menyinkronisasikan batasan jumlah permintaan (*rate limiting*) di seluruh *instance* server untuk melindungi API dari serangan *brute force* dan penyalahgunaan lalu lintas (*traffic),* sekaligus memastikan performa sistem tetap optimal dengan latensi penyimpanan data yang sangat rendah. Namun pada mode pengembangan, Redis tidak digunakan secara *default* agar dapat menyederhanakan arsitektur lokal dan mempermudah *debugging*, Selain itu, membuat setup proyek di lingkungan lokal jauh lebih ringan karena pengembang tidak wajib menjalankan *container* Redis walaupun tidak akan berfungsi pada mode pengembangan lokal. Oleh karena itu, Redis akan diaktifkan secara penuh pada lingkungan produksi (*production*).

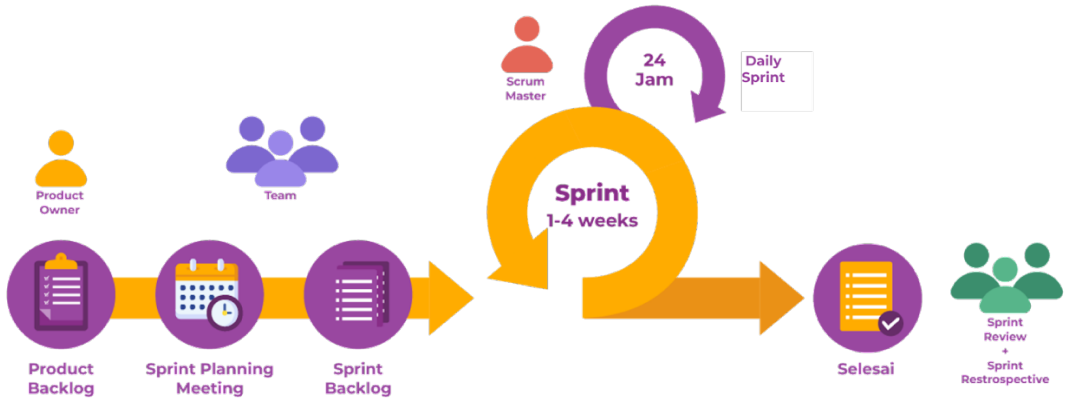
2.9 Docker

Docker adalah teknologi virtualisasi berbasis kontainer sumber terbuka (*open source)* yang memungkinkan aplikasi berjalan dalam lingkungan virtual yang ringan bagi aplikasi (Prasetyo, 2021). Untuk mengelola ekosistem tersebut, Docker menyediakan alat berupa Docker Compose yang memungkinkan pengelolaan layanan secara terisolasi dan berjalan secara mandiri dalam sebuah konfigurasi yang terstruktur. Docker Compose merupakan alat yang disediakan dari Docker yang memungkinkan sistem berjalan lebih efisien dengan konfigurasi lingkungan yang terisolasi dan mudah dikembangkan karena memastikan setiap layanan dapat berdiri sendiri tanpa konflik dengan sistem lain. Setiap layanan dikemas dalam *container* yang terpisah dan memiliki dependensi masing-masing sehingga proses pemeliharaan hingga *deployment* dapat dilakukan secara konsisten di berbagai lingkungan (Saputra & Firdonsyah, 2025). Pendekatan ini sangat bermanfaat karena dapat menjalankan lebih dari satu *container* dengan menggunakan satu konfigurasi yang terstruktur.

Docker Compose memungkinkan pengembang mendefinisikan berbagai layanan yang digunakan beserta konfigurasi seperti *image*, *port*, *volume*, dan pengaturan jaringan dalam satu file konfigurasi. Melalui pendekatan ini, proses penyusunan, *deployment*, serta replikasi lingkungan aplikasi di berbagai sistem menjadi terstruktur dan konsisten (Hartadi, 2025).

2.10 Scrum

Scrum merupakan sebuah kerangka kerja (*framework*) dan bagian dari *Software Development Life Cycle* (SDLC), serta menggunakan pendekatan *Agile* untuk menghasilkan suatu nilai melalui solusi adaptif sehingga ditujukan untuk mengatasi masalah yang kompleks (Hardiansyah et al., 2023). Penggunaan Scrum dilakukan dengan cara membagi pengembangan perangkat lunak ke dalam siklus-siklus berkelanjutan yang memungkinkan hasil kerja lebih mudah diprediksi (Warkim et al., 2020).



Gambar 2.1 Scrum *Framework*

Sumber: (Sabrina, 2024)

2.11 Tinjauan Pustaka

Berikut ini merupakan sejumlah penelitian yang dilakukan sebelumnya dan mengangkat topik terkait aplikasi pengelolaan pelanggan (B2B), pemanfaatan *Customer Relationship Management* (CRM), serta integrasi dengan layanan pengiriman email berbasis web.

Tabel 2.1 Penelitian Sejenis

No	Judul	Tahun	Pembahasan
1	Desain Aplikasi B2b Sistem Manajemen Pergudangan dalam Penunjang Keputusan bisnis (Roihan et al., 2019).	2019	Penelitian ini membahas terkait perancangan dan pembangunan aplikasi B2B sistem manajemen pergudangan berbasis web untuk PT Super Tata Raya Steel untuk mengatasi masalah pengelolaan data produk yang masuk dan keluar yang sebelumnya masih ditangani secara manual oleh bagian gudang dan ekspedisi.

No	Judul	Tahun	Pembahasan
2	Customer Relationship Management in Business-toBusiness Marketing: The Impact on Corporate Performance (Quzwen et al., 2021b)	2021	Penelitian ini membahas terkait pengaruh implementasi CRM terhadap kinerja perusahaan B2B di industri kurir Indonesia, dengan melihat peran faktor organisasi, strategi, budaya, serta bagaimana ketidakstabilan pasar melemahkan hubungan antara CRM dan kinerja perusahaan.
3	Perancangan Sistem Informasi B2B Berbasis Website pada PT. Mitra Makmur Dwijaya (Darmawan & Christianto, 2026)	2026	Penelitian ini merancang dan membangun sistem informasi B2B berbasis website untuk PT Mitra Makmur Dwijaya untuk otomatisasi terkait proses penawaran harga, order pembelian, invoice, pelacakan pesanan, dan pengembalian agar transaksi lebih cepat, akurat, dan transparan.
4	Implementasi Sistem Customer Relationship Management (CRM) Berbasis Website untuk Pengelolaan Membership dan Loyalty Point pada CV. Honda Karya Utama Kisaran (Rezkika et al., 2026)	2026	Membahas integrasi fitur loyalitas (poin) dan keanggotaan pada dealer otomotif. Dari implementasi terhadap fitur-fitur tersebut, hasilnya dapat mempercepat efisiensi waktu transaksi dan mendukung pengambilan keputusan manajerial berbasis data riwayat pelanggan.
5	Pengaruh Customer Relationship Management (CRM) Terhadap Customer Satisfaction dan Customer Loyalty (Azani, 2025)	2026	Penelitian ini menganalisis beberapa penelitian yang menunjukkan bahwa penerapan CRM berpengaruh positif dalam memperkuat hubungan serta meningkatkan kepuasan dan loyalitas pelanggan.
6	Transformasi Digital dalam Industri Distribusi Bahan Bangunan (Studi Kasus: Uji Kepuasan Platform B2B pada PT. Bumi Pembangunan Pertiwi di Kab. Madiun, Jawa Timur) (Pradana et al., 2025)	2025	Membahas tentang transformasi digital pada PT. Bumi Pembangunan Pertiwi di Madiun melalui pengembangan platform B2B untuk mengatasi hambatan operasional seperti pengelolaan stok yang tidak akurat, penagihan berbasis kertas, dan prosedur persetujuan yang lambat.
7	Implementasi Metode Scrum dan Analisis SWOT Sebagai Strategi Framework Customer Relationship Management (CRM) Pada Perusahaan Aqilla Rent Car Semarang (Raharjo, 2023)	2023	Penelitian ini membahas terkait implementasi metode Scrum dan analisis SWOT sebagai strategi untuk merancang *framework* Customer Relationship Management (CRM) di perusahaan Aqilla Rent Car Semarang.
8	Implementasi Customer Relationship Management (CRM) pada Sistem Informasi Pemasaran dengan Menggunakan Framework React.js (Muhammad et al., 2022)	2021	Penelitian ini membahas tentang perancangan sistem informasi berbasis web menggunakan *framework* React.js dengan menerapkan konsep Customer Relationship Management (CRM) pada CV. Reswara Arvin Jaya.

No	Judul	Tahun	Pembahasan
9	ARIA–Automated Requirement & Interaction Agent (AI-Powered Multi-Channel Client Communication and Task Management System)	2026	Penelitian ini mengembangkan sebuah sistem manajemen tugas dan komunikasi klien berbasis AI untuk mengotomatisasi pengumpulan kebutuhan proyek secara banyak saluran melalui WhatsApp, Slack, dan portal web, yang secara otomatis mengeksekusi alur kerja digital seperti pembuatan kartu tugas di Trello, pengiriman notifikasi, serta pemantauan proyek secara real-time melalui *dashboard* admin pada sistem dari artikel ini.
10	Implementasi Customer Relationship Management (CRM) pada Toko Roti Ganda Berbasis Web (Siallagan et al., 2020)	2020	Penelitian ini membahas tentang perancangan sistem e-commerce berbasis web menggunakan pendekatan *Customer Relationship Management* (CRM) dan metode *User Centered Design* (UCD) pada Toko Roti Ganda Pematangsiantar.

Tabel 2.2 Perbandingan Penelitian Sebelumnya

No	Aplikasi Web	Menggunakan Konsep CRM	Konteks B2B	Integrasi Email API	Email Template Builder	Menggunakan Scrum
1	Ya	Tidak	Ya	Tidak	Tidak	Tidak
2	Tidak	Ya	Ya	Tidak	Tidak	Tidak
3	Ya	Tidak	Ya	Tidak	Tidak	Tidak
4	Ya	Ya	Tidak	Tidak	Tidak	Tidak
5	Tidak	Ya	Tidak	Tidak	Tidak	Tidak
6	Ya	Tidak	Ya	Tidak	Tidak	Ya
7	Ya	Ya	Tidak	Tidak	Tidak	Ya
8	Ya	Ya	Tidak	Tidak	Tidak	Tidak

9	Ya	Ya	Ya	Ya	Tidak	Ya
10	Ya	Ya	Tidak	Tidak	Tidak	Tidak
11 (Penelitian Ini)	Ya	Ya	Ya	Ya	Ya	Ya

Berdasarkan tabel perbandingan pada penelitian sebelumnya, penelitian ini dapat disimpulkan memiliki keunggulan karena menggabungkan beberapa aspek yang belum dimanfaatkan secara menyeluruh pada penelitian-penelitian sebelumnya, yaitu pengembangan aplikasi pengelolaan pelanggan dalam konteks CRM B2B berbasis web, terintegrasi dengan *email API* dari Resend, dilengkapi dengan modul *email template builder* berbasis web, sehingga *template* email yang akan ditujukan kepada klien dapat disesuaikan isi kontennya berdasarkan kebutuhan, serta proses pengembangan dilakukan dengan menggunakan metode Scrum.

BAB III
PELAKSANAAN MAGANG

3.1 Manajemen Proyek

Kegiatan magang yang dilaksanakan kurang lebih selama enam bulan, pada bulan pertama, aktivitas yang dilakukan meliputi pengenalan lingkungan perusahaan, instalasi alat atau *tools* untuk mendukung proses pengembangan, serta penyelesaian beberapa tantangan atau tugas dari *Product Lead* berupa perbaikan *bug* atau *error* pada proyek-proyek sebelumnya untuk menguji kompetensi penulis. Dalam mengerjakan tantangan tersebut, penulis berperan sebagai *frontend* yang bertanggung jawab dalam mengerjakan antarmuka pengguna (UI) menggunakan *framework* React.js. Setelah itu, penulis berperan sebagai *fullstack developer* dalam mengerjakan tiga proyek, jadwal pelaksanaan setiap proyek diperlihatkan pada Tabel 3.1.

Selama kegiatan magang, manajemen proyek dan metode pengembangan dilakukan dengan pendekatan *Agile,* Meskipun pada dasarnya perusahaan Varnion telah mengarahkan penggunaan kerangka kerja (*framework*) Scrum untuk diimplementasikan bagi setiap tim atau departemen. Namun, dalam realita pelaksanaannya di divisi *Research and Development* (R&D), terdapat perbedaan terhadap penerapannya khusus bagi penulis sebagai mahasiswa magang. Penulis menggunakan dokumen Spreadsheet pemberian *Product Lead* sebagai alat pelaporan progress tugas, sedangkan tim menggunakan media terpisah yaitu outline.

Tabel 3.1 Jadwal Pelaksanaan Proyek

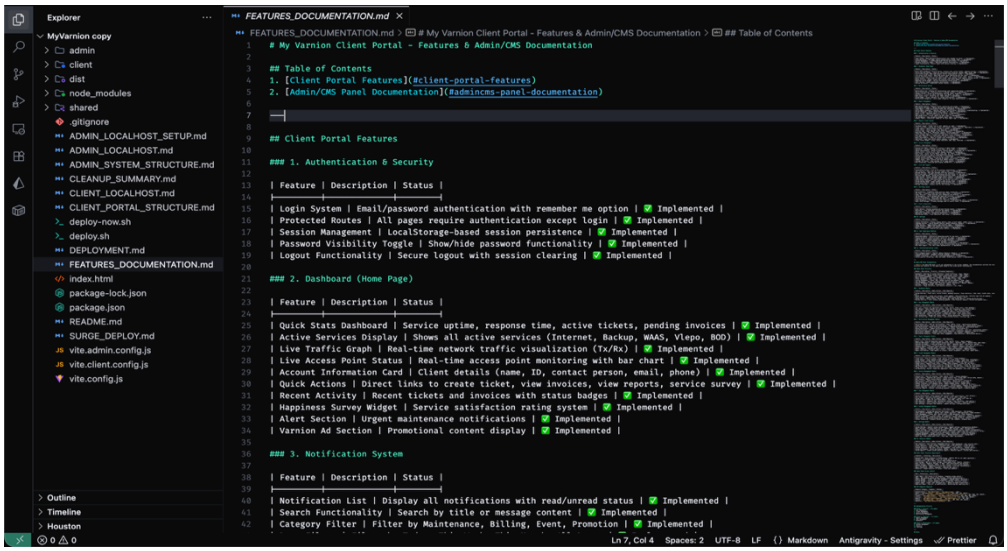
Nama Proyek	Waktu Mulai	Waktu Berakhir
Pixku CMS	25 September 2025	24 November 2025
Arclab	1 Desember 2025	16 Januari 2026
My Varnion	05 Januari 2026	13 Maret 2026

3.2 Proyek My Varnion

My Varnion merupakan aplikasi berbasis web untuk memenuhi kebutuhan internal PT Varnion Technology Semesta dalam mengelola data pelanggan B2B. Aplikasi ini dirancang dengan landasan untuk meningkatkan efisiensi operasional perusahaan. Sebelum aplikasi My Varnion dirancang, sudah ada *prototype* dari aplikasi ini. Namun, dari sisi desain dan komponen atau fitur belum sepenuhnya sesuai dengan alur yang diharapkan, sehingga penulis

melakukan perbaikan desain antarmuka dan menyempurnakan alur bisnis pada fitur yang ada pada aplikasi tersebut.

Proyek My Varnion dilaksanakan dalam kurun waktu kurang lebih dua bulan, dimulai dari bulan Januari sampai dengan Maret. Proses pengembangan dilakukan dengan memanfaatkan serta mengembangkan kembali kode sumber (*source code*) dasar My Varnion yang bersumber dari berkas kompresi (.zip) pemberian *Product Lead.* Struktur dan dokumentasi kode dari sumber awal yang harus diimplementasikan pada proyek ini diperlihatkan pada Gambar 3.1.



Gambar 3.1 *Source Code* My Varnion dari *Product Lead*

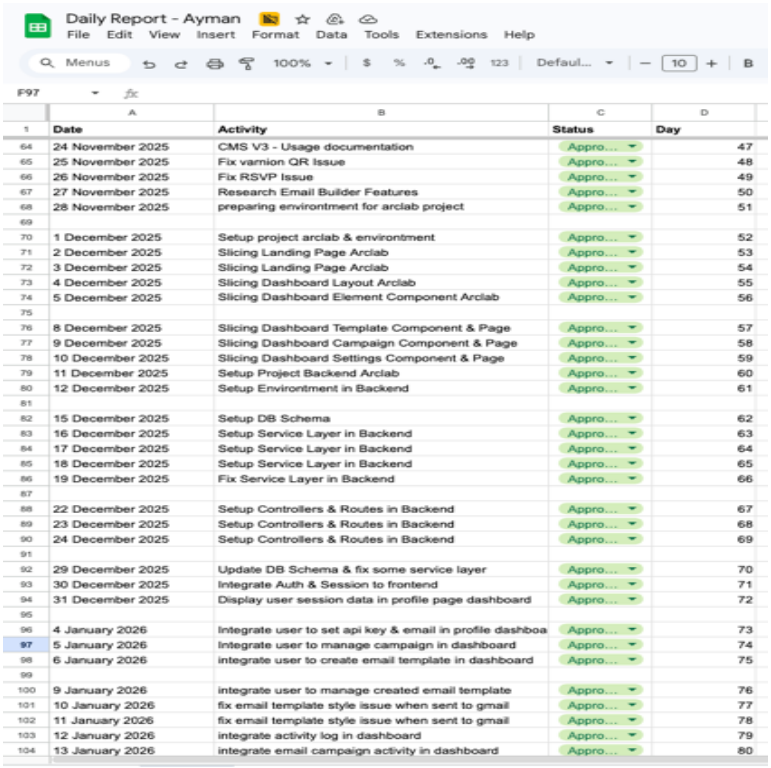
Pengembangan proyek ini dilakukan dengan menggunakan metode pendekatan *Agile* secara iteratif agar penulis selaku pengembang dapat melaporkan progres pengembangan pada sebuah fitur secara cepat pada *Product Lead* sebelum melanjutkan ke fitur selanjutnya. Proyek ini dimulai dengan pengembangan pada lapisan presentasi (*presentation layer*), yaitu menerapkan desain fitur dan halaman pada aplikasi berdasarkan *source code* dari *Product Lead*, dilanjutkan dengan lapisan logika bisnis (*application layer*), dan lapisan akses data (*data access layer*).

3.3 *Tools* dalam proses Pengembangan My Varnion

Selama mengikuti kegiatan magang dan ketika mengerjakan proyek My Varnion, terdapat berbagai perangkat lunak (*tools*) yang digunakan baik untuk mendukung proses pengembangan sistem maupun manajemen tugas penulis. Semua kegiatan pengembangan dilakukan dengan memanfaatkan beberapa perangkat berikut:

a. Google Spreadsheet

Dalam pengembangan proyek My Varnion, Google Spreadsheet dimanfaatkan sebagai alat manajemen dan pelacakan tugas. Penulis mendokumentasikan daftar tugas harian atau *daily task* yang akan dikerjakan pada hari kerja berdasarkan *template* yang telah disediakan oleh *Product Lead* seperti yang ditampilkan pada Gambar 3.2. Tugas harian tersebut berupa fitur yang akan dikerjakan dan ditentukan oleh penulis sendiri, kemudian penulis akan memperbarui kolom status secara *real-time* apabila fitur yang dikerjakan telah selesai.


