

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Setiap perusahaan memiliki struktur organisasi yang menunjukkan kedudukan serta hubungan koordinasi antara departemen dan divisi di dalamnya. Kedudukan merupakan posisi atau tempat seseorang dalam struktur organisasi [15], sedangkan koordinasi adalah aktivitas untuk menghubungkan berbagai bagian dalam organisasi agar tercipta kerja sama yang harmonis, menghindari kesalahan komunikasi, dan menyelaraskan sumber daya menuju tujuan yang ditetapkan [16].

3.1.1 Kedudukan

Kegiatan magang dilaksanakan di Departemen IT MIS, yang merupakan bagian dari Divisi IT *Infrastructure*, dengan kedudukan sebagai ERP *Developer*. Selama pelaksanaan magang, bimbingan diberikan langsung oleh *Head* ERP sebagai supervisor dan salah satu anggota tim ERP *Developer* sebagai mentor. Bentuk bimbingan ini mencakup pengarahan terkait mekanisme pengembangan aplikasi, yang dilaksanakan dengan koordinasi yang baik. Dalam struktur organisasi, kedudukan magang berada di bawah tim ERP *Developer*, dengan tanggung jawab mendukung pelaksanaan tugas-tugas teknis serta berkontribusi dalam pengembangan sistem ERP di lingkungan perusahaan.

ERP *Developer* memiliki peran penting dalam merancang dan mengembangkan solusi *Enterprise Resource Planning* (ERP) yang efektif. Tugas utamanya dimulai dengan melakukan analisis atau riset kebutuhan bisnis untuk memahami secara mendalam sistem yang dibutuhkan perusahaan. Dalam proses ini, ERP *Developer* juga aktif berkomunikasi dengan pengguna bisnis untuk menggali kebutuhan spesifik mereka terhadap sistem ERP. Selain itu, ERP *Developer* bertanggung jawab dalam proses pengembangan aplikasi, khususnya pada modul *Human Resources Information System* (HRIS), dengan

memanfaatkan teknologi seperti PHP (Laravel), HTML, dan CSS untuk memastikan sistem berjalan optimal sesuai kebutuhan perusahaan.

3.1.2 Koordinasi

Koordinasi dan komunikasi antar anggota departemen dilakukan melalui berbagai media, seperti WhatsApp, Google Meet, atau secara langsung di kantor. Selama pelaksanaan kerja magang, seluruh aktivitas dilakukan secara penuh di kantor (*full Work From Office/WFO*), dengan jam kerja dimulai pukul 08.00 pagi hingga 17.00 sore. Pola kerja ini memungkinkan kolaborasi yang lebih efektif serta memudahkan proses diskusi dan evaluasi tugas secara langsung.

Koordinasi dengan supervisor memegang peran penting dalam kelancaran proses pengerjaan tugas sebagai *ERP Developer*. Setiap tugas yang diberikan biasanya memiliki estimasi waktu pengerjaan antara 1 hingga 3 hari. Setelah tugas diselesaikan, hasil pekerjaan dilaporkan kepada supervisor untuk didiskusikan bersama. Melalui diskusi ini, supervisor dapat memberikan masukan dan memastikan bahwa pekerjaan telah sesuai dengan standar serta kebutuhan yang ditetapkan. Koordinasi yang intens dan terarah ini memungkinkan setiap tahapan pekerjaan berjalan secara terstruktur dan efisien. Dengan komunikasi yang baik, pemahaman terhadap ekspektasi menjadi lebih jelas, sehingga pelaksanaan tugas selanjutnya dapat dilakukan dengan lebih tepat sasaran.

Melalui pelaksanaan tugas yang diberikan, kemampuan teknis dan pemahaman praktis terhadap proses kerja di dunia industri dapat diasah, khususnya dalam pengembangan sistem ERP. Di samping itu, pengalaman dalam berkomunikasi secara profesional, bekerja secara tim, serta memahami alur kerja dan tanggung jawab yang harus dijalankan dalam lingkungan kerja yang sesungguhnya turut diperoleh. Keseluruhan pengalaman ini menjadi bekal berharga untuk mempersiapkan diri menghadapi dunia kerja setelah menyelesaikan studi.

3.2 Tugas dan Uraian Kerja Magang

Pada pelaksanaan kerja magang di PT. Trans Digital Media yang berlangsung selama enam bulan, terdapat beberapa tugas dan uraian dari tugas kerja magang tersebut yang dilaksanakan selama proses magang berlangsung. Sebagai *ERP Developer*, terdapat tanggung jawab untuk menyelesaikan semua tugas yang diberikan dengan baik dan sesuai dengan standar perusahaan. Tugas dan uraian ini dapat diberikan oleh supervisor atau mentor yang membutuhkan bantuan untuk kelangsungan proyek yang berjalan. Tabel 3.1 memperlihatkan pelaksanaan tugas kerja magang, yang mencakup proyek HR *Newsletter* yang dilaksanakan, kegiatan yang dilakukan dalam proyek tersebut, serta tanggal mulai dan tanggal selesai masing-masing kegiatan proyek. HR *Newsletter* adalah platform yang dirancang untuk mempermudah departemen HR dalam membuat dan mengirim *newsletter* sebagai media penyebaran informasi internal.