Daily Report - Ayman

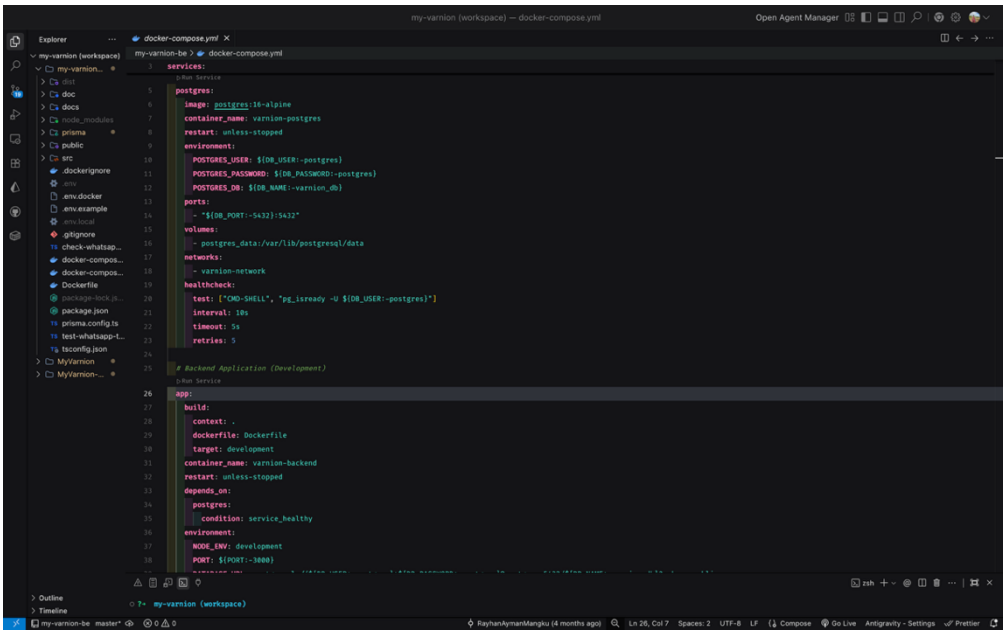
File Edit View Insert Format Data Tools Extensions Help

	Date	Activity	Status	Day
64	24 November 2025	CMS V3 - Usage documentation	Appro...	47
65	25 November 2025	Fix varnion QR Issue	Appro...	48
66	26 November 2025	Fix RSVP Issue	Appro...	49
67	27 November 2025	Research Email Builder Features	Appro...	50
68	28 November 2025	preparing environtment for arclab project	Appro...	51
69				
70	1 December 2025	Setup project arclab & environtment	Appro...	52
71	2 December 2025	Slicing Landing Page Arclab	Appro...	53
72	3 December 2025	Slicing Landing Page Arclab	Appro...	54
73	4 December 2025	Slicing Dashboard Layout Arclab	Appro...	55
74	5 December 2025	Slicing Dashboard Element Component Arclab	Appro...	56
75				
76	8 December 2025	Slicing Dashboard Template Component & Page	Appro...	57
77	9 December 2025	Slicing Dashboard Campaign Component & Page	Appro...	58
78	10 December 2025	Slicing Dashboard Settings Component & Page	Appro...	59
79	11 December 2025	Setup Project Backend Arclab	Appro...	60
80	12 December 2025	Setup Environtment in Backend	Appro...	61
81				
82	15 December 2025	Setup DB Schema	Appro...	62
83	16 December 2025	Setup Service Layer in Backend	Appro...	63
84	17 December 2025	Setup Service Layer in Backend	Appro...	64
85	18 December 2025	Setup Service Layer in Backend	Appro...	65
86	19 December 2025	Fix Service Layer in Backend	Appro...	66
87				
88	22 December 2025	Setup Controllers & Routes in Backend	Appro...	67
89	23 December 2025	Setup Controllers & Routes in Backend	Appro...	68
90	24 December 2025	Setup Controllers & Routes in Backend	Appro...	69
91				
92	29 December 2025	Update DB Schema & fix some service layer	Appro...	70
93	30 December 2025	Integrate Auth & Session to frontend	Appro...	71
94	31 December 2025	Display user session data in profile page dashboard	Appro...	72
95				
96	4 January 2026	Integrate user to set api key & email in profile dashboa	Appro...	73
97	5 January 2026	Integrate user to manage campaign in dashboard	Appro...	74
98	6 January 2026	integrate user to create email template in dashboard	Appro...	75
99				
100	9 January 2026	integrate user to manage created email template	Appro...	76
101	10 January 2026	fix email template style issue when sent to gmail	Appro...	77
102	11 January 2026	fix email template style issue when sent to gmail	Appro...	78
103	12 January 2026	integrate activity log in dashboard	Appro...	79
104	13 January 2026	integrate email campaign activity in dashboard	Appro...	80

Gambar 3.2 Tampilan Spreadsheet

b. Antigravity

Antigravity merupakan *Integrated Development Environment* (IDE) dan juga sebagai *tools source-code editor* yang diluncurkan langsung oleh Google dan berbasis agen AI (*AI Agent*). Editor ini sama seperti IDE pada umumnya seperti VS Code karena editor ini mengadopsi basis kode dari VS Code, sehingga antarmuka atau tampilan, navigasi dan pintasan, serta ekosistem bahasa dan ekstensi yang sebagian besar dapat di unduh dan berfungsi normal di Antigravity. Pada proyek ini, Antigravity digunakan untuk mengerjakan tiap lapisan (*layer*) dari proyek My Varnion seperti lapisan antarmuka hingga basis data. Di sisi lain, dengan menggunakan Antigravity, penulis sangat terbantu dalam proses pengembangan karena didukung oleh *AI Agent* yang dapat membantu mengefisiensi waktu proses pengembangan proyek. Selain itu, Antigravity juga digunakan untuk membuat file konfigurasi Docker yang bernama "`docker-compose.yml`". Gambar 3.3 merupakan visualisasi dari file "`docker-compose.yml`" pada Antigravity.

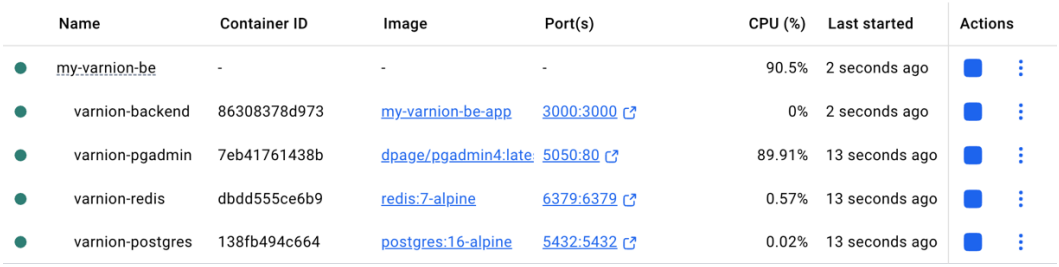


Gambar 3.3 Tampilan Antigravity

c. Docker Compose

Docker Compose merupakan *tools* yang disediakan oleh Docker yang berfungsi untuk mendefinisikan dan menjalankan aplikasi secara *multi-container*. Setiap layanan yang dikemas memiliki konfigurasi dependensi masing-masing, sehingga memungkinkan setiap layanan dapat berjalan secara mandiri di dalam lingkungan yang terisolasi.

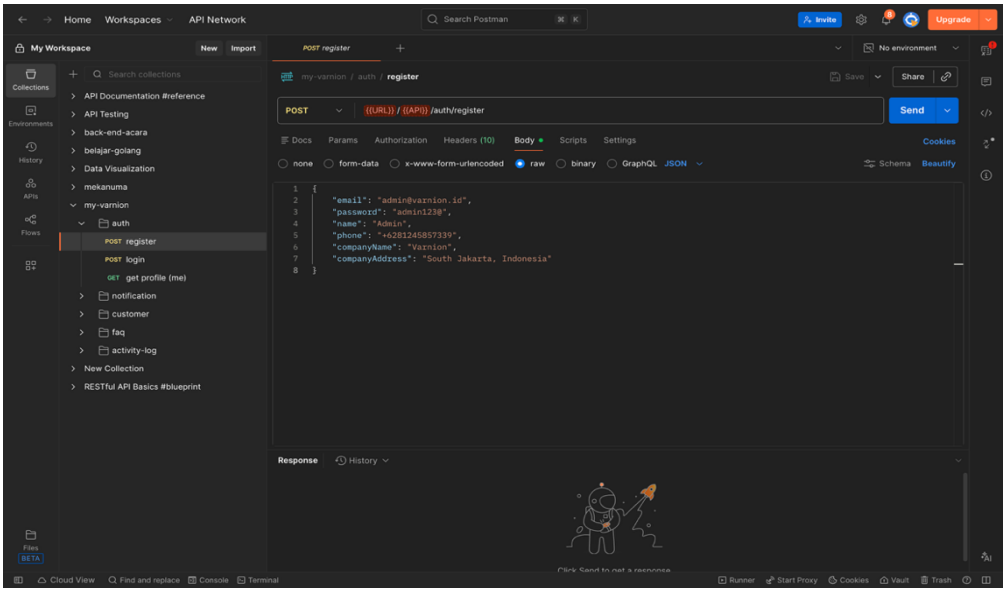
Dalam proyek ini, seluruh *container* pada Docker dapat berjalan melalui terminal IDE hanya dengan menjalankan satu perintah, yaitu "`npm run docker:dev`". Di balik layar, perintah tersebut akan mengeksekusi instruksi "`docker-compose up -d`". Kumpulan dari kontainer yang sedang berjalan dapat dilihat pada Gambar 3.4.



Gambar 3.4 Tampilan Docker Desktop *Container Group* My Varnion

d. Postman

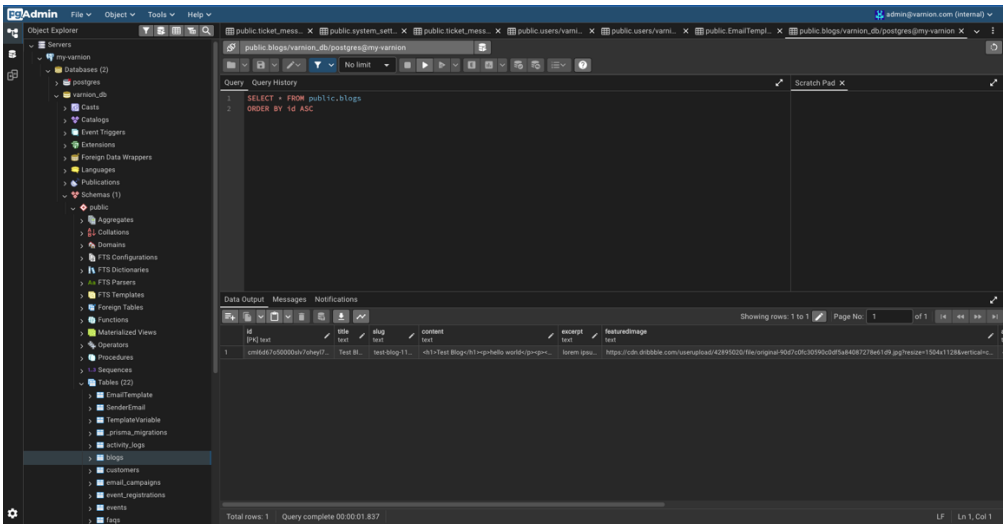
Postman merupakan *tools* yang secara umum digunakan oleh pengembang untuk menguji setiap *endpoint* API yang dihasilkan dari *backend,* tepatnya pada lapisan logika bisnis (*application layer*). Pada proyek My Varnion, Postman digunakan untuk menguji kumpulan endpoint yang telah disusun di dalam sebuah *collections,* seperti *collections auth, notification, customer, faq, activity-log* dan lainnya, seperti yang diperlihatkan pada Gambar 3.5 merupakan tampilan Postman yang sedang menunjukkan proses pengujian pada *endpoint auth* menggunakan *method POST* untuk menguji registrasi pengguna.



Gambar 3.5 Tampilan Postman

e. pgAdmin

gAdmin merupakan perangkat manajemen basis data berbasis web dan grafis (GUI) yang digunakan untuk mengelola *database* PostgreSQL. Penggunaan pgAdmin dalam proyek My Varnion tidak dijalankan secara lokal melalui aplikasi pgAdmin, melainkan dijalankan melalui browser pada url "`http://localhost:5050`" karena sudah dikemas ke dalam *container* yang terisolasi dan memiliki semua dependensi dan konfigurasi yang siap pakai, sehingga untuk menjalankannya cukup dengan menjalankan perintah seperti yang telah disebutkan pada poin pembahasan terkait Docker Compose untuk menjalankan seluruh *container.*



Gambar 3.6 Tampilan pgAdmin

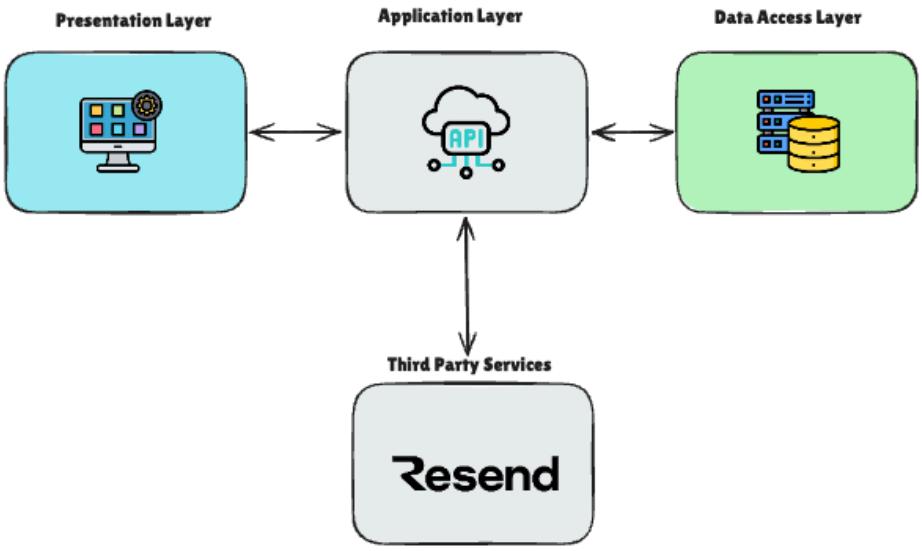
3.4 Pengembangan Proyek My Varnion

Berikut merupakan hasil dari pengembangan proyek My Varnion yang telah dilakukan pada PT. Varnion Technology Semesta. Pengembangan proyek terhadap tiga lapisan utama aplikasi mendapatkan beberapa hasil seperti berikut:

3.4.1 Perancangan Sistem

Tahap perancangan sistem ini dilaksanakan dalam kurun waktu kurang lebih dua bulan, terhitung mulai tanggal 05 Januari sampai dengan 13 Maret 2026. Tahap ini merupakan langkah yang sangat penting untuk memetakan arsitektur dan fungsionalitas aplikasi sebelum masuk ke tahap implementasi kode program. Perancangan ini difokuskan pada penyusunan alur kerja pengelolaan pelanggan B2B serta mekanisme integrasi fitur *email template builder*

berbasis web. Alur integrasi antara modul internal aplikasi dengan komponen *email template builder* digambarkan melalui arsitektur sistem pada Gambar 3.7.



Gambar 3.7 Arsitektur Sistem

Perancangan arsitektur sistem pada aplikasi My Varnion mengadopsi pola arsitektur tiga lapis (*three-tier architecture*). Pemisahan kepada tiga *layer* ini dilakukan untuk menjaga keamanan data, mempermudah pemeliharaan kode, serta memisahkan integrasi layanan pihak ketiga. Berikut merupakan penjelasan dan hubungan dari setiap *layer* seperti yang ditampilkan pada Gambar 3.7:

A. Presentation Layer

Lapisan atau *layer* ini merupakan lapisan antarmuka (*frontend*) berbasis web yang berinteraksi langsung dengan pengguna, lapisan ini dibuat ke dalam dua repositori kode terpisah, yaitu `my-varnion-admin` yang memuat *dashboard* utama serta fitur *email template builder* bagi admin internal perusahaan, dan `my-varnion-client` sebagai panel khusus untuk pelanggan B2B. Kedua aplikasi ini berinteraksi dengan server melalui REST API yang disediakan dari sisi *backend*. Lapisan ini bertanggung jawab untuk menampilkan komponen visual aplikasi, menyediakan fitur editor *email template builder* dan mengambil data dari interaksi pengguna untuk diteruskan ke lapisan berikutnya melalui protokol HTTP/HTTPS.

B. Application Layer

Lapisan ini bertindak sebagai pusat kendali dan tempat untuk memproses logika bisnis utama aplikasi (*backend*) yang diimplementasikan pada folder `my-varnion-be`. Lapisan ini menyediakan sekumpulan *endpoint* REST API internal yang akan dikonsumsi oleh *Presentation Layer*. Selain itu, lapisan ini berfungsi sebagai jembatan yang memproses permintaan dari *Presentation Layer,* melakukan validasi, mengolah logika pengelolaan pelanggan B2B, memproses data dari *template builder* menjadi dokumen HTML yang aman dan siap kirim, serta mengatur komunikasi dengan *Data Access Layer* melalui REST API internal dan *Third-Party Services* dengan REST API eksternal.

C. Data Access Layer

Merupakan lapisan penyimpanan data yang menggunakan manajemen basis data relasional (RDBMS) PostgreSQL. Untuk mendukung replikasi lingkungan pengembangan dan konsistensi variabel lingkungan (*environment variables*), basis data ini dijalankan di dalam *container* Docker. Komunikasi dan manipulasi data antara *Application Layer* ke lapisan basis data ini dilakukan secara aman menggunakan Prisma sebagai *Object-Relational Mapping* (ORM), sehingga mengurangi penulisan *query* SQL secara manual dan menjaga konsistensi tipe data.

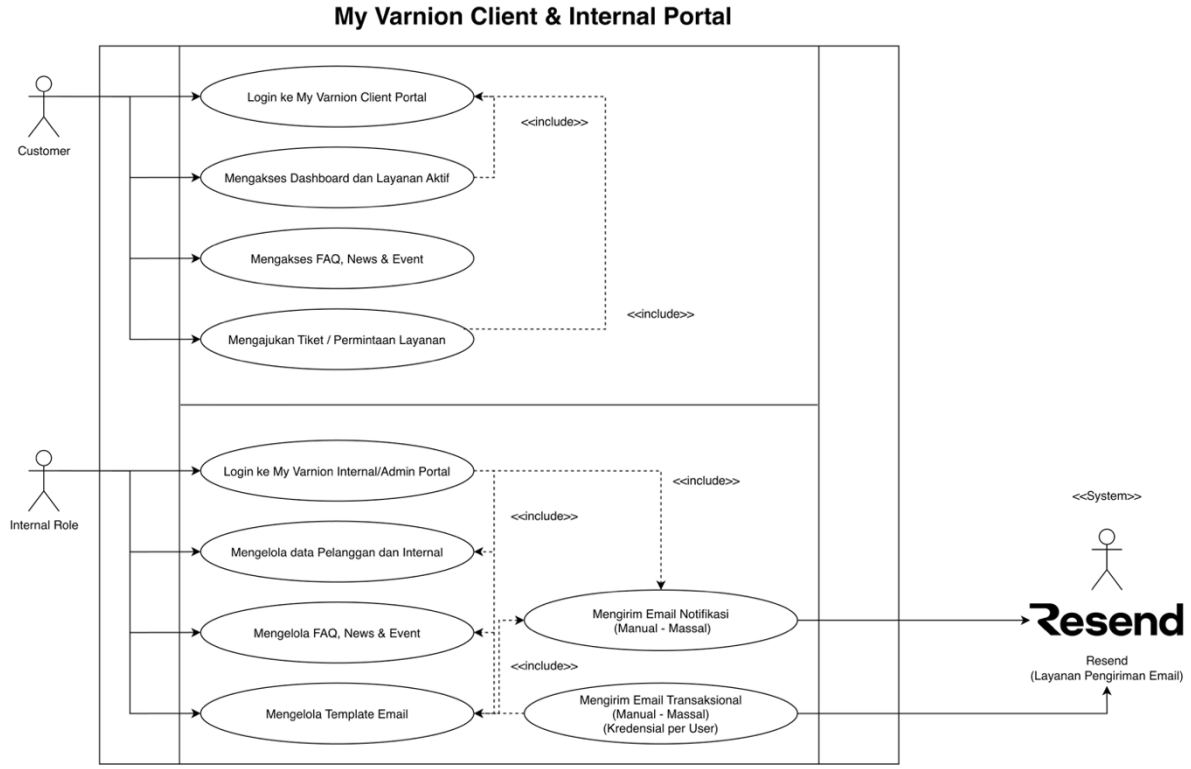
D. Third Party Services

Merupakan layanan infrastruktur eksternal di luar dari tiga *layer* pada arsitektur inti yang terintegrasi melalui REST API layanan Resend. Layanan ini, dimanfaatkan secara khusus untuk menangani proses pengiriman email kepada pelanggan B2B. Komunikasi layanan ini dikendalikan oleh *Application Layer* untuk menjaga keamanan kredensial seperti *API key*, dan konfigurasi *email sender*.

Secara keseluruhan, komunikasi antar komponen dalam arsitektur ini berbasis pada arsitektur layanan web. Lapisan *Presentation Layer* (`my-varnion-client` dan `my-varnion-admin`) berinteraksi langsung dengan *Application Layer* (`my-varnion-be`) melalui REST API internal untuk melakukan seluruh operasi *create, read, update,* dan *delete* (CRUD) data pelanggan B2B dan menyimpan *template.* Selanjutnya apabila pengguna internal atau admin memicu pengiriman email, *Application Layer* akan melakukan pemanggilan REST API

eksternal yang disediakan oleh Resend (*Third-Party Services*) dengan melampirkan payload HTML yang siap di kirim kepada penerima (pelanggan).

Setelah merancang struktur arsitektur dari aplikasi My Varnion, Langkah berikutnya adalah menjabarkan fungsionalitas sistem berdasarkan interaksi pengguna. Rancangan fungsionalitas pada aplikasi ini digambarkan melalui *Use Case Diagram* untuk mendefinisikan aksi dari setiap aktor yang terlibat di dalam sistem, baik dari pihak internal perusahaan (admin) maupun pihak eksternal (pelanggan B2B). Visualisasi dari *Use Case Diagram* aplikasi ini diperlihatkan pada Gambar 3.8.