Tabel 3.1 Pelaksanaan Tugas Kerja Magang

Proyek	Kegiatan	Mulai	Selesai
Proyek HR <i>Newsletter</i>	Riset <i>newsletter</i> dan analisis kebutuhan HR	3 Februari 2025	6 Februari 2025
	Pembuatan konsep dan fitur aplikasi <i>website</i>	6 Februari 2025	11 Februari 2025
	Pembuatan desain <i>template newsletter</i> dan prototipe aplikasi <i>website</i>	12 Februari 2025	17 Februari 2025
	<i>Review</i> dan finalisasi desain	17 Februari 2025	28 Februari 2025
	Pengembangan <i>frontend</i> dan <i>backend</i> aplikasi <i>website</i>	3 Maret 2025	20 Juni 2025

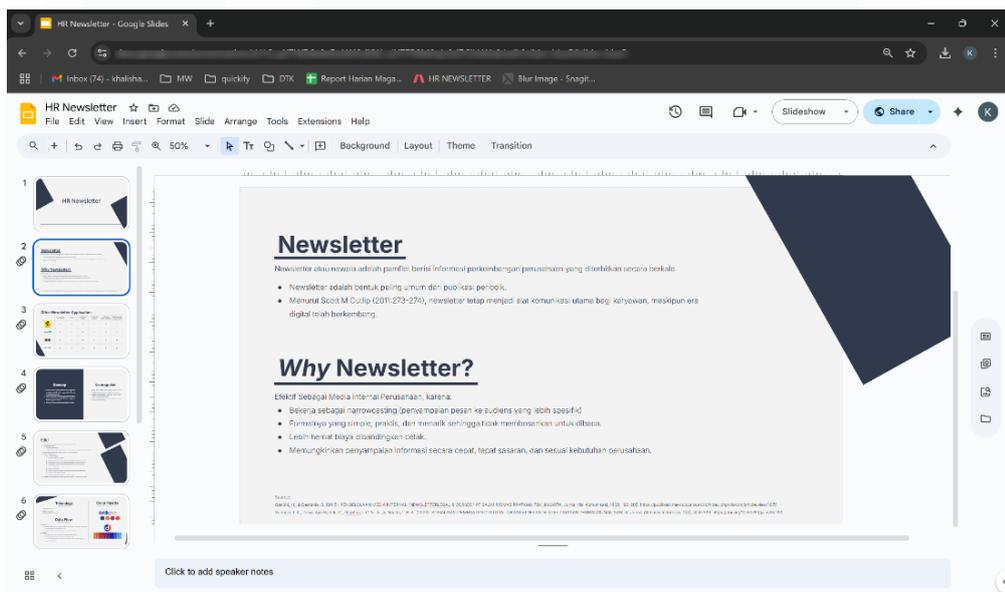
(Sumber olahan peneliti, 2025)

3.2.1 Riset *Newsletter* dan Analisis Kebutuhan HR

Pelaksanaan proyek HR *Newsletter* dimulai pada minggu pertama masa magang dengan melakukan riset terkait pengembangan *newsletter*. Riset ini bertujuan untuk memahami secara mendalam pengembangan aplikasi berbasis *website* yang dapat membantu tim HR dalam membuat, mengelola, dan mengirimkan konten *newsletter* secara efisien kepada seluruh karyawan di

detikcom. Riset mencakup analisis konsep, fitur, serta kebutuhan lainnya yang diperlukan dalam proyek ini.