Gambar 3.8 *Use Case Diagram*

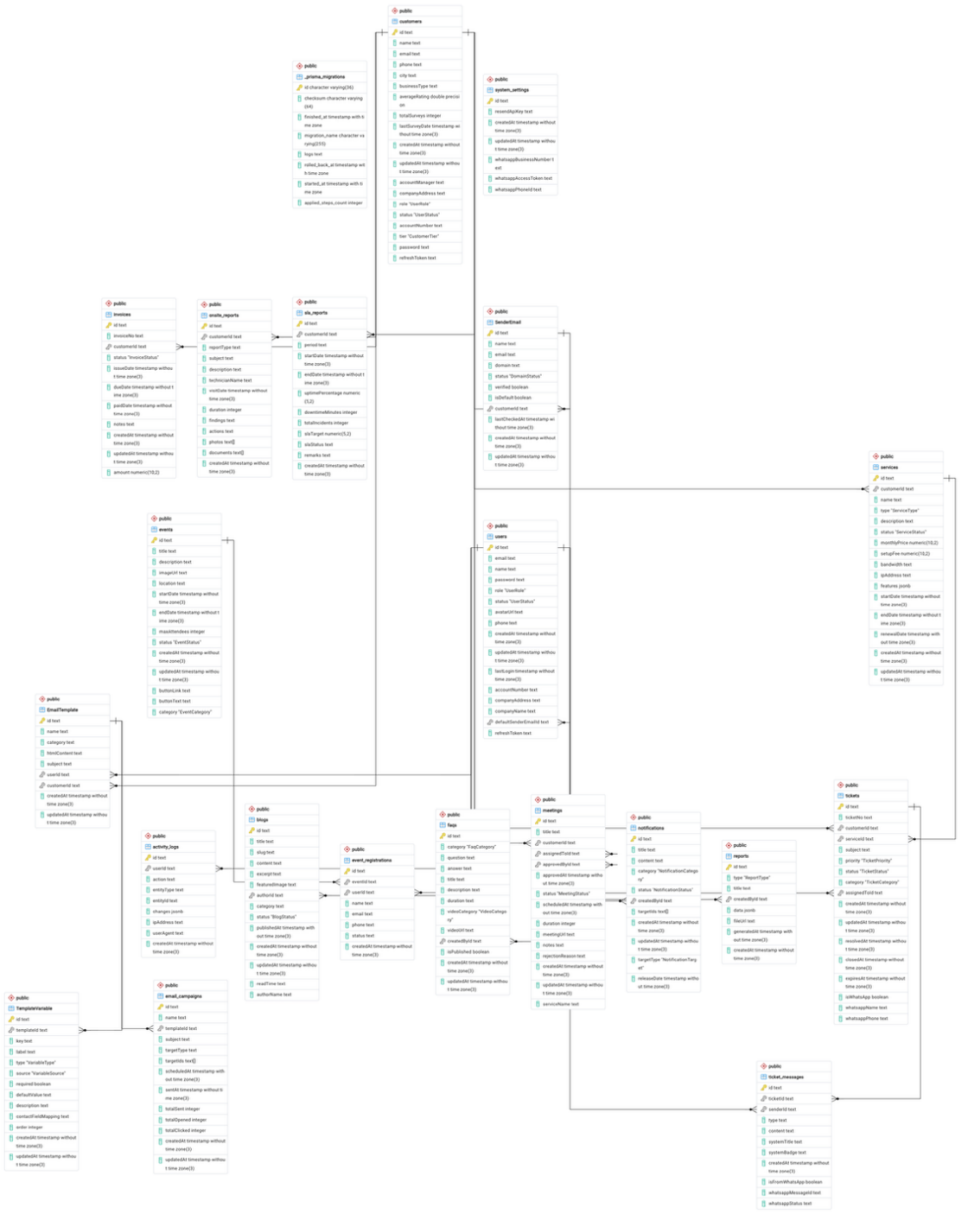
Berdasarkan Gambar 3.8, dapat dilihat bahwa terdapat tiga aktor utama yang berinteraksi di dalam sistem, yaitu internal perusahaan (admin), pelanggan B2B, dan Resend sebagai sistem eksternal. Penjelasan mengenai fungsionalitas dari setiap *use case* utama yang wajib melalui proses autentikasi (*include login*), detail deskripsi fungsionalitas sebagaimana dijelaskan pada Tabel 3.1.

Tabel 3.2 Deskripsi *Use Case*

No	Use Case / Fitur	Aktor	Deskripsi
1	*Login*	Internal, Pelanggan B2B	Proses auntetikasi pengguna untuk masuk ke dalam sistem masing-masing sesuai dengan kredensial pengguna yang terdaftar.
2	Mengelola Pelanggan B2B	Internal	Admin internal melakukan manajemen data pelanggan dari perusahaan mitra seperti proses CRUD melalui panel admin serta dapat merespons tiket yang telah diajukan oleh pelanggan.
3	Mengelola FAQ, *News & Event*	Internal	Admin internal mengelola FAQ, *News & Event* yang akan ditampilkan di panel pelanggan melalui panel admin.
4	Mengelola *Email Template*	Internal	Admin internal mengelola *email template* seperti proses CRUD untuk *template* melalui panel admin.
5	Mengirim Email Notifikasi atau Transaksional	Internal, Resend	Admin internal mengeksekusi pengiriman email menggunakan *template* yang dipilih kepada pelanggan baik itu secara tunggal maupun secara massal.
6	Melihat *Dashboard* & Layanan Aktif	Pelanggan	Pelanggan dapat memantau data layanan dan profil melalui panel pelanggan.
7	Melihat FAQ, *News & Event*	Pelanggan	Pelanggan dapat melihat informasi FAQ, *News & Event* yang telah dibuat sebelumnya oleh pengguna internal melalui panel internal pada panel pelanggan.
8	Membuat Tiket / Permintaan Layanan	Pelanggan	Pelanggan dapat membuat Tiket / Permintaan seperti pengaduan terkait layanan pada panel pelanggan.

3.4.2 Pemodelan Basis Data

Tahap pemodelan basis data diawali dengan penyusunan skema relasi inti sebagai tahap awal yang dilaksanakan pada tanggal 19 Januari sampai dengan 21 Januari. Pemodelan basis data dilakukan untuk menata struktur penyimpanan data transaksional aplikasi secara relasional. Perancangan basis data pada aplikasi My Varnion diimplementasikan melalui *Entity Relationship Diagram* (ERD) dan menggunakan basis data PostgreSQL melalui Prisma ORM. Mengingat pengembangan sistem ini menerapkah pendekatan *Agile*, skema basis data yang dihasilkan pada tahap awal tersebut tidak bersifat final, melainkan berperan sebagai fondasi utama. Selanjutnya seiring dengan dimulainya proses pengembangan dan integrasi antara lapisan logika bisnis (`my-varnion-be`) dengan lapisan presentasi admin (`my-varnion-admin`) serta pelanggan (`my-varnion-client`), struktur basis data ini terus mengalami penyesuaian dan penyempurnaan secara iteratif (berulang) demi mengikuti perubahan alur bisnis aplikasi. Berdasarkan struktur akhir yang telah dirancang, transaksi utama pengelolaan pelanggan B2B didominasi oleh arsitektur relasi *One-to-Many* (1:N). Selain itu, sistem ini juga mengimplementasikan relasi *Many-to-Many* (M:N) dengan memanfaatkan tabel perantara. Visualisasi dari skema ERD dapat dilihat pada Gambar 3.9.



Gambar 3.9 Tampilan *Entity Relationship Diagram (ERD)*

Secara arsitektur, skema ini menerapkan relasi *One-to-Many* sebagai transaksional utama. Model bisnis di aplikasi ini berpusat pada tabel *Customer* yang bertindak sebagai induk bagi berbagai transaksi. Sebagai contoh satu *Customer* dapat mengajukan berbagai komplain melalui tabel *Ticket*, dan Berlangganan lebih dari satu layanan (*Service*). Pola relasi ini berulang hingga ke level yang lebih spesifik, seperti satu *Ticket* yang menampung banyak riwayat percakapan (*TicketMessage*).

Di sisi lain, aplikasi ini juga menggunakan relasi *Many-to-Many* secara eksplisit untuk menangani interaksi antar pengguna yang lebih kompleks, khususnya pada fitur acara (*Event*). Karena satu pengguna (*User*) dapat mendaftar ke banyak acara, dan satu Event dapat diikuti

oleh banyak pengguna, relasi ini tidak dapat menggunakan pola *One-to-Many.* Oleh karena itu, skema ini menggunakan tabel perantara yang bernama *EventRegistration* sebagai jembatan penghubungnya. Penggunaan tabel perantara (*explicit join table*) sangat ideal pada konteks ini karena sistem membutuhkan tempat untuk menampung data spesifik yang hanya terjadi saat momen pendaftaran itu sendiri, seperti nama peserta, nomor telepon, status kehadiran, tanpa harus mengubah tabel induk seperti *User* dan *Event*. Untuk melihat bagaimana relasi ini berinteraksi dengan proses bisnis lainnya, pemetaan secara lengkap dapat dilihat pada Tabel 3.2.

Tabel 3.3 Pemetaan Relasi Basis Data

Tabel Utama	Tabel Terkait	Jenis Relasi	Deskripsi Proses Bisnis
Berpusat pada Pelanggan			
Customer	Ticket	One-to-Many	Satu pelanggan dapat mengajukan banyak tiket komplain atau bantuan.
Customer	Service	One-to-Many	Satu pelanggan dapat berlangganan lebih dari satu layanan/produk.
Customer	Meeting	One-to-Many	Satu pelanggan dapat mengajukan banyak jadwal pertemuan.
Customer	SenderEmail	One-to-Many	Satu pelanggan dapat dikonfigurasikan dengan banyak email pengirim kustom.
Customer	EmailTemplate	One-to-Many	Satu pelanggan dapat memiliki banyak template email spesifik.
Berpusat pada pengguna Internal			
User	Blog	One-to-Many	Satu pengguna internal dapat menulis dan publikasi banyak artikel blog.
User	Ticket	One-to-Many	Satu pengguna internal yang ditugaskan dapat menangani banyak tiket.
User	TicketMessage	One-to-Many	Satu pengguna internal yang ditugaskan dapat mengirim balasan pesan ke pelanggan.

Tabel Utama	Tabel Terkait	Jenis Relasi	Deskripsi Proses Bisnis
User	Meeting	One-to-Many	Satu pengguna internal yang ditugaskan dapat menyetujui banyak jadwal pertemuan yang telah diajukan pelanggan.
User	EmailTemplate	One-to-Many	Satu pengguna internal dapat membuat dan mendesain banyak template email.
User	ActivityLog	One-to-Many	Satu pengguna internal memiliki riwayat aktivitas sistem yang sangat banyak.
User	Notification	One-to-Many	Satu pengguna internal yang ditugaskan dapat menyebarkan notifikasi sistem ke pelanggan.
User	Faq	One-to-Many	Satu pengguna internal dapat menulis banyak FAQ (Frequently Asked Questions) yang akan ditampilkan pada portal pelanggan.
Relasi Operasional & Konfigurasi			
Ticket	TicketMessage	One-to-Many	Satu ruang tiket bertindak sebagai wadah untuk menampung banyak pesan obrolan.
Service	Ticket	One-to-Many	Satu layanan dapat dikaitkan dengan banyak tiket kendala.
SenderEmail	User	One-to-Many	Satu konfigurasi email default dapat digunakan bersama-sama oleh banyak pengguna internal (User).
EmailTemplate	EmailCampaign	One-to-Many	Satu desain template email dapat digunakan berulang kali untuk banyak campaign berbeda.
Tabel Perantara (Explicit Join)			
User & Event	EventRegistration	Many-to-Many	Tabel perantara EventRegistration menghubungkan tabel User dan Event. Memungkinkan banyak User mendaftar ke banyak Event, sekaligus menyimpan data kehadiran secara mandiri.

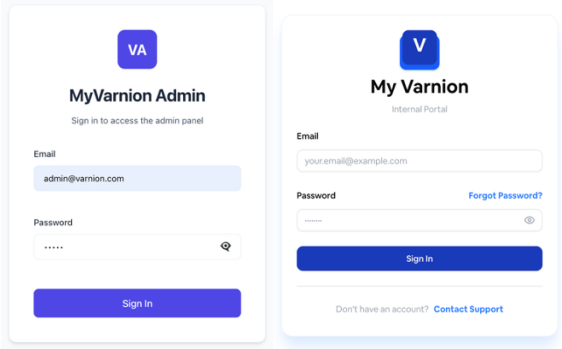
3.4.3 Implementasi Antarmuka Aplikasi Pengelolaan Pelanggan B2B

Tahap ini merupakan visualisasi nyata dari seluruh perancangan fungsional sistem, implementasi antarmuka pada aplikasi My Varnion diwujudkan ke dalam dua repositori *frontend* terpisah agar arsitektur menjadi lebih modular. Kedua dari repositori tersebut yaitu `my-varnion-admin` sebagai portal atau panel kendali bagi pengguna internal dan `my-varnion-client` sebagai portal interaksi bagi pelanggan B2B. Dengan menggunakan pendekatan tersebut, proses pemeliharaan (*maintenance*) dan pembagian kerja terhadap tim di masa mendatang dapat dilakukan dengan lebih mudah tanpa mengganggu ekosistem dari masing-masing platform. Penjelasan mengenai implementasi halaman antarmuka beserta hubungannya dengan pemetaan relasi basis data dijabarkan sebagai berikut:

A. Antarmuka Portal Pengguna Internal (`my-varnion-admin`)

1) Halaman *Login* Internal

Halaman *login* internal merupakan pintu masuk utama bagi staf untuk mengakses portal `my-varnion-admin`. Halaman ini berfungsi sebagai gerbang autentikasi untuk memastikan bahwa hanya pengguna yang memiliki hak akses yang dapat masuk ke sistem dan mengelola data pelanggan B2B. Visualisasi halaman *login* versi sebelumnya dapat dilihat pada Gambar 3.10 (a), sedangkan versi yang telah didesain ulang dapat dilihat pada Gambar 3.10 (b).



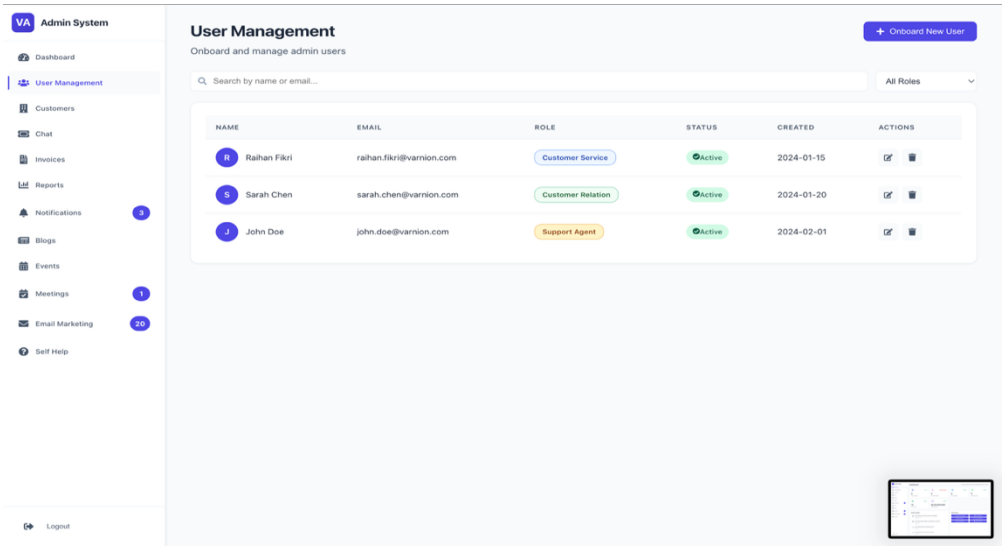
(a) (b)

Gambar 3.10 Antarmuka Halaman Login Internal

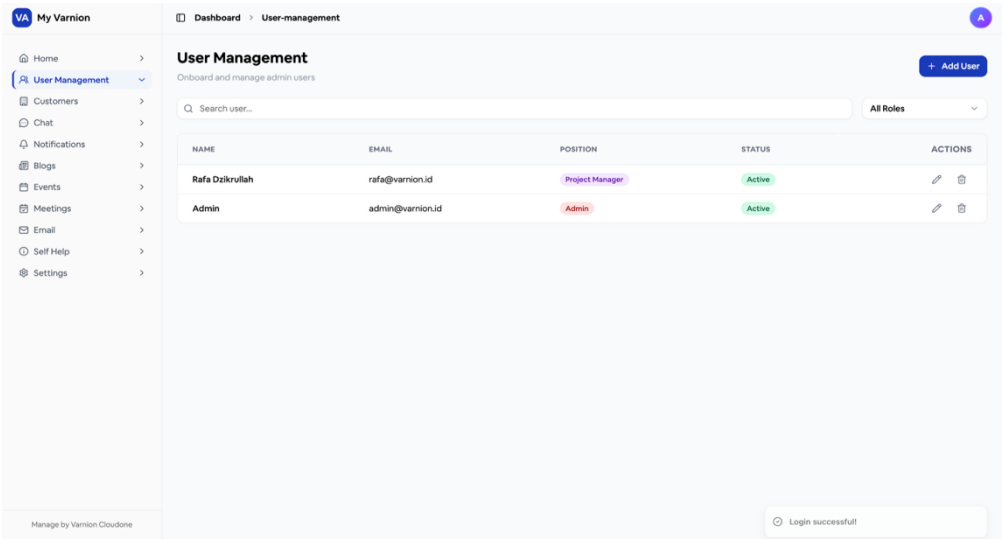
Selain perubahan terhadap desain, juga terdapat integrasi dengan lapisan logika bisnis (*backend*) pada halaman ini.

2) Halaman Manajemen Pengguna Internal

Halaman manajemen pengguna internal merupakan wadah untuk mengelola akun. Halaman ini memungkinkan terjadi aksi-aksi seperti mendaftar, mengubah data, dan menghapus akun internal. Aksi tersebut hanya dapat dilakukan oleh pengguna internal yang memiliki otorisasi dan hak akses. Visualisasi halaman ini untuk versi sebelumnya diperlihatkan pada Gambar 3.11 (a), di mana sistem tidak dapat melakukan aksi untuk mengelola data, sedangkan versi yang telah didesain ulang dan terintegrasi diperlihatkan pada Gambar 3.11 (b) yang memungkinkan staf dapat mengelola data.



(a)

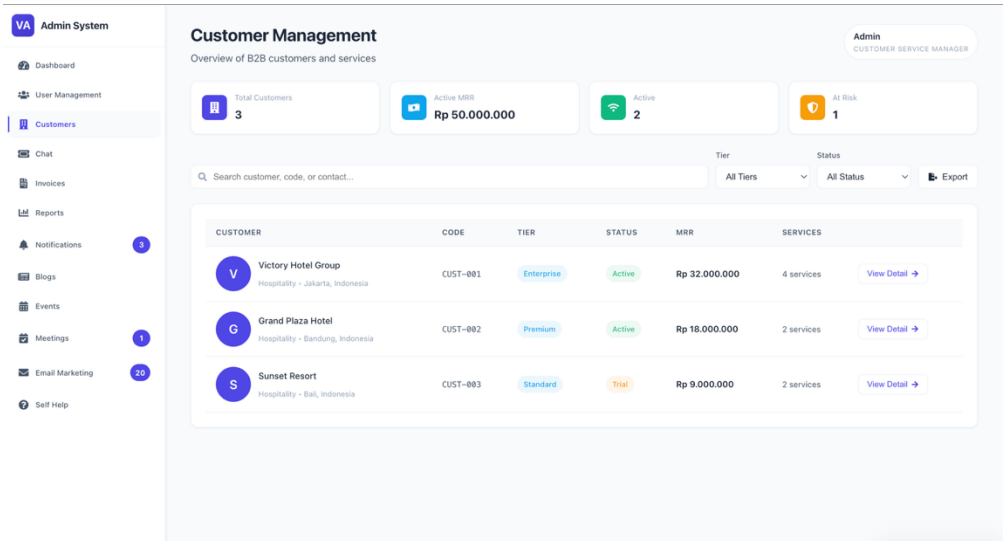


(b)

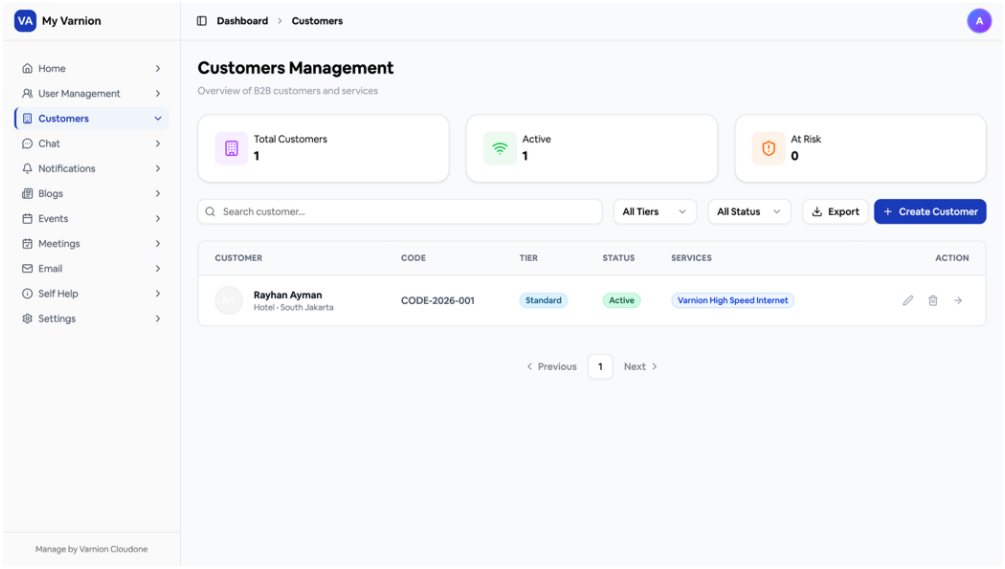
Gambar 3.11 Antarmuka Halaman Manajemen Pengguna Internal

3) Halaman Manajemen Pelanggan

Halaman manajemen pelanggan merupakan salah satu fitur utama pada sistem ini. Pengguna yang telah terautentikasi dan memiliki hak akses dapat melakukan seluruh operasi, seperti mendaftarkan pelanggan baru, mengedit data pelanggan, dan menghapus akun yang sudah tidak aktif berlangganan. Visualisasi halaman manajemen pelanggan versi sebelumnya yang belum terintegrasi diperlihatkan pada Gambar 3.12 (a), sedangkan versi yang telah didesain ulang dan terintegrasi diperlihatkan pada Gambar 3.12 (b).



(a)

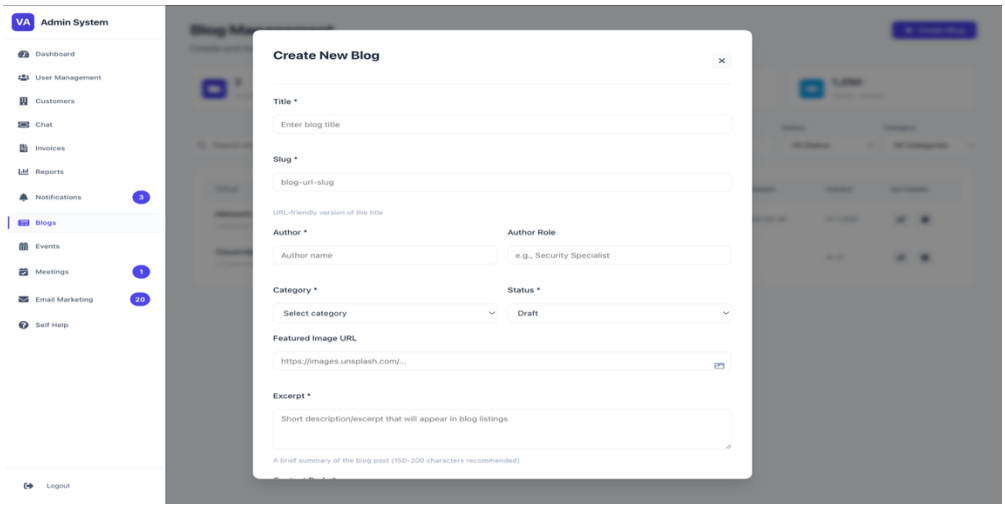


(b)

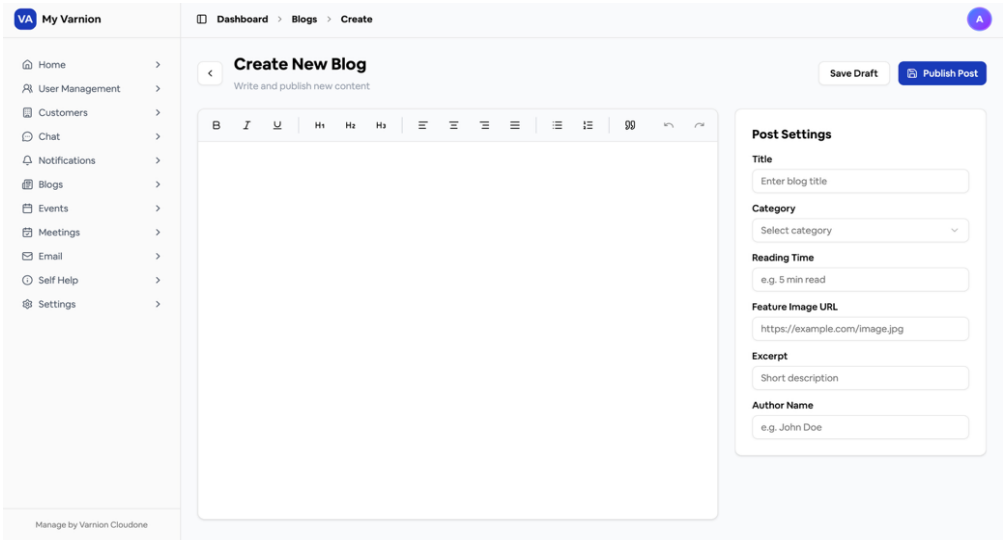
Gambar 3.12 Antarmuka Halaman Manajemen Pelanggan B2B

4) Halaman Manajemen Berita (*News/Blog*)

Halaman ini disediakan bagi pengguna internal atau staf untuk membuat berita atau artikel blog. Artikel yang telah disusun dapat langsung diterbitkan atau disimpan sebagai draf. Artikel dengan status terbit akan muncul pada halaman Informasi Berita dan Acara di portal pelanggan. Tampilan sebelum didesain ulang masih menggunakan kolom masukan terpisah seperti yang diperlihatkan pada Gambar 3.13 (a), sedangkan versi terbaru sudah menggunakan komponen kanvas. Visualisasi fitur editor yang didesain ulang dapat dilihat pada Gambar 3.13 (b) dan telah berfungsi untuk mengelola artikel.



(a)

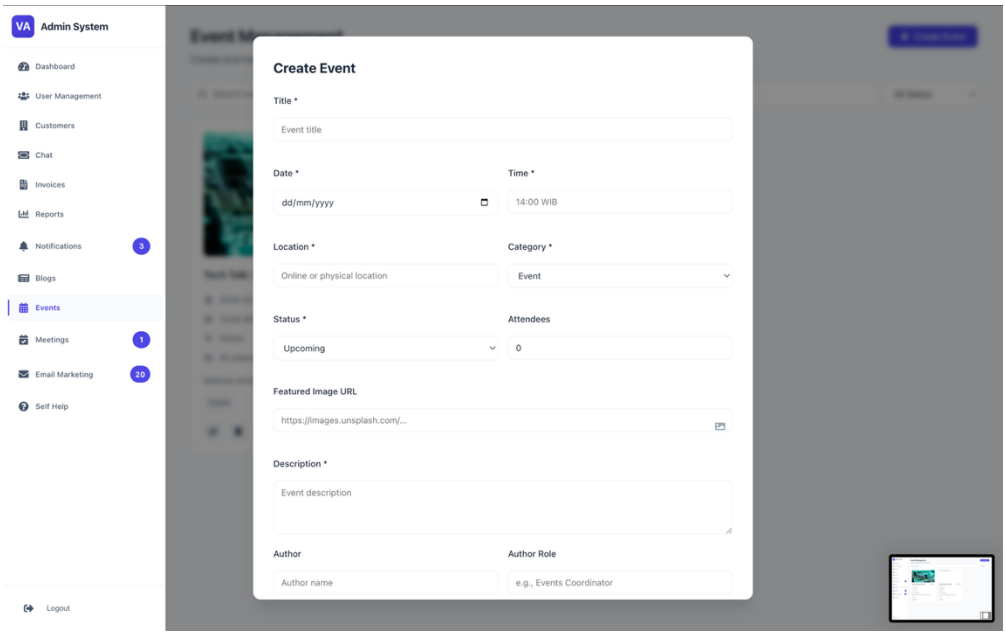


(b)

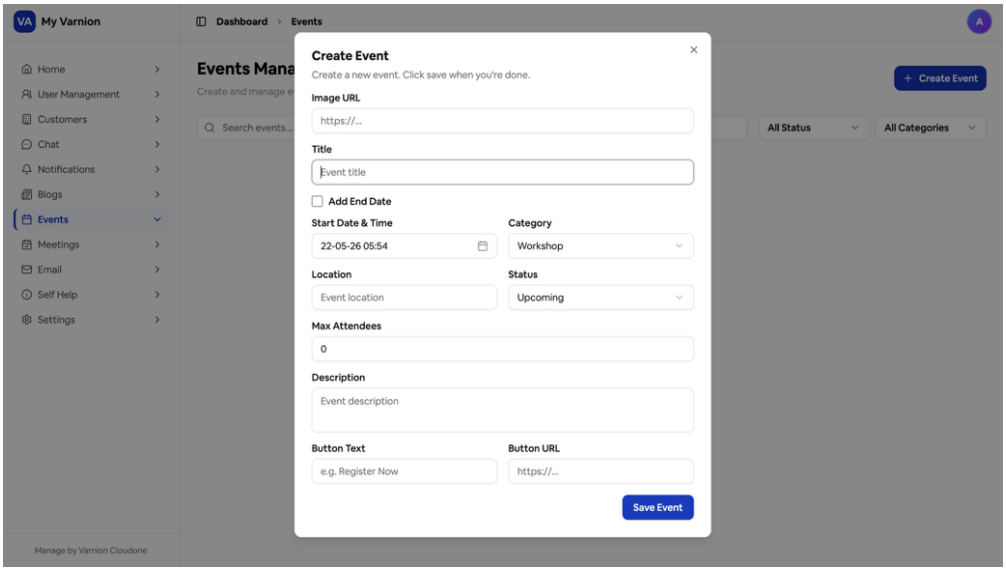
Gambar 3.13 Antarmuka Editor Blog pada Halaman Manajemen Berita

5) Halaman Manajemen Acara (*Event*)

Halaman ini menampilkan komponen *pop up* yang memungkinkan pengguna internal atau staf dapat membuat acara dengan mengisi *url* gambar, judul acara, maksimal partisipan, lokasi acara, serta waktu mulai dan berakhir. Setelah data berhasil disimpan, informasi acara akan otomatis ditampilkan pada halaman informasi berita dan acara di portal pelanggan. Tampilan antarmuka versi sebelumnya di perlihatkan pada Gambar 3.14 (a), di mana masih terdapat beberapa kolom masukan yang kurang relevan, seperti kolom *author* dan *author role* dan belum sepenuhnya berfungsi. Pada versi terbaru seperti yang telah didesain ulang, seperti yang diperlihatkan pada Gambar 3.14 (b), kolom-kolom tersebut sudah disesuaikan agar tetap relevan dan telah berfungsi dalam mengelola data.



(a)

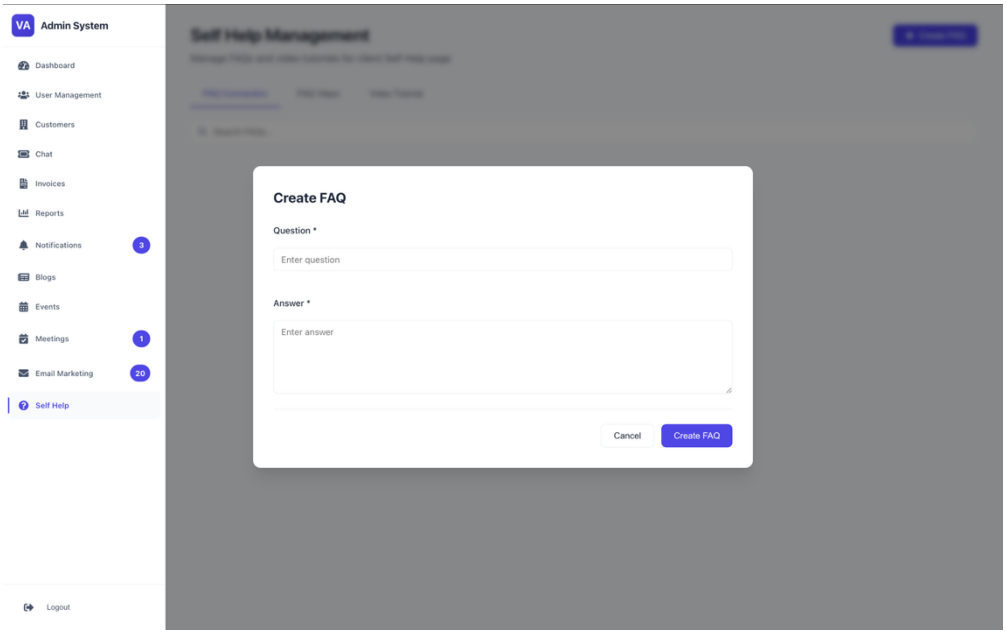


(b)

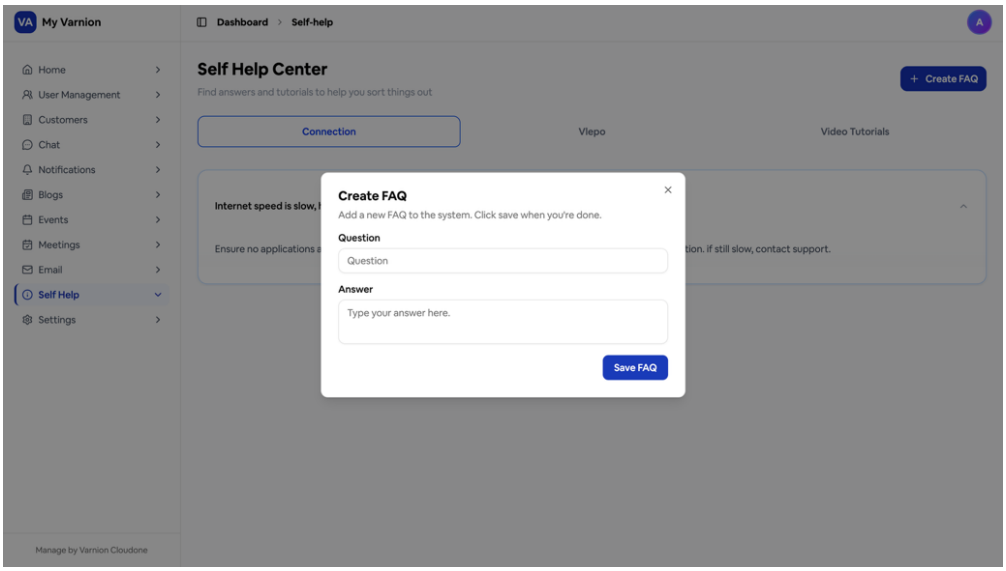
Gambar 3.14 Antarmuka Formulir Tambah Acara pada Halaman Manajemen Acara

6) Halaman Manajemen Pusat Bantuan (FAQ)

Halaman ini menampilkan komponen *pop up* yang memungkinkan pengguna internal untuk membuat *Frequently Asked Questions* (FAQ) bagi pelanggan. FAQ yang telah dibuat akan ditampilkan secara otomatis pada halaman pusat bantuan di portal pelanggan. Visualisasi komponen FAQ versi sebelumnya diperlihatkan pada Gambar 3.15 (a), di mana komponen tersebut belum berfungsi sepenuhnya, sedangkan versi terbaru yang telah didesain ulang diperlihatkan pada Gambar 3.15 (b) dan telah berfungsi dalam mengelola FAQ.



(a)

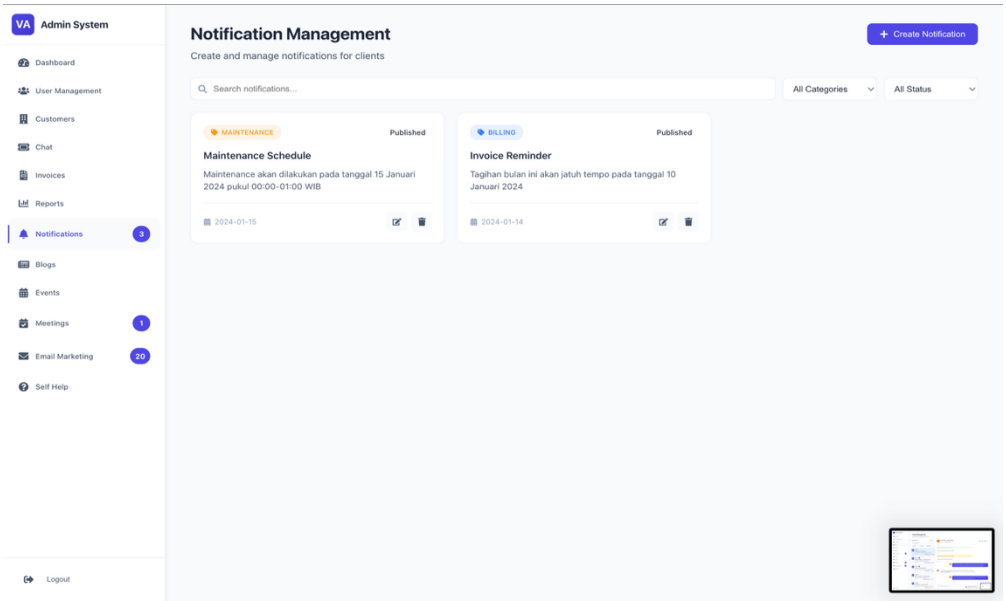


(b)

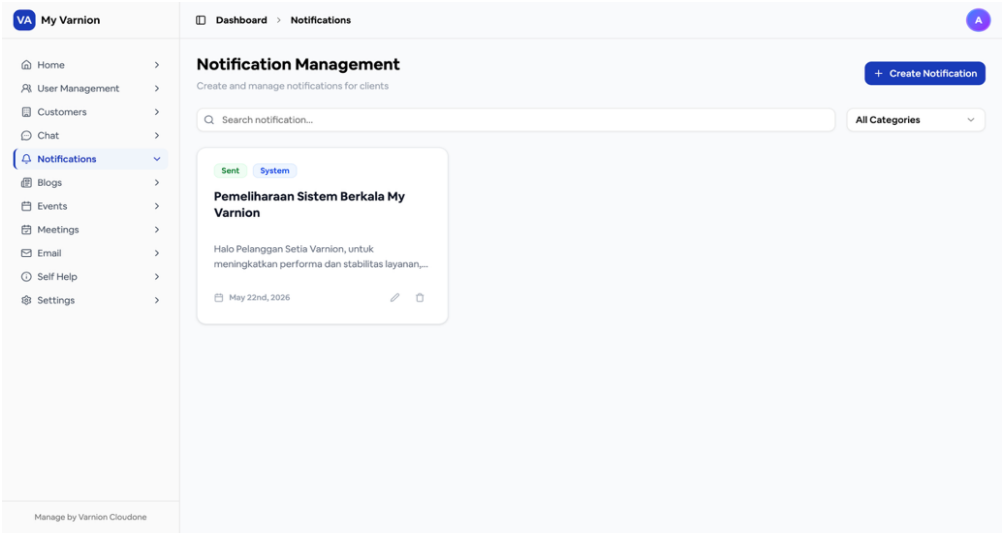
Gambar 3.15 Antarmuka Tambah FAQ pada Halaman Manajemen Pusat Bantuan

7) Halaman Manajemen Notifikasi (*Notification*)

Halaman ini menampilkan daftar notifikasi yang telah dibuat sebelumnya oleh pengguna internal atau staf. Selain itu, terdapat tombol untuk menampilkan *pop up* untuk membuat notifikasi baru. Pengguna dapat mengisi data seperti judul, isi pesan, dan target penerima untuk semua pelanggan maupun spesifik. Visualisasi halaman ini untuk versi sebelumnya diperlihatkan pada Gambar 3.16 (a), sedangkan versi terbaru yang telah didesain ulang diperlihatkan pada Gambar 3.16 (b).



(a)

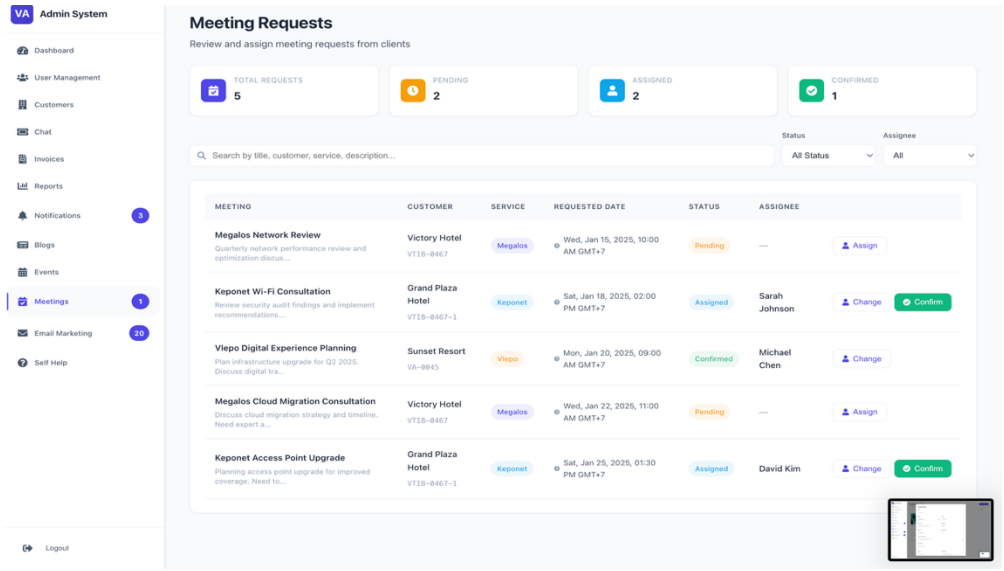


(b)

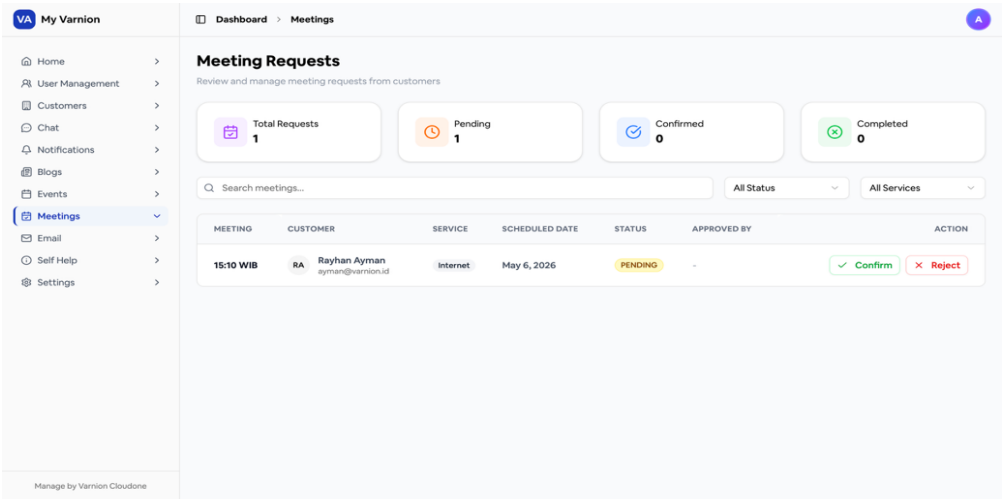
Gambar 3.16 Antarmuka Daftar Notifikasi pada Halaman Manajemen Notifikasi

8) Halaman Manajemen Pengajuan Pertemuan (*Meeting*)

Halaman manajemen pertemuan disediakan bagi pengguna internal untuk mengelola permintaan pertemuan yang diajukan oleh pelanggan B2B. Pada halaman ini, pengguna internal dapat melakukan beberapa aksi, seperti menerima atau menolak permintaan yang masuk melalui portal pelanggan. Visualisasi halaman ini untuk versi sebelumnya diperlihatkan pada Gambar 3.17 (a), di mana data yang tampil merupakan data tiruan dan tidak bisa dikelola. Sementara itu, versi terbaru hasil desain ulang diperlihatkan pada Gambar 3.17 (b) yang telah terintegrasi dan berfungsi dalam mengelola permintaan pertemuan dari pelanggan B2B.



(a)

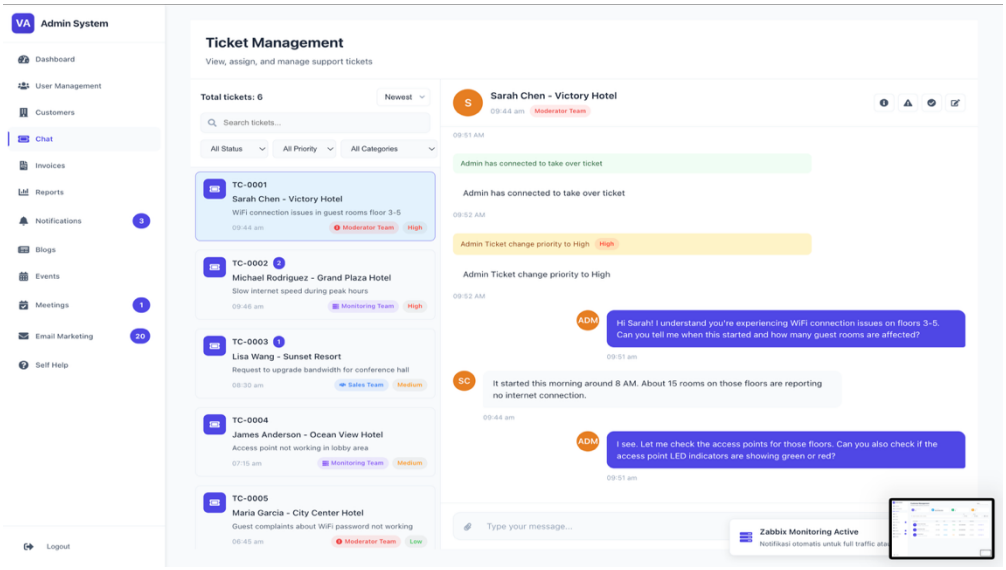


(b)

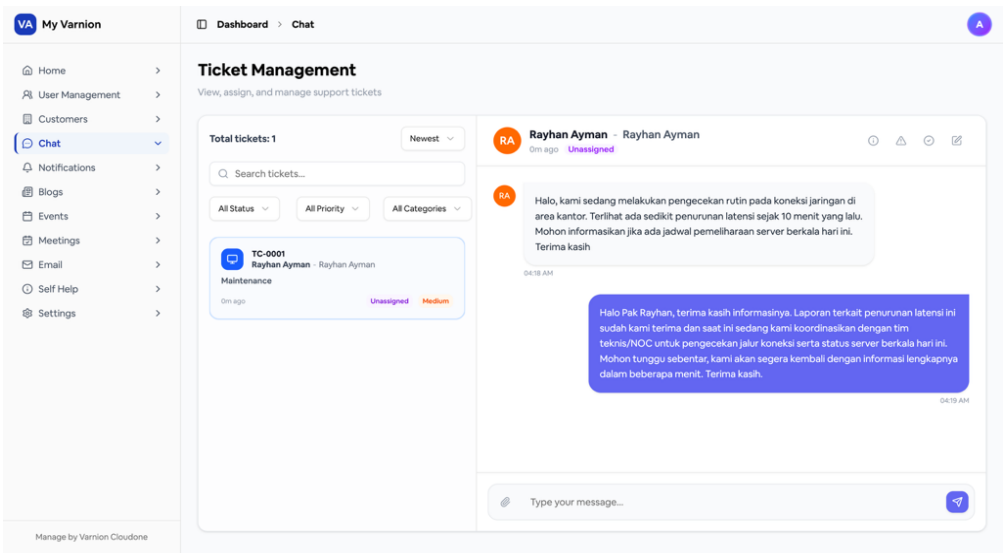
Gambar 3.17 Antarmuka Halaman Manajemen Pertemuan

9) Halaman Manajemen Tiket (*Ticket*)

Halaman ini berfungsi untuk pengguna internal melihat daftar tiket yang diajukan oleh pelanggan B2B, sekaligus memungkinkan pengguna internal untuk memberikan respons pada setiap tiket yang tersedia. Visualisasi mengenai pengelolaan salah satu tiket pada halaman ini untuk versi sebelumnya diperlihatkan pada Gambar 3.18 (a) yang memperlihatkan data yang ditampilkan merupakan data tiruan dan tidak dapat dikelola. Sementara itu, versi terbaru hasil desain ulang diperlihatkan pada Gambar 3.18 (b) yang memungkinkan staf dapat mengelola tiket.



(a)



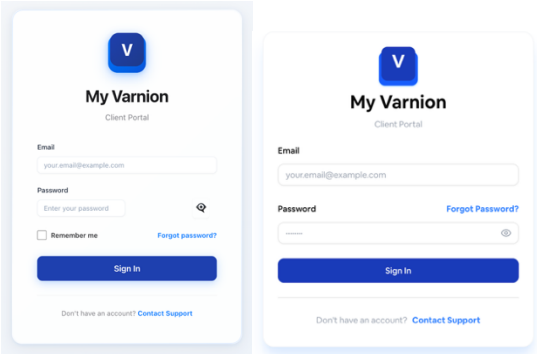
(b)

Gambar 3.18 Antarmuka Halaman Manajemen Tiket

B. Antarmuka Portal Pelanggan (`my-varnion-client`)

1) Halaman *Login* Pelanggan

Halaman *login* pelanggan merupakan pintu masuk utama bagi pelanggan B2B untuk mengakses portal `my-varnion-client`. Halaman ini berfungsi sebagai gerbang autentikasi untuk pelanggan agar dapat masuk ke sistem dan melihat layanan aktif. Visualisasi halaman *login* versi sebelumnya dapat dilihat pada Gambar 3.19 (a), di mana sistem autentikasi belum terintegrasi dengan semestinya. Sementara itu, versi hasil desain ulang yang telah terintegrasi dan berfungsi dengan baik dapat dilihat pada Gambar 3.19 (b).

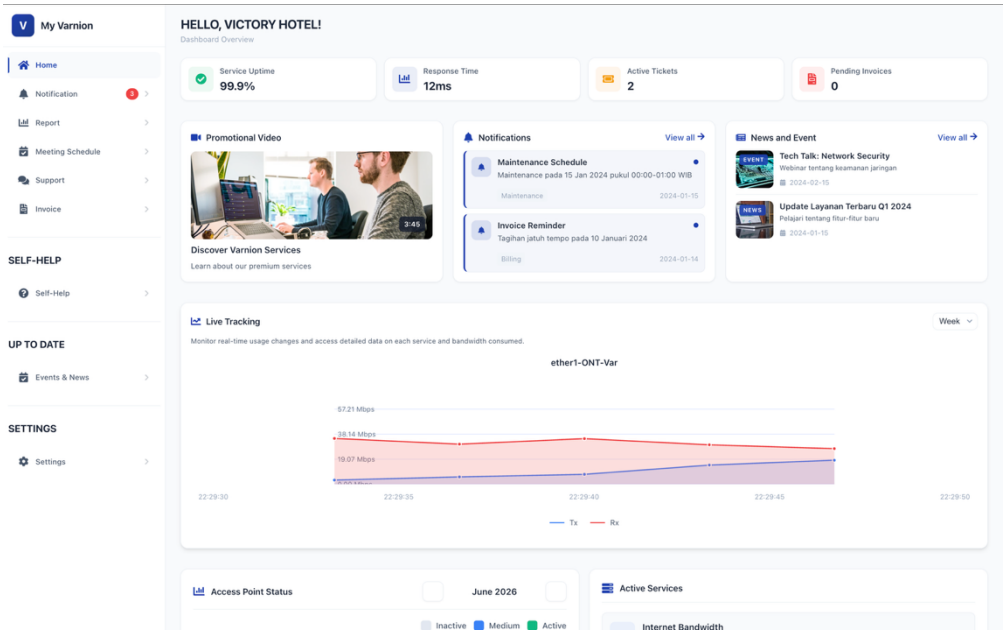


(a) (b)

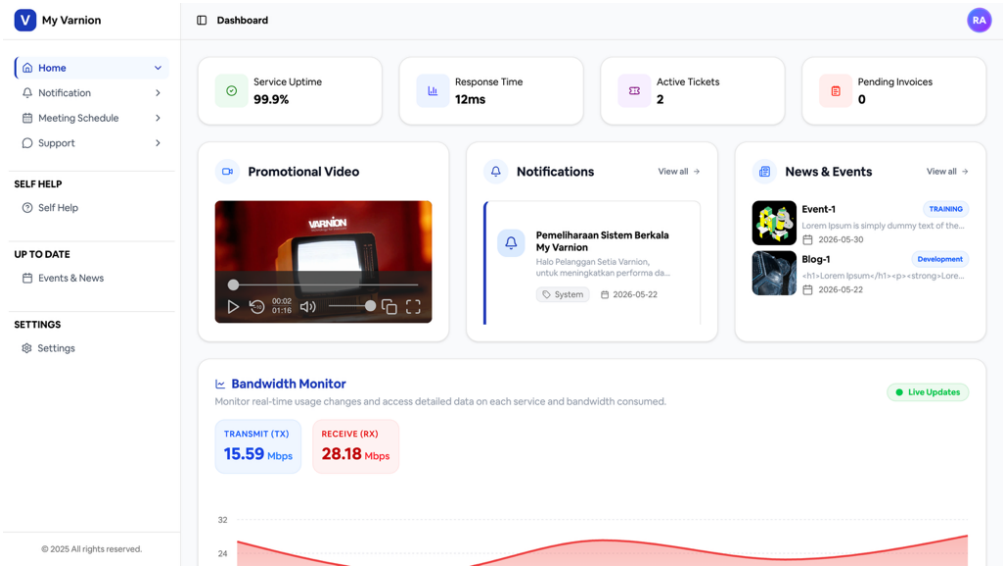
Gambar 3.19 Antarmuka Halaman *Login* Pelanggan B2B

2) Halaman *Dashboard*

Halaman ini merupakan tahap awal setelah pelanggan terauntetikasi, menampilkan informasi terkait layanan, notifikasi, berita dan acara, serta informasi umum lainnya. Visualisasi halaman *dashboard* versi sebelumnya diperlihatkan pada Gambar 3.20 (a), sedangkan versi hasil desain ulang diperlihatkan pada Gambar 3.20 (b).



(a)



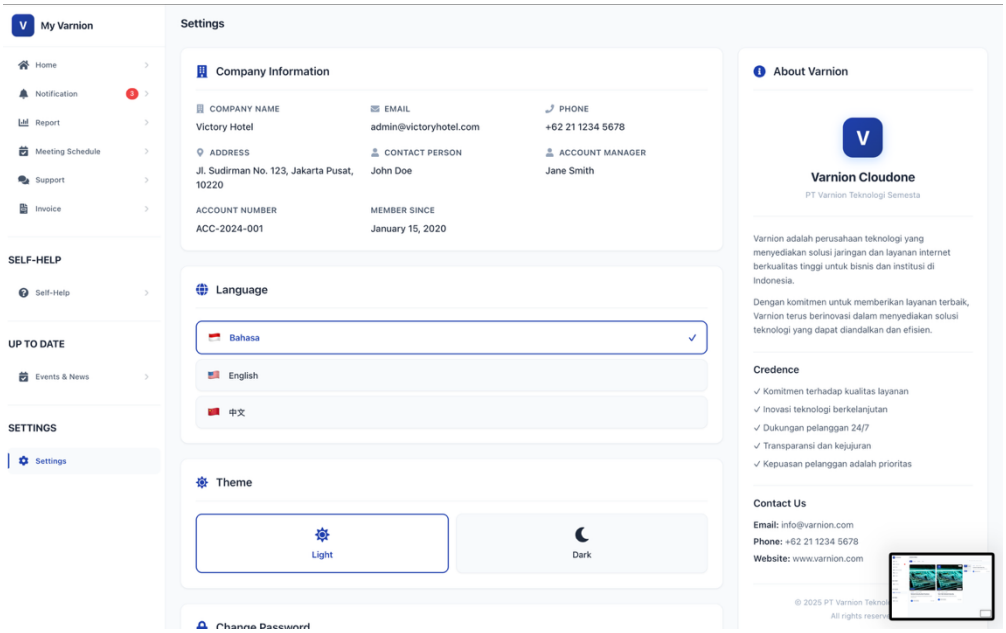
(b)

Gambar 3.20 Antarmuka Halaman *Dashboard* Pelanggan

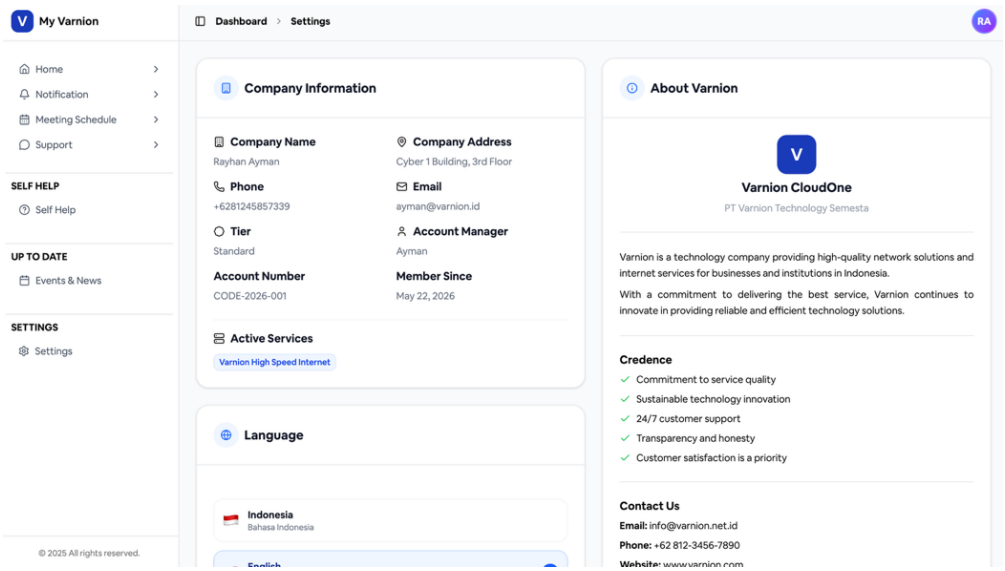
3) Halaman Pengaturan (*settings*)

Halaman pengaturan berfungsi sebagai wadah yang menampilkan profil pelanggan B2B, seperti informasi spesifik terkait akun dan layanan, serta memungkinkan pengguna mengubah konfigurasi preferensi bahasa yang digunakan pada sistem. Visualisasi versi sebelumnya pada halaman ini diperlihatkan pada Gambar 3.21 (a), di mana data yang muncul merupakan data tiruan, sedangkan versi terbaru yang didesain

ulang pada Gambar 3.21 (b) yang menampilkan data pengguna yang telah *login* pada sistem.



(a)

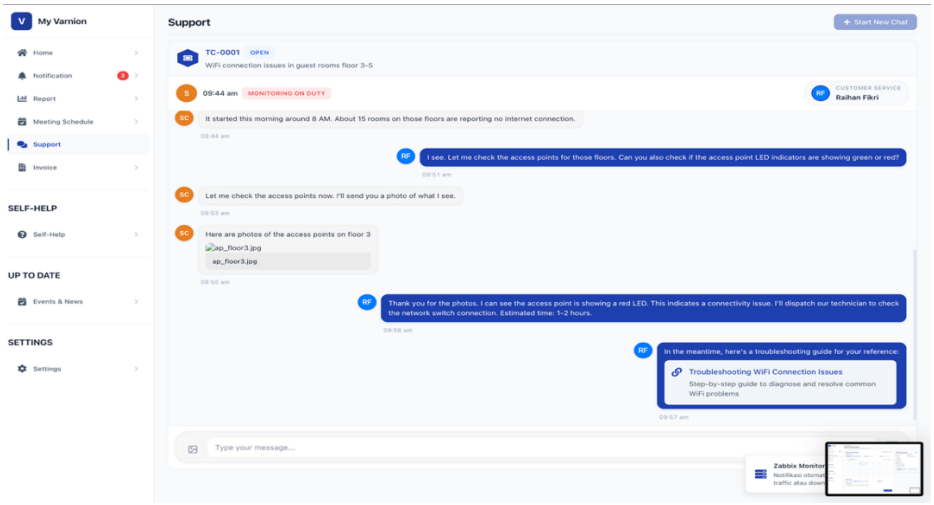


(b)

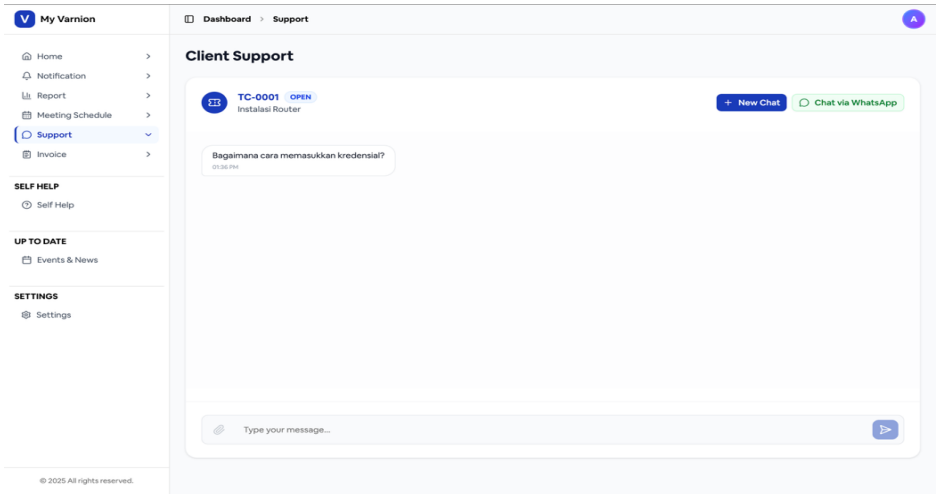
Gambar 3.21 Antarmuka Halaman Pengaturan Pelanggan

4) Halaman Pengajuan Tiket Kendala (*Ticket*)

Halaman ini bertujuan untuk pelanggan B2B melakukan pengajuan terkait keluhan, bantuan teknis, dan layanan dengan menggunakan konsep tiket. Setelah itu, pesan yang diajukan akan direspons oleh staf melalui portal internal. Visualisasi saat proses pengajuan tiket versi sebelumnya diperlihatkan pada Gambar 3.22 (a) di mana proses pengajuan yang dilakukan belum sepenuhnya lengkap dengan alur yang seharusnya, seperti tombol memulai pengajuan belum berfungsi dan langsung menampilkan daftar tiket. Sementara itu, versi hasil desain ulang diperlihatkan pada Gambar 3.22 (b) yang telah ditata ulang dan terintegrasi antara portal pelanggan dan internal.



(a)

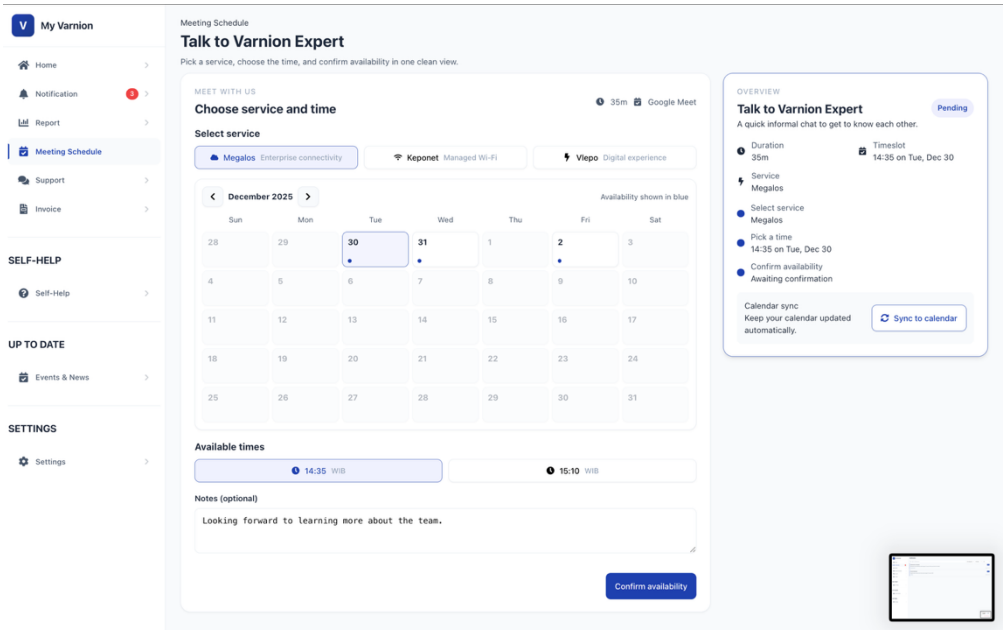


(b)

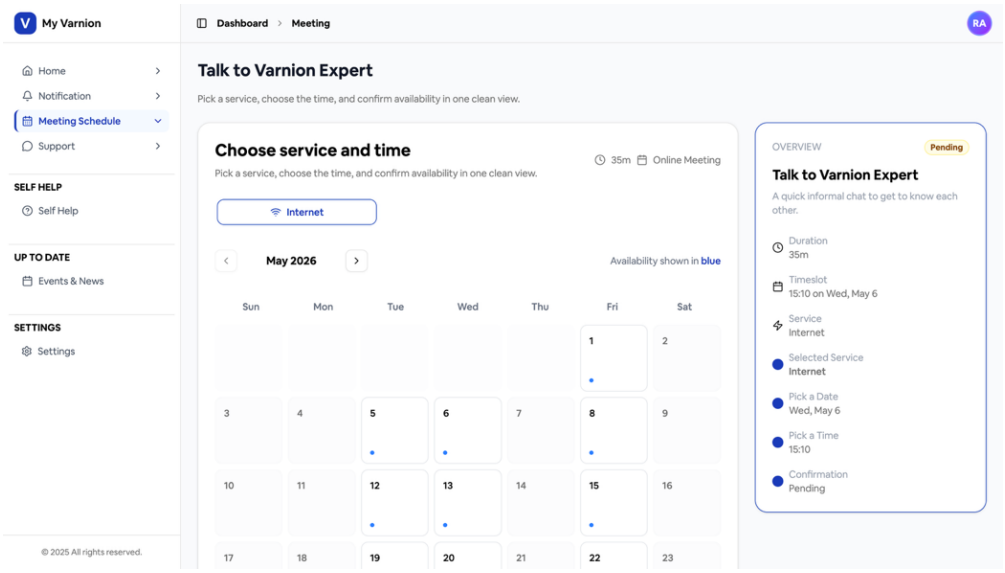
Gambar 3.22 Antarmuka Halaman Tiket Pelanggan

5) Halaman Pengajuan Pertemuan (*Meeting*)

Halaman ini merupakan tempat untuk pelanggan B2B mengajukan pertemuan kepada internal dan memungkinkan pengguna untuk memilih tanggal, jam, serta layanan. Setelah itu, permintaan yang diajukan akan berstatus *pending,* artinya menunggu persetujuan dari internal. Visualisasi halaman pengajuan pertemuan versi sebelumnya diperlihatkan pada Gambar 3.23 (a) yang memperlihatkan bahwa pengajuan belum berfungsi dan menyimpan data pengajuan di penyimpanan lokal *browser*. Sementara itu, versi terbaru yang telah didesain ulang diperlihatkan pada Gambar 3.23 (b) dan telah terintegrasi dengan sistem.



(a)

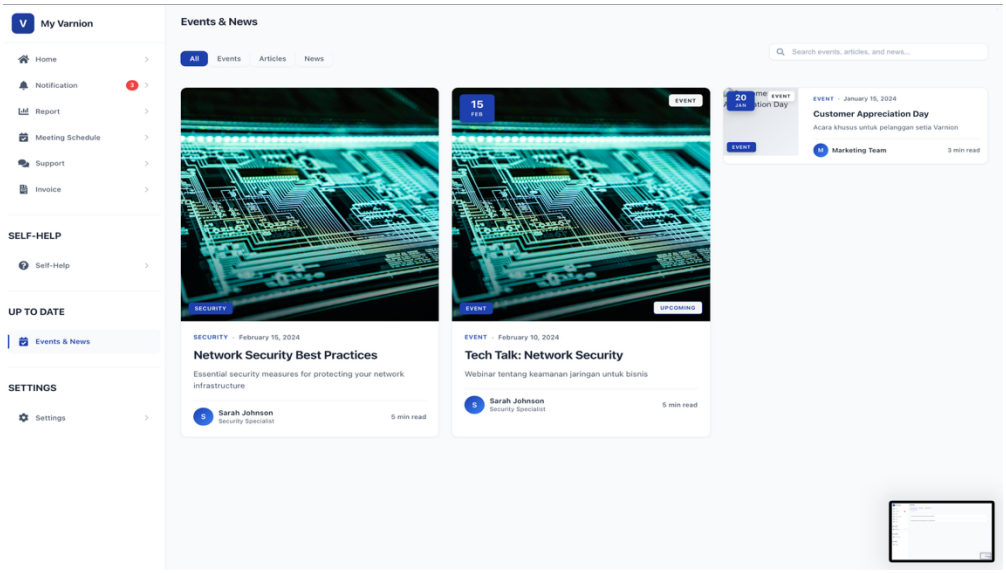


(b)

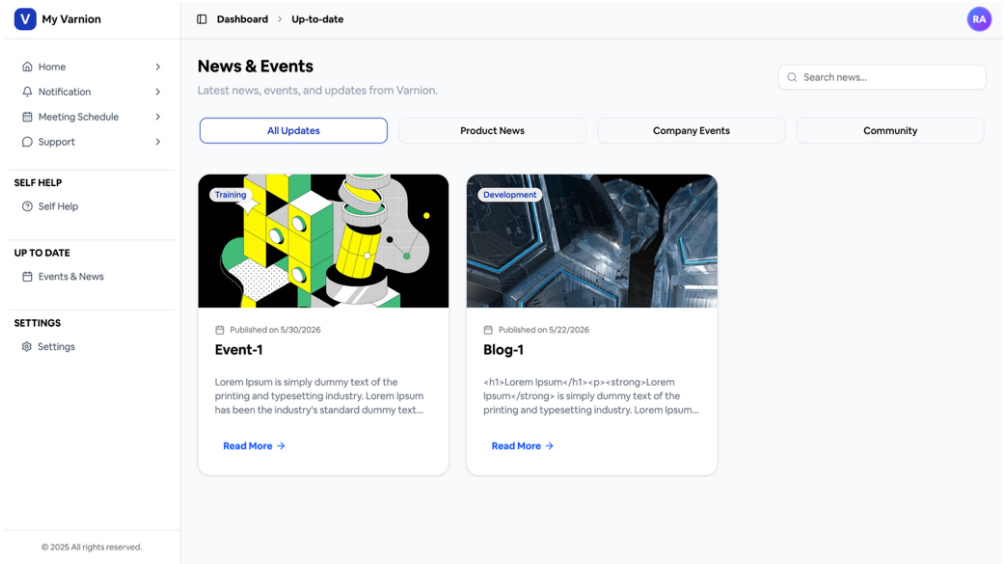
Gambar 3.23 Antarmuka Halaman Pengajuan Pertemuan Pelanggan

6) Halaman Informasi Berita & Acara (*News & Event*)

Pada halaman ini, pelanggan B2B dapat melihat informasi terkait berita dan acara. Selanjutnya, apabila pelanggan mengklik *event* maka diarahkan ke tempat untuk mendaftar acara itu sesuai yang telah diatur oleh staf melalui portal internal. Namun, jika pelanggan mengklik *blog,* akan diarahkan ke halaman detail blog tersebut. Tampilan daftar berita dan acara versi sebelumnya ditampilkan pada Gambar 3.24 (a), di mana sistem masih menampilkan data tiruan. Sementara itu, versi terbaru yang telah didesain ulang diperlihatkan pada Gambar 3.24 (b), merupakan versi yang terintegrasi, serta menampilkan data berita dan acara yang dibuat oleh staf di portal internal.



(a)

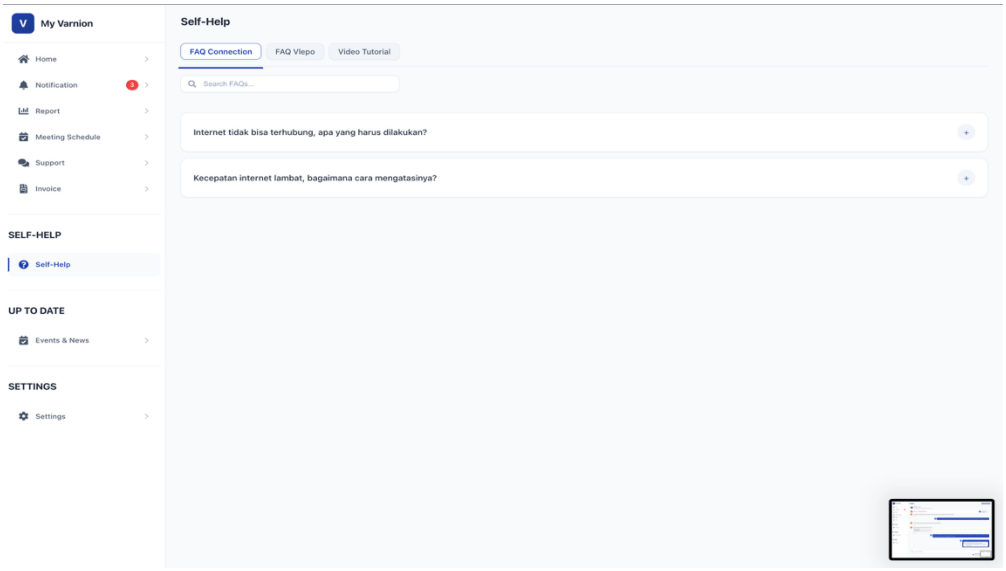


(b)

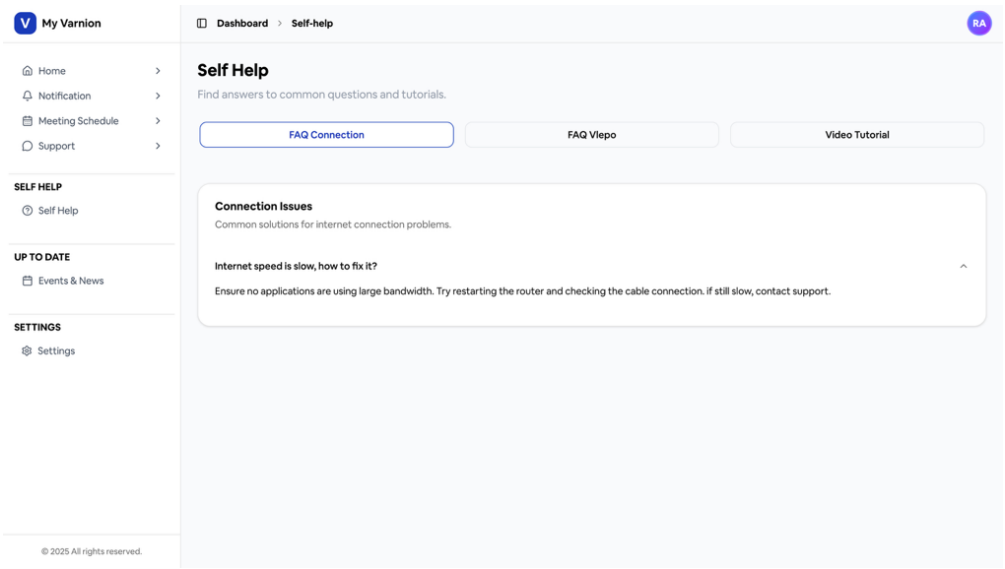
Gambar 3.24 Antarmuka Halaman Berita & Acara Pelanggan

7) Halaman Pusat Bantuan (FAQ)

Halaman ini menampilkan daftar pertanyaan beserta jawaban secara umum terkait layanan yang dapat dibaca oleh pelanggan B2B. Selain kumpulan pertanyaan, juga terdapat video tutorial yang telah disediakan oleh internal. Visualisasi kumpulan pertanyaan dan jawaban halaman FAQ versi sebelumnya ditampilkan pada Gambar 3.25 (a) yang masih menampilkan data tiruan. Sementara itu, versi terbaru hasil desain ulang ditampilkan pada Gambar 3.25 (b) dan menampilkan data FAQ yang telah dibuat oleh staf di portal internal.



(a)



(b)

Gambar 3.25 Antarmuka Halaman FAQ Pelanggan

3.4.4 Implementasi Fitur *Email Template Builder* & Integrasi API Resend

Tahap ini merupakan pembahasan mengenai implementasi fitur *email template builder* serta integrasi dengan layanan pihak ketiga, API Resend. Fitur ini dikembangkan untuk menjembatani kebutuhan komunikasi antara perusahaan (Varnion) dengan pelanggan B2B secara konsisten dan efisien. Dengan mengembangkan fitur ini, pengguna internal atau staf pada portal `my-varnion-admin` dapat merancang, mengatur, dan mengirimkan email notifikasi

terkait layanan dengan tampilan yang dinamis tanpa perlu mengubah kode pemrograman lapisan presentasi pada sistem.

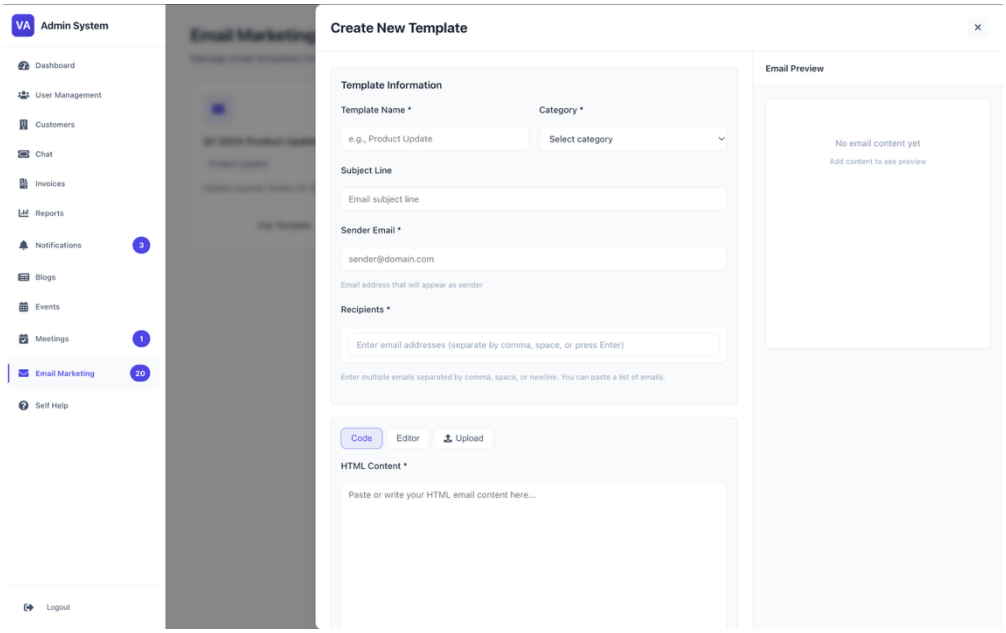
Secara teknis, implementasi fitur ini terbagi atas dua tahapan, yaitu pengembangan editor *template* email di sisi *frontend* dan integrasi sistem pengiriman data di sisi *backend*, dengan rincian sebagai berikut:

a. Pengembangan Fitur *Email Template Builder*

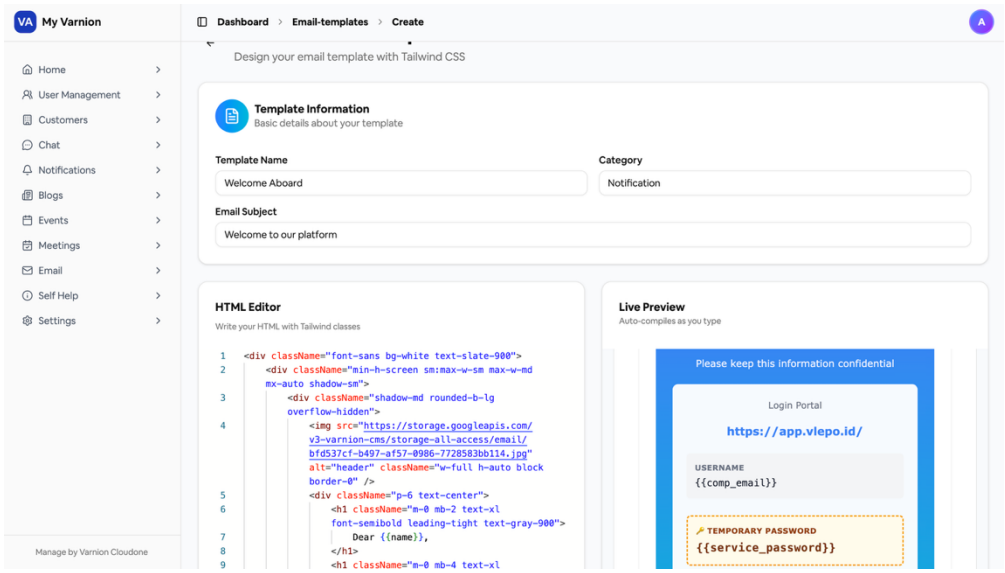
Pengembangan fitur ini menyediakan komponen antarmuka di sisi *frontend* yang memungkinkan staf internal untuk menyusun struktur teks, tata letak, dan gaya visual email secara dinamis seperti yang diperlihatkan pada Gambar 3.26 (a), sedangkan versi terbaru yang telah didesain ulang diperlihatkan pada Gambar 3.26 (b). Pada versi terbaru ini, dilengkapi dengan kemampuan penyisipan variabel dinamis, seperti `{{customer_name}}` atau `{{service_type}}` pada bagian editor HTML. Variabel yang disisipkan tersebut nantinya akan diisi dengan data rill sebelum dikirimkan kepada pelanggan B2B, sebagaimana diperlihatkan pada Gambar 3.27. Fitur penyisipan variabel dinamis ini merupakan pembaruan fungsional yang belum ada pada versi sebelumnya.

Template email yang dihasilkan berupa notifikasi seperti poster pembaruan layanan yang dapat diterima seluruh pelanggan, maupun *template* transaksional kustom untuk menampilkan data kredensial yang berbeda di setiap pelanggan seperti yang diperlihatkan pada Gambar 3.27 yang menampilkan proses pengisian data kredensial.

Untuk mendukung kompatibilitas dan konsistensi desain yang telah dibuat pada berbagai *email client* seperti Gmail, sistem mengimplementasikan React Email pada sisi *backend*. Alur pemrosesan ini dimulai dengan mengonversi kode HTML mentah beserta kelas Tailwind CSS menjadi elemen React menggunakan `html-react-parser`. Selanjutnya, elemen tersebut dibungkus dalam konfigurasi `<Tailwind>` dan diproses melalui modul `@react-email/render` untuk menghasilkan hasil akhir berupa *string* HTML dengan *inline styles.* Transformasi ini sangat krusial agar struktur visual dari *template* yang dibuat tidak berubah saat diterima oleh pelanggan seperti yang ditampilkan pada Gambar 3.28.

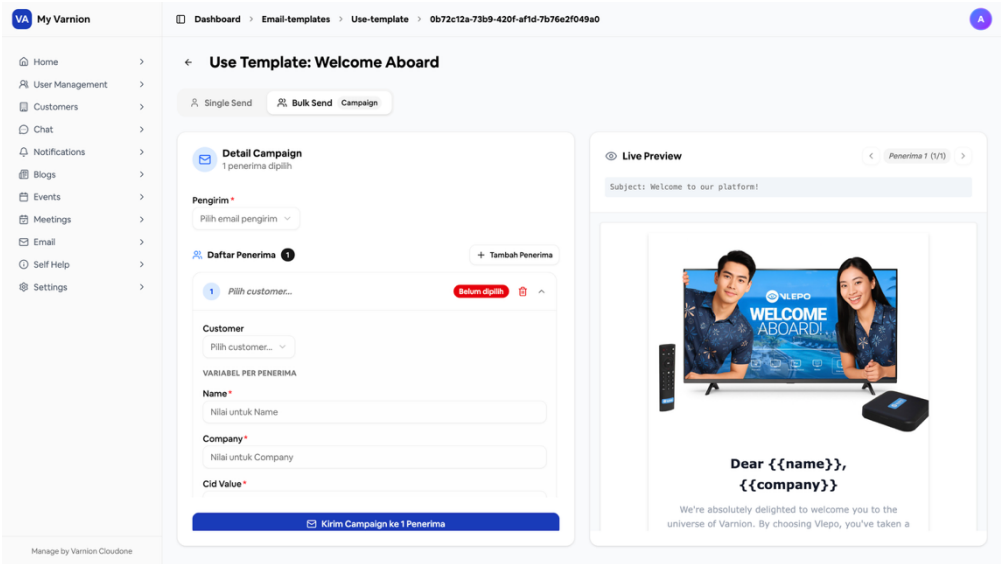


(a)

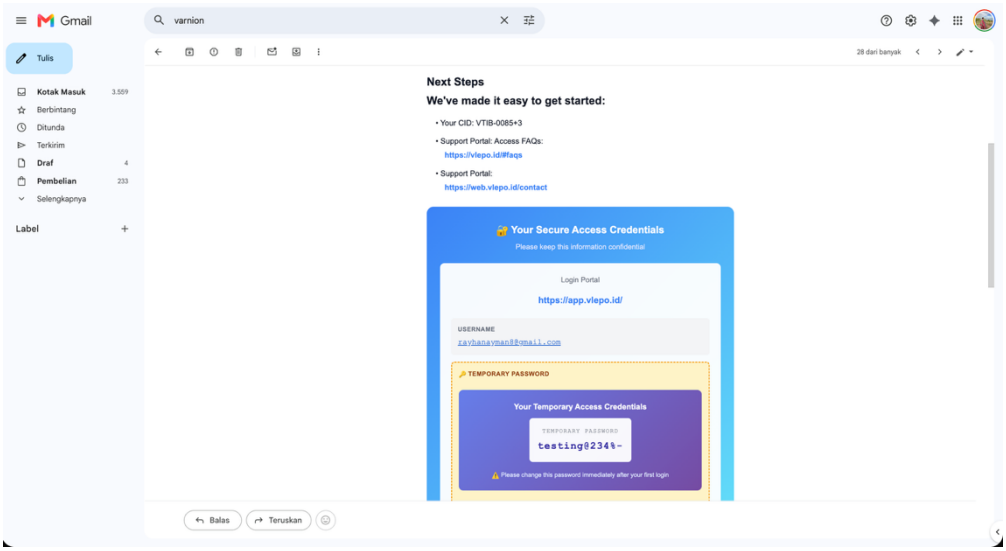


(b)

Gambar 3.26 Antarmuka Editor Halaman *Email Template*



Gambar 3.27 Antarmuka Pengisian Variabel Dinamis pada Halaman *Use Template*



Gambar 3.28 Tampilan Hasil *Template Email* yang Terkirim pada Gmail

b. Integrasi Pengiriman Email Menggunakan API Resend

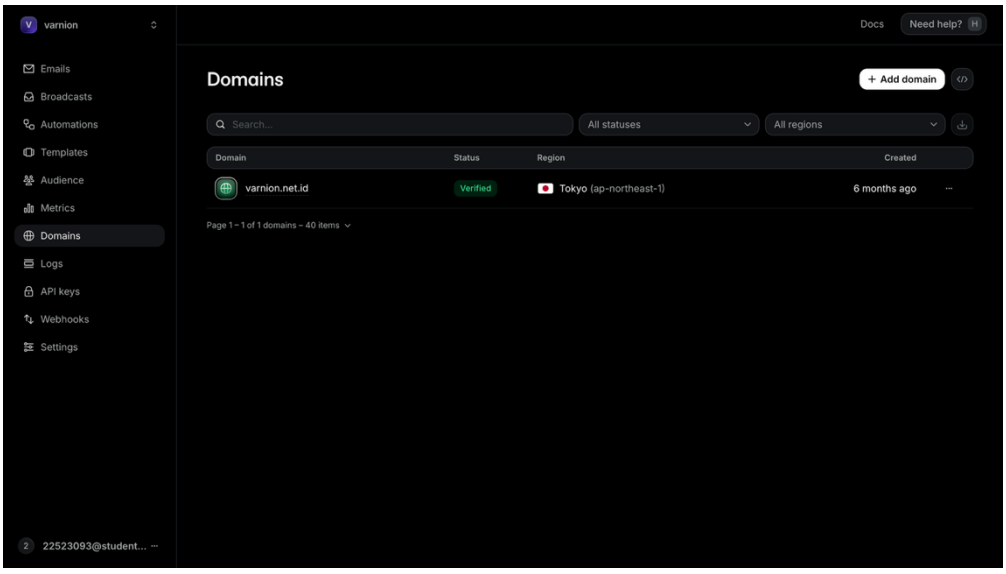
Setelah *template* email dirancang dan dikonversi menjadi string HTML dengan format *inline styles,* proses pengiriman data sepenuhnya ditangani oleh lapisan logika bisnis (*backend*). Aplikasi *backend* yang dibangun menggunakan Express.js diintegrasikan dengan layanan pihak ketiga yaitu, API Resend untuk memproses pengiriman email transaksional kepada pelanggan B2B.

Secara teknis, proses integrasi pengiriman email pada sisi *backend* diimplementasikan dengan beberapa tahapan berikut:

1) Konfigurasi Resend

Konfigurasi layanan eksternal Resend dilakukan pada dua sisi utama, yaitu melalui *dashboard* resmi Resend dan pada lapisan logika bisnis (*backend*) aplikasi My Varnion. Proses konfigurasi ini bertujuan untuk menghubungkan sistem dengan layanan email serta memastikan domain pengirim terverifikasi dengan baik.

Tahapan awal dimulai dengan mendaftarkan domain email resmi milik Varnion pada *dashboard* Resend. Setelah proses pendaftaran berhasil, platform Resend akan menghasilkan sebuah kunci akses unik yang disebut API *Key* serta data verifikasi keamanan seperti DKIM (*DomainKeys Identified Mail*) dan SPF (*Sender Policy Framework*). Data rekam DNS *records* tersebut wajib dikonfigurasikan pada panel manajemen DNS atau infrastruktur server yang digunakan. Namun, pada PT Varnion Technology Semesta, konfigurasi dilakukan melalui platform Proxmox. Proses ini sangat krusial untuk membuktikan otoritas server dalam mengirimkan pesan email, sehingga mencegah email masuk ke dalam folder spam pada *email client* pelanggan B2B. Visualisasi *dashboard* layanan Resend diperlihatkan pada Gambar 3.29.



Gambar 3.29 Tampilan *Dashboard* Resend

Setelah domain berhasil terverifikasi, konfigurasi pada repositori *backend* dilakukan dengan pengaturan variabel lingkungan (e*nvironment variables*). Parameter konfigurasi global sistem disimpan ke dalam file `.env.docker` demi menjaga keamanan data kredensial aplikasi. Namun, khusus untuk API *Key* yang dihasilkan oleh Resend, data tersebut disimpan secara dinamis di dalam basis data (*database*) melalui portal internal.

Selanjutnya, melakukan inisialisasi *library* dengan pemanggilan modul ke dalam skrip menggunakan sintaks `import { Resend } from 'resend'`. API *Key* Resend yang dimuat dari basis data kemudian dilewatkan untuk menginisialisasi objek Resend baru sebagai *instance* melalui perintah kode `const resend = new Resend(apiKey)`. Setelah itu, memanfaatkan metode utama yang disediakan oleh SDK Resend, yaitu `resend.emails.send` untuk memproses permintaan pengiriman email secara tunggal atau individu, serta `resend.batch.send` untuk menangani proses pengiriman email secara massal. Kode lengkap implementasi konfigurasi Resend dapat dilihat pada Gambar 3.30.

```
import { Resend } from 'resend';

export interface SendEmailParams {
     from: string
     to: string[]
     subject: string
     html: string
}

export const sendEmail = async (apiKey: string, params: SendEmailParams)
=> {
       if (!apiKey) {
            throw new Error("Resend API key is required");
       }

       const resend = new Resend(apiKey);

       const result = await resend.emails.send({
            from: params.from,
            to: params.to,
            subject: params.subject,
            html: params.html
       });

       return result;
}

export const sendBatch = async (apiKey: string, emails: SendEmailParams[])
=> {
       if (!apiKey) {
            throw new Error("Resend API key is required");
       }

        const resend = new Resend(apiKey);

        const results = await resend.batch.send(
            emails.map(email => ({
                 from: email.from,
                  to: email.to,
                  subject: email.subject,
                  html: email.html
        })));
        return results;
};
```

Gambar 3.30 Konfigurasi Fungsi Pengiriman Email

2) Penyediaan *Endpoint* API dan *Payload*

Setelah melewati proses konfigurasi Resend dari dua sisi, tahap selanjutnya adalah menyediakan kumpulan rute (*endpoint*) API untuk di konsumsi pada lapisan presentasi (*frontend*), khususnya pada repositori `my-varnion-admin`. Terdapat dua *endpoint* yang digunakan pada sisi *frontend,* yaitu `/api/v1/campaign/send` untuk pengiriman tunggal dan `/api/v1/campaign/send-bulk`, untuk pengiriman massal. Kedua *endpoint* tersebut didefinisikan di dalam file `campaign.route.ts`, seperti yang ditampilkan pada Gambar 3.31. Melalui file tersebut, terdapat dua endpoint API yang telah disediakan melalui dua fungsi utama, yaitu `sendOneOffEmailApi` dan `sendCampaignApi`.

```
import type { SendCampaignInput, SendOneOffEmailInput } from
"@/types/campaign.type";
import api from "./base/api";

const sendOneOffEmailApi = async (data: SendOneOffEmailInput): Promise<{
status: string; message: string }> => {
const response = await api.post("/campaign/send", data);
       return response.data;
}

const sendCampaignApi = async (data: SendCampaignInput): Promise<{ status:
string; message: string; data: any }> => {
const response = await api.post("/campaign/send-bulk", data);
return response.data;
}

export {
     sendOneOffEmailApi,
     sendCampaignApi
}
```

Gambar 3.31 Konfigurasi Rute *Endpoint* API *campaign.route.ts*

3) Keamanan Kredensial

Keamanan data kredensial merupakan aspek paling krusial dalam pengembangan perangkat lunak untuk mencegah kebocoran kunci akses dan eksploitasi. Pada sistem ini, seluruh konfigurasi yang bersifat sensitif, tidak di tulis langsung ke dalam baris kode (*hardcoded*). Oleh karena itu, lapisan logika bisnis menerapkan manajemen variabel lingkungan dengan memanfaatkan file konfigurasi `.env.docker`. File ini berfungsi sebagai wadah penyimpanan terpusat untuk seluruh variabel lingkungan yang nantinya akan dimuat secara otomatis ke dalam kontainer Docker saat aplikasi *backend*

dijalankan. Struktur dari isi berkas konfigurasi tersebut diperlihatkan pada Gambar 3.32.

Namun, terdapat pengecualian khusus untuk konfigurasi API *Key* serta data identitas pengirim (*email sender*) dari layanan Resend, data kredensial tersebut disimpan secara dinamis di dalam basis data (*database*) dan dimasukkan melalui halaman *settings* di `my-varnion-admin`, seperti yang diperlihatkan pada Gambar 3.33. Metode ini diterapkan agar perubahan terhadap data pengiriman email dapat dilakukan sewaktu-waktu tanpa perlu menjalankan ulang kontainer server. Selain itu, sistem menerapkan pembatasan hak akses, sehingga hanya staf dengan otoritas tertentu yang dapat mengakses, mengubah, dan melakukan konfigurasi terhadap API *Key* serta *email sender* pada panel internal.

```
# Application
NODE_ENV=development
PORT=3000

# Database (PostgreSQL)
DB_USER=postgres
DB_PASSWORD=postgres
DB_NAME=varnion_db
DB_PORT=5432
DATABASE_URL=postgresql://postgres:postgres@postgres:5432/varnion_db?schem
a=public

# JWT
JWT_SECRET=$2a$12$7j24NGobY5RZYY4YzAyFyOGQNrIppg2aglJ61LNt17E3mWCaqGW.i
JWT_EXPIRES_IN=30m

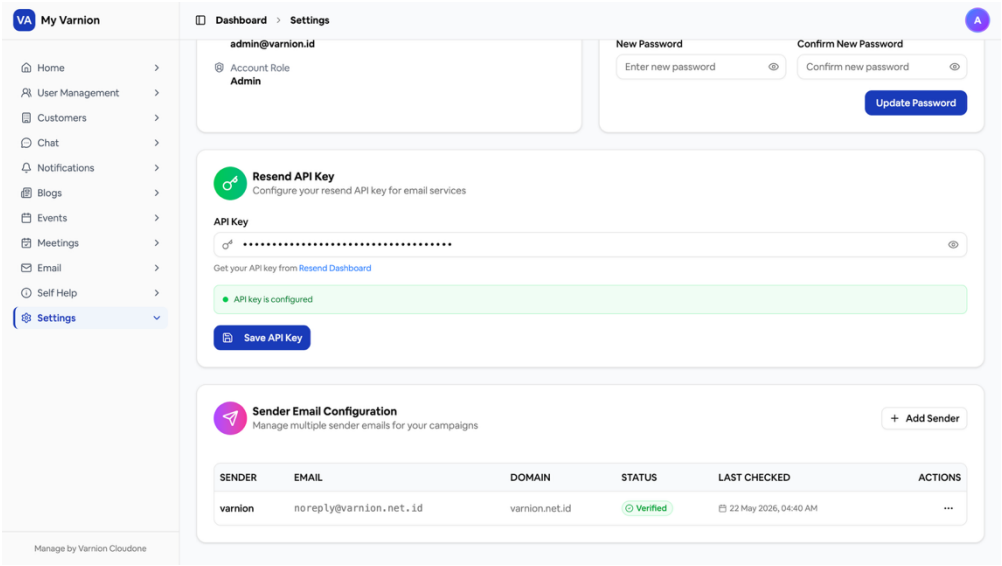
# CORS
CORS_ORIGIN=http://localhost:5173

# Rate Limiting
RATE_LIMIT_ENABLED=true
RATE_LIMIT_GLOBAL_MAX=100
RATE_LIMIT_AUTH_MAX=5
RATE_LIMIT_EMAIL_MAX=10

# Redis
REDIS_URL=redis://:xxx@redis:6379
REDIS_PASSWORD=xxx
REDIS_PORT=6379
REDIS_TLS=false

# pgAdmin
PGADMIN_EMAIL=admin@varnion.com
PGADMIN_PASSWORD=xxxxx
PGADMIN_PORT=5050
```

Gambar 3.32 Konfigurasi Variabel Lingkungan *.env.docker*



Gambar 3.33 Halaman *Settings* portal internal *my-varnion-admin*

3.4.5 Hasil Pengujian Sistem

Tahap pengujian sistem dilakukan untuk memastikan seluruh fitur yang dibuat pada aplikasi My Varnion berfungsi sesuai kebutuhan sebagaimana yang telah ditetapkan. Pengujian dilakukan menggunakan metode *black-box testing* pada beberapa skenario terhadap interaksi pengguna internal dan pelanggan B2B terhadap seluruh fitur utama sistem serta dari sisi pengalaman tanpa menampilkan seluruh kode.

Skenario pengujian pada aplikasi ini mencakup proses autentikasi kedua tipe pengguna (internal dan pelanggan), pengelolaan data pelanggan B2B, pengelolaan konten FAQ, berita dan acara, tiket, pengajuan pertemuan, *email template.* Serta akses pelanggan terhadap informasi profil dan layanan ketika mengakses portal pelanggan. Daftar terhadap hasil pengujian aplikasi My Varnion diperlihatkan pada Tabel 3.3.

Tabel 3.4 Hasil Pengujian Aplikasi

No.	Fitur yang Diuji	Skenario	Aktor	Hasil yang Diharapkan	Hasil Uji	Status
Pengujian fitur pada portal internal						
1	*Login* Internal	Staf memasukkan kredensial yang valid pada halaman *login* internal	Staf	Kredensial valid dan navigasi ke halaman *dashboard* internal	Sesuai Harapan	Berhasil
2	*Login* Internal	Staf memasukkan kredensial yang tidak valid pada halaman *login* internal	Staf	Kredensial tidak valid dan tidak navigasi ke halaman *dashboard* internal	Sesuai Harapan	Berhasil
3	Mengelola Data Pengguna Internal	Staf dengan hak akses tertentu menambah pengguna internal baru dengan mengisi seluruh data yang sesuai	Staf	Data pengguna internal berhasil tersimpan dan terlihat pada tabel daftar pengguna internal	Sesuai Harapan	Berhasil
4	Mengelola Data Pengguna Internal	Staf dengan hak akses tertentu menambah pengguna internal baru dengan mengisi kolom data yang tidak sesuai	Staf	Data gagal tersimpan, sehingga staf harus melengkapi atau mengisi kolom dengan data yang sesuai	Sesuai Harapan	Berhasil
5	Mengelola Data Pelanggan B2B	Staf menambahkan data pelanggan baru dengan mengisi seluruh data yang sesuai	Staf	Data pelanggan berhasil tersimpan dan terlihat pada tabel daftar pelanggan	Sesuai Harapan	Berhasil
6	Mengelola Data Pelanggan B2B	Staf menambahkan data pelanggan baru dengan tidak mengisi salah satu kolom masukan	Staf	Data pelanggan gagal tersimpan dan muncul peringatan untuk mengisi seluruh kolom dengan data yang sesuai	Sesuai Harapan	Berhasil
7	Mengelola FAQ	Staf menambahkan satu FAQ baru melalui portal internal	Staf	FAQ berhasil dibuat dan muncul pada portal pelanggan	Sesuai Harapan	Berhasil

No.	Fitur yang Diuji	Skenario	Aktor	Hasil yang Diharapkan	Hasil Uji	Status
8	Mengelola Berita dan Acara	Staf menambahkan berita atau acara baru dengan mengisi data yang valid pada komponen editor	Staf	Data berhasil tersimpan dan muncul pada portal pelanggan dan internal	Sesuai Harapan	Berhasil
9	Mengelola *Email Template*	Staf membuat *template email* dengan mengisi data HTML dan Tailwind CSS	Staf	Templat baru berhasil tersimpan dan dapat digunakan	Sesuai Harapan	Berhasil
10	Mengelola *Email Template*	Staf mengedit *template email* yang sudah ada	Staf	Revisi templat berhasil tersimpan dan dapat digunakan	Sesuai Harapan	Berhasil
11	Pengiriman Email ke Pelanggan B2B Melalui portal internal	Staf memilih salah satu templat yang akan dikirim ke pelanggan B2B serta mengisi seluruh kolom *input* data variabel dinamis	Staf	Email berhasil terkirim dan diterima dengan tampilan konsisten di perangkat pelanggan	Sesuai Harapan	Berhasil
12	Pengiriman Email ke Pelanggan B2B Melalui portal internal	Staf memilih salah satu templat yang akan dikirim ke pelanggan B2B namun tidak mengisi seluruh kolom *input* data variabel dinamis	Staf	Email gagal terkirim ke Gmail pelanggan	Sesuai Harapan	Berhasil
Pengujian fitur pada portal pelanggan						
13	*Login* Pelanggan	Pelanggan B2B memasukkan kredensial yang valid pada halaman login pelanggan	Pelanggan	Kredensial valid dan navigasi ke halaman *dashboard* pelanggan	Sesuai Harapan	Berhasil
14	*Login* Pelanggan	Pelanggan B2B memasukkan kredensial yang tidak valid pada halaman *login* pelanggan	Pelanggan	Kredensial tidak valid dan gagal navigasi ke halaman *dashboard* pelanggan	Sesuai Harapan	Berhasil

No.	Fitur yang Diuji	Skenario	Aktor	Hasil yang Diharapkan	Hasil Uji	Status
15	Mengajukan Pertemuan (*Meeting*)	Pelanggan mengajukan pertemuan dengan mengisi seluruh kolom dan memilih jadwal	Pelanggan	Pengajuan berhasil dengan status *pending* dan menunggu status konfirmasi internal	Sesuai Harapan	Berhasil
16	Mengelola Pengajuan Pertemuan (*Meeting*)	Staf mengelola daftar permintaan pertemuan dari pelanggan B2B dengan melakukan aksi menolak atau menerima pengajuan	Staf	Status berhasil berubah terhadap aksi yang dilakukan melalui portal internal dan perubahan status terlihat di portal pelanggan	Sesuai Harapan	Berhasil
17	Mengajukan Tiket	Pelanggan B2B melakukan pengajuan tiket terkait kendala atau layanan melalui halaman tiket pada portal pelanggan	Pelanggan	Tiket berhasil diajukan dan terlihat pada daftar tiket di halaman tiket portal internal	Sesuai Harapan	Berhasil
18	Mengelola Tiket Pelanggan	Staf merespons tiket yang diajukan pelanggan B2B melalui halaman tiket portal internal	Staf	Respon tiket berhasil terkirim dan terlihat pada kolom *chat* tiket pada portal pelanggan	Sesuai Harapan	Berhasil
Pengujian sistem Resend						
19	Pengiriman email melalui layanan email Resend	Resend memproses pengiriman *email template* yang dilakukan oleh staf ke Gmail pelanggan	Sistem	Resend berhasil mengirim templat ke Gmail pelanggan dan menampilkan status *endpoint* berhasil pada *dashboard log* Resend	Sesuai Harapan	Berhasil

Berdasarkan hasil pengujian yang diperlihatkan pada Tabel 3.3, seluruh skenario mencakup seluruh fitur pada aplikasi My Varnion memberikan hasil yang sesuai dengan hasil yang diharapkan, sehingga sistem telah berjalan dengan baik secara fungsional pada lingkungan pengujian. Pengujian pada aplikasi ini dilakukan oleh penulis karena aplikasi belum digunakan oleh pelanggan B2B secara resmi.

3.5 Kondisi Akhir dan Keberlanjutan Sistem

Setelah melalui tahap pengembangan aplikasi My Varnion yang dikerjakan oleh penulis, akses terhadap setiap repositori Github dari aplikasi tersebut telah dibuka secara penuh untuk akses internal (*internal access*) bagi pengembang lain di dalam divisi *Research and Development* (R&D) PT Varnion Technology Semesta.

Pembukaan *source code* ini bertujuan untuk mendukung pengembangan lanjutan aplikasi My Varnion oleh tim R&D pasca magang penulis berakhir. Oleh karena itu, tim R&D dapat melanjutkan ekspansi atau pengembangan sistem menggunakan alur kerja (*flow*) logika bisnis dan desain antarmuka (UI/UX) yang telah penulis bangun sebelumnya.

Proses pengembangan lanjutan oleh tim R&D dilakukan secara komprehensif dengan melibatkan seluruh anggota dalam tim berbagai peran (*role*). Melalui struktur tim yang lengkap di mana setiap anggota memiliki spesifikasi tugas masing-masing, diharapkan aplikasi My Varnion dapat disempurnakan hingga mencapai tahap siap produksi secara menyeluruh untuk melayani pelanggan B2B.

BAB IV
REFLEKSI PELAKSANAAN MAGANG

4.1 Relevansi Akademik

Berdasarkan pengalaman penulis dalam melakukan pengembangan aplikasi atau sistem yang telah didapatkan selama masa perkuliahan, terdapat perbedaan selama proses pengembangan, seperti pola kerja, teknologi yang digunakan, dan manajemen proyek. Berikut poin-poin perbedaan antara pengalaman selama perkuliahan dan yang didapatkan selama magang:

1. **Manajemen Proyek**

   Selama masa perkuliahan, proses pengembangan proyek umumnya belum dilakukan secara terstruktur dan terstandarisasi pada implementasinya. Aspek-aspek krusial seperti pembagian tugas, pengaturan jadwal kerja, dan metode koordinasi sering terjadi dengan tidak teratur karena perlu penyesuaian dengan anggota dalam tim walaupun hal tersebut telah dipelajari dan diimplementasikan pada mata kuliah Manajemen Pengembangan Teknologi Informasi.

   Di sisi lain, selama masa magang, proses pengembangan proyek dilakukan dengan menggunakan kerangka Scrum. Namun khusus bagi penulis, proses pengembangan terhadap proyek-proyek yang dikerjakan, salah satunya yaitu My Varnion, diimplementasikan dengan menggunakan metode pendekatan *Agile* yang mengadopsi beberapa elemen dari kerangka kerangka Scrum, seperti *sprint review* di setiap akhir pekan.

   Dalam pelaksanaannya, penulis tidak hanya mengerjakan instruksi dari *Product Lead*, tetapi juga memberi kontribusi berupa masukan terhadap fitur yang akan dikerjakan. Koordinasi dengan *Product Lead* rutin dilakukan dengan melakukan pertemuan (*meeting*) setiap minggu untuk meninjau kembali pekerjaan yang telah dikerjakan sebelumnya dan merencanakan target kedepannya. Selain itu, proses pelacakan tugas dikelola secara transparan menggunakan dokumen Google Spreadsheet yang dibuat oleh *Product Lead* untuk memantau status pengembangan setiap fitur.

2. Teknologi yang digunakan

Selama masa perkuliahan, penulis terbiasa melakukan pengembangan secara lokal tanpa kontainerisasi (Docker) pada proyek-proyek yang dikerjakan walaupun materi dasar dari Docker telah dipelajari di beberapa mata kuliah, seperti mata kuliah Sistem Jaringan dan Komputer dan Pengembangan Aplikasi Berbasis Web, serta menyusun kode secara manual dengan berpedoman materi pembelajaran dan membaca dokumentasi resmi. Namun selama magang, penulis dituntut untuk beradaptasi dengan berbagai teknologi baru yang sebelumnya belum cukup memahami dalam penggunaannya dalam konteks di industri. Beberapa di antaranya adalah penggunaan Docker untuk membungkus aplikasi (*containerization*), serta pemanfaatan kecerdasan buatan (*AI-assisted development*) untuk meningkatkan efisiensi dalam proses pengembangan. Penerapan metode tersebut bukan inisiatif penulis, melainkan arahan langsung dari *Product Lead* untuk diimplementasikan bagi seluruh anggota di tim R&D. Meskipun beberapa bagian dalam penulisan kode dibantu oleh AI, pemahaman teknis fundamental dari penulis tetap sangat diuji. Setiap baris kode yang dihasilkan oleh AI tidak secara langsung diimplementasikan dan diterima, melainkan harus di analisis dan disesuaikan dengan logika bisnis aplikasi yang telah ditetapkan agar sistem tetap berjalan konsisten dan sesuai dengan rancangan awal.

3. Peran dan fokus pekerjaan

Selama mengikuti kegiatan magang, pada hari pertama penulis ditugaskan oleh *Product Lead* untuk berperan sebagai *frontend developer* yang bertugas dalam menerapkan antarmuka aplikasi. Posisi ini ditujukan untuk memperbaiki *bug* atau *error* yang ditemukan pada sistem internal Varnion. Setelah menyelesaikan beberapa tugas atau tantangan yang diberikan, penulis diberi peran sebagai *fullstack developer* dalam menangani tiga proyek selama magang, seperti Pixku, Arclab, dan My Varnion. Peran ini menuntut penulis untuk menangani seluruh alur pengembangan proyek secara menyeluruh terhadap tiga lapisan utama arsitektur aplikasi, seperti perancangan sisi antarmuka (*presentation layer*), logika bisnis (*application layer*), dan basis data (*data access layer*), serta integrasi dengan layanan pihak ketiga (*third party*) yang digunakan pada proyek Arclab dan My Varnion, seperti API Resend.

Secara keseluruhan, pengalaman yang penulis dapatkan selama magang memberikan pemahaman dan keterampilan baru yang belum pernah didapatkan sebelumnya. Penulis menyadari adanya perbedaan antara metode kerja, perangkat pengembangan, dan peran yang dilakukan pada masa perkuliahan dengan praktik kerja di industri. Oleh karena itu, hal ini merupakan pelajaran berharga bagi penulis untuk terus berkembang sebagai pengembang sistem, sehingga ilmu yang didapat dapat diterapkan dalam lingkungan kerja yang lebih menantang.

4.2 Pembelajaran Magang

1. Manfaat

Pengalaman magang di PT Varnion Technology Semesta, selama kurang lebih enam bulan merupakan kesempatan yang sangat berharga bagi penulis. Lingkungan kerja yang suportif membantu penulis untuk menambah pengetahuan teknis, memperluas relasi profesional, serta memperbanyak sudut pandang baru terhadap dunia kerja di industri teknologi. Selama pelaksanaan magang, penulis dikenalkan dengan berbagai produk utama Varnion seperti Fiberzone, Vlepo, dan Keponet, sekaligus dilibatkan secara langsung dalam proses pengembangan proyek berbasis web seperti Pixku, Arclab, dan My Varnion, sehingga pemahaman penulis mengenai alur pengembangan aplikasi web di lingkungan industri menjadi semakin komprehensif.

Dari sisi kemampuan teknis, magang ini membantu penulis untuk memperdalam pemanfaatan teknologi pengembangan web modern, seperti Next.js, Express.js, PostgreSQL, Tailwind CSS, serta penggunaan *tools* pendukung saat pengembangan seperti Docker, Postman, VS Code, Antigravity, Figma, dan layanan email Resend, yang diintegrasikan ke dalam proyek My Varnion. Selain itu, pengalaman mengerjakan tiga proyek internal juga melatih dalam praktik menggunakan *version control* Git, melakukan integrasi dengan layanan *cloud* seperti Vercel dan Google Cloud, serta berkolaborasi dengan tim lintas divisi seperti divisi *creative* untuk menyelesaikan *task* dan melakukan *progress report* secara rutin.

2. Kendala, Hambatan, dan Tantangan

Selama mengikuti magang kurang lebih enam bulan, penulis tidak hanya mendapatkan manfaat secara praktis, tetapi memiliki beberapa tantangan. Tantangan yang pertama adalah menyesuaikan diri dengan ritme kerja di lingkungan industri yang

menuntut penyelesaian tugas secara tepat waktu dan pastinya berorientasi pada hasil, sehingga memungkinkan penulis untuk berlatih mengatur prioritas dan membagi waktu antara pengerjaan tugas kuliah, dan proyek magang. Selain itu, penulis juga perlu mempelajari beberapa teknologi dan alat kerja yang sebelumnya belum pernah digunakan, seperti *state management* pada ekosistem React JS, serta integrasi dengan layanan pihak ketiga seperti layanan email Resend, sehingga penulis sering menghabiskan waktu untuk membaca dokumentasi, menonton tutorial, dan berdiskusi dengan *Product Lead* maupun anggota tim lainnya agar dalam proses implementasinya sudah sesuai dengan metode kerja dalam tim.

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan pelaksanaan kegiatan magang yang telah dilakukan oleh penulis di PT Varnion Technology Semesta, serta pembahasan yang diuraikan pada bab sebelumnya, dapat disimpulkan bahwa tujuan penulisan tugas akhir ini telah tercapai. Penulis telah berhasil merancang dan mengembangkan aplikasi pengelolaan pelanggan B2B bernama My Varnion yang terintegrasi dengan fitur *email template builder*.

Selama pengembangan My Varnion, penulis ditugaskan untuk berperan sebagai *Fullstack Developer.* Penulis berhasil mengimplementasikan berbagai fitur utama, seperti pengelolaan pelanggan B2B dan manajemen *email template*, serta melakukan serangkaian pengujian untuk memastikan aplikasi dapat berjalan dan berfungsi sesuai dengan spesifikasi dan dokumentasi fitur yang diharapkan. Selain itu, proses pengembangan yang dilakukan menggunakan pendekatan *Agile* dengan mengadaptasi beberapa elemen dari kerangka kerja Scrum selama mengikuti magang di PT Varnion Technology Semesta.

Secara keseluruhan, kegiatan magang ini memberikan banyak wawasan dan pemahaman praktis dalam pengembangan aplikasi pengelolaan pelanggan B2B secara *end-to-end* bagi penulis. Selain meningkatkan kemampuan dalam penulisan kode, pengalaman ini juga melatih kemampuan analisis teknis. Oleh karena itu, memungkinkan bagi penulis untuk memberi kontribusi lainnya, seperti memberikan masukan kepada *Product Lead* terkait pengembangan fitur, sekaligus melatih adaptasi bekerja di lingkungan industri yang sesungguhnya.

5.2 Saran

Meskipun perancangan dan pengembangan aplikasi pengelolaan B2B My Varnion telah berhasil diselesaikan sesuai dengan target peran dan dokumentasi fitur, masih terdapat beberapa aspek yang perlu ditingkatkan. Saran-saran berikut disusun agar pengembangan dan pemeliharaan aplikasi kedepannya dapat berjalan lebih optimal dan berskala besar (*scalable*).

Pada sisi *backend*, meskipun penulis telah mengimplementasikan sistem hingga menjadi sebuah proyek yang berfungsi, masih terdapat ruang untuk mengeksplorasi terhadap fitur-fitur lanjutan. Hal ini disebabkan karena penulis mempelajari berbagai tumpukan teknologi (*tech stack*) yang digunakan secara mandiri dengan waktu yang terbatas selama magang. Oleh karena itu, diharapkan untuk program magang selanjutnya, pihak PT Varnion Technology Semesta

dapat menyediakan sesi mentoring teknis sehingga peserta magang dapat beradaptasi lebih cepat dan memaksimalkan potensi pengembangnya.

DAFTAR PUSTAKA

Azani, M. B. (2025). *Pengaruh Customer Relationship Management (CRM) Terhadap Customer Satisfaction Dan Customer Loyalty (Studi Systematic Literature Review)*. *5*, 250–263. https://doi.org/10.61104/alz.v4i1.3998

Darmawan, N., & Christianto, K. (2026). PERANCANGAN SISTEM INFORMASI B2B BERBASIS WEBSITE PADA PT. MITRA MAKMUR DWIJAYA. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *10*(1), 283–289. https://doi.org/10.36040/jati.v10i1.16625

Davis, A. (2023). *Creating a responsive, API-powered, PaaS-deployed, single-page application in React* [Bachelor's thesis, Degree Programme in Computer Applications, Häme University of Applied Sciences]. https://www.theseus.fi/bitstream/handle/10024/800734/Davis_Ashley.pdf?sequence=5&isAllowed=y

Fitrianto, K. A. (2023). *PENGEMBANGAN FRONT END SISTEM INFORMASI BERBASIS WEB MENGGUNAKAN FRAMEWORK CODEIGNITER DAN BOOTSTRAP (STUDI KASUS: KREATIFHUB)* [Universitas Islam Indonesia]. https://dspace.uii.ac.id/bitstream/handle/123456789/57600/18523192.pdf

Hardiansyah, F., Rizal, A., & Purnamasari, I. (2023). IMPLEMENTASI METODE AGILE SCRUM DALAM PENGEMBANGAN APLIKASI PEMBELAJARAN OLAHRAGA. *Jurnal Mahasiswa Teknik Informatika*, *7*(2). https://doi.org/10.36040/jati.v7i2.6734

Hartadi, T. B. (2025). *Pengembangan Basis Data Multilingual Menggunakan Headless CMS Directus pada Website Radyakartiyasa* [Tugas Akhir Jalur Magang, Universitas Islam Indonesia]. https://dspace.uii.ac.id/bitstream/handle/123456789/58605/21523102.pdf

Komperla, V., Deenadhayalan, P., Ghuli, P., & Pattar, R. (2022). React: A detailed survey. *Indonesian Journal of Electrical Engineering and Computer Science*, *26*(3), 1710–1717. https://doi.org/10.11591/ijeecs.v26.i3

Kristyna Vrbova. (2024). *What is an Email API? Everything you need to know*. https://sidemail.io/articles/what-is-email-api/

Muhammad, F., Fitri, I., & Nuraini, R. (2022). Implementasi Customer Relationship Management (CRM) pada Sistem Informasi Pemasaran dengan Menggunakan

Framework React.JS Berbasis Website. *Jurnal Teknologi Informasi Dan Komunikasi)*, *6*(1), 2022. https://doi.org/10.35870/jti

Nahjan, M. R., Ridha, A. A., Heryana, N., & Voutama, A. (2023). RANCANG BANGUN WEBSITE PENCARIAN INFORMASI BERITA DAN CUACA DAERAH DI INDONESIA MENGGUNAKAN API DAN EXPRESS.JS. *Jurnal Mahasiswa Teknik Informatika*, *7*(5), 3309–3313. https://doi.org/10.36040/jati.v7i5.7382

Pradana, M., Shiddiqi, A. M., & Triyono, B. (2025). TRANSFORMASI DIGITAL DALAM INDUSTRI DISTRIBUSI BAHAN BANGUNAN. *Citizen : Jurnal Ilmiah Multidisiplin Indonesia*, *5*(1), 250–263. https://doi.org/10.53866/jimi.v5i1.705

Prasetyo, S. E. (2021). Design and Implementation of Lightweight Virtualization Using Docker Container in Distributing Web Application with Experimental Methods. *JOURNAL OF INFORMATICS AND TELECOMMUNICATION ENGINEERING*, *4*(2), 270–276. https://doi.org/10.31289/jite.v4i2.4019

Quzwen, M. H., Hurriyati, R., & Sultan, M. A. (2021a). *Customer Relationship Management in Business-to-Business Marketing: The Impact on Corporate Performance*. 517–523. https://doi.org/10.2991/aebmr.k.210831.101

Quzwen, M. H., Hurriyati, R., & Sultan, M. A. (2021b). Customer Relationship Management in Business-to-Business Marketing: The Impact on Corporate Performance. *Proceedings of the 5th Global Conference on Business, Management and Entrepreneurship (GCBME 2020)*, 517–523. https://doi.org/10.2991/aebmr.k.210831.101

Raharjo, D. A. S. (2023). Implementasi Metode Scrum Dan Analisis Swot Sebagai Strategi Framework Customer Relationship Management (CRM) Pada Perusahaan Aqilla Rent Car Semarang. *Sammajiva: Jurnal Penelitian Bisnis Dan Manajemen*, *1*(3), 345–361. https://doi.org/10.47861/sammajiva.v1i3.927

Resend. (n.d.). *Email API · Start sending today.* Retrieved April 20, 2026, from https://resend.com/features/email-api

Rezkika, W. R., Handayani, M., & Larasati, M. F. (2026). Implementasi Sistem Customer Relationship Management (CRM) Berbasis Website untuk Pengelolaan Membership dan Loyalty Point pada CV. Honda Karya Utama Kisaran. *Jurnal Sistem Informasi Akuntansi*, *7*(1), 50–58. https://doi.org/10.31294/justian.v7i1.12220

Ridhalri, R. (2022). PEMANFAATAN CACHING SYSTEM MENGGUNAKAN REDIS UNTUK SISTEM PENGELOLAAN INFORMASI AMBALAN ASHABUL KAHFI

BERBASIS WEB. *Jurnal DIALOGIKA : Manajemen Dan Administrasi*, *4*(1), 39–56. https://doi.org/10.31949/dialogika.v4i1.3750

Roihan, A., Arribathi, A. H., & Nurussaadah, N. (2019). Desain Aplikasi B2b Sistem Manajemen Pergudangan dalam Penunjang Keputusan Bisnis. *Creative Communication and Innovative Technology Journal*, *12*(1), 33–40. https://ejournal.raharja.ac.id/ccit/article/view/598/527

Sabrina. (2024). *Apa itu Scrum? Komponen & Implementasi dengan Scrum Board.* Binar Academy. https://www.binar.co.id/blog/apa-itu-scrum-komponen-implementasi-dengan-scrum-board

Saiholau, M. N. (2024). RANCANG BANGUN BACKEND WEBSITE PEMUNGUTAN SUARA DENGAN MENGGUNAKAN FRAMEWORK EXPRESS.JS. *Jurnal Informatika Dan Teknik Elektro Terapan*, *12*(2). https://doi.org/10.23960/jitet.v12i2.4261

Sangeetha, K. (2016). E-Marketing. *PARIPEX - Indian Journal of Research*, *5*, 166–167. https://www.worldwidejournals.com/paripex/recent_issues_pdf/2016/January/emarketing_January_2016_4361005997_0105569.pdf

Saputra, N. A., & Firdonsyah, A. (2025). Penerapan docker pada website reschedule class menggunakan metode docker compose. *Prosiding Seminar Nasional Penelitian Dan Pengabdian Kepada Masyarakat LPPM Universitas 'Aisyiyah Yogyakarta*, *3*, 837–840. https://proceeding.unisayogya.ac.id/index.php/prosemnaslppm/article/view/1164/604

Sarmanela, C. B., Samiya, A. N., & Yuliana, M. E. (2023). *Mengenal Backend Developer: Skill, dan Tanggung Jawab Menjadi Backend Developer*. 128–134. https://ojs.udb.ac.id/Senatib/article/view/3019/2579

Siallagan, Y. O., Saragih, N. F., & Hutapea, J. M. (2020). IMPLEMENTASI CUSTOMER RELATIONSHIP MANAGEMENT(CRM) PADA TOKO ROTI GANDA BERBASIS WEB. *Jurnal METHODIKA*, *6*(1), 6–13. https://doi.org/10.46880/mtk.v6i1.247

Simarmata, S., Agustya, P. D., Ramadhan, A. R., Putro, D. R. T., Aziz, F. S. N., Ikhsan, I. N., Fauzi, M. A., Anwar, L., Ghozi, M. F., & Primadillah, R. (2025). Rancang Bangun Sistem Helpdesk Berbasis Web Menggunakan Laravel dan PostgreSQL Untuk Optimalisasi Pengelolaan Keluhan Di Mts Pembangunan Nurul Islam. *RIGGS: Journal of Artificial Intelligence and Digital Business*, *4*(4), 4533–4538. https://doi.org/10.31004/riggs.v4i4.4312

Sselvianto, S. C., Angelika, S., Daely, G., & Wijaya, A. (2025). Implementasi Data Warehouse Industri Film Menggunakan PostgreSQL. In *Jurnal Sistem Informasi dan Teknologi Peradaban (JSITP)* (Vol. 6, Number 2). www.journal.peradaban.ac.id

Suwandi, E. A., Mutohar, P. M., & Suijianto, A. E. (2023). Implementasi Customer Relationship Management (CRM) dalam Meningkatkan Pelayanan dan Loyalitas Pelanggan. *JMPI: Jurnal Manajemen, Pendidikan, Dan Pemikiran Islam Journal.as-Salafiyah*, *1*(1), 13–26. https://journal.as-salafiyah.id/index.php/jmpi

Tu, Z. (2023). Research on the Application of Layered Architecture in Computer Software Development. *Journal of Computing and Electronic Information Management*, *11*(3), 34–38. https://pdfs.semanticscholar.org/65d0/4f4679c345d8bed2fbdc957eb7aae3b67356.pdf

Warkim, W., Muslim, M. H., Harvianto, F., & Utama, S. (2020). Penerapan Metode SCRUM dalam Pengembangan Sistem Informasi Layanan Kawasan. *Jurnal Teknik Informatika Dan Sistem Informasi*, *6*(2). https://doi.org/10.28932/jutisi.v6i2.2711

LAMPIRAN

Lampiran A



Lampiran B