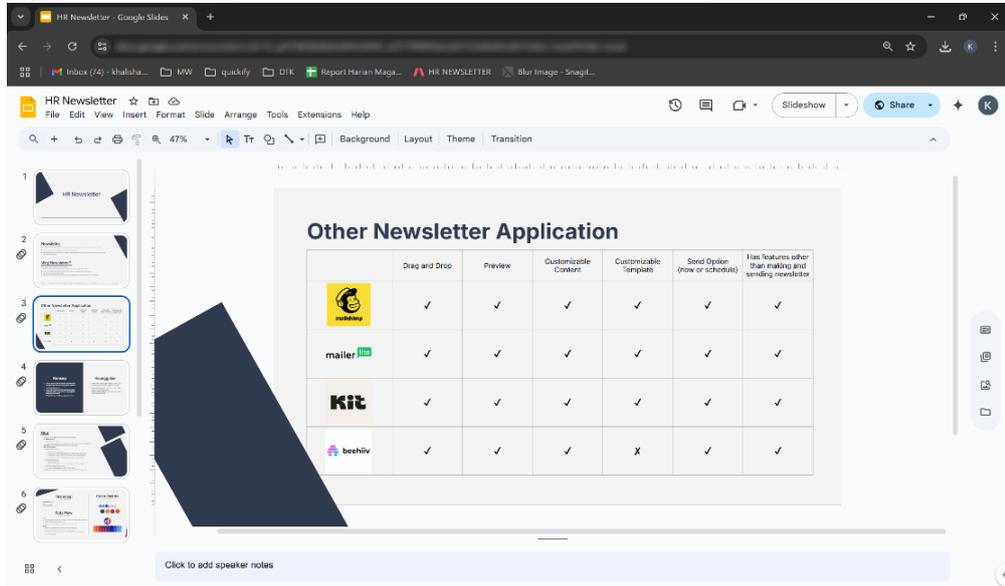
Riset diawali dengan pencarian mendalam mengenai *newsletter* melalui beberapa jurnal dan artikel di Google Chrome. Pencarian dimulai dari pengertian *newsletter*, yaitu salah satu bentuk publikasi periodik yang berfungsi sebagai alat komunikasi utama bagi karyawan. Selanjutnya, dibahas karakteristik *newsletter* serta alasan mengapa media ini menjadi salah satu alat komunikasi perusahaan yang efektif, yaitu karena sifatnya yang praktis, hemat biaya, dan tepat sasaran. Gambar 3.1 menunjukkan hasil riset terkait pengertian, karakteristik, serta alasan efektivitas penggunaan media *newsletter* pada Google Slides.



Gambar 3.1 Pengertian, Karakteristik, serta Alasan Efektivitas Penggunaan Media *Newsletter*

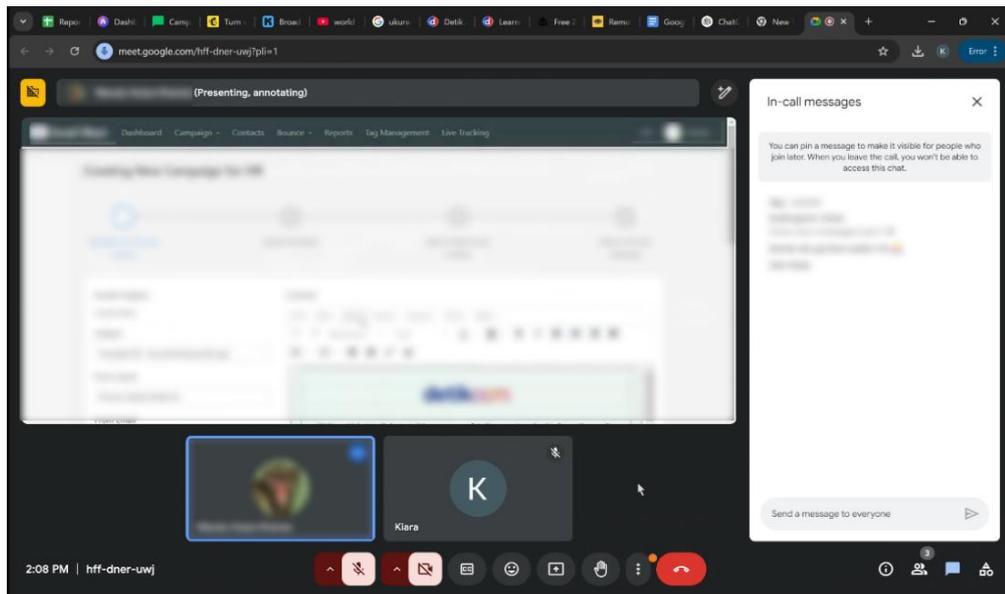
Kemudian, riset dilanjutkan dengan menelusuri berbagai aplikasi *website* yang dirancang untuk membuat dan mengirimkan *newsletter*, khususnya yang sedang populer saat ini, melalui Google Chrome. Beberapa aplikasi yang dijadikan referensi antara lain Mailchimp, MailerLite, KIT, dan Beehiiv. Setelah mencoba keempat aplikasi tersebut, dilakukan analisis terhadap kelebihan dan kekurangan masing-masing. Hasil analisis ini kemudian

dibandingkan dan dimasukkan ke dalam Google Slides dalam bentuk tabel, seperti pada Gambar 3.2.



Gambar 3.2 Perbandingan Empat Aplikasi *Newsletter*

Setelah itu, dilakukan wawancara dengan salah satu anggota departemen HR untuk menggali lebih dalam kebutuhan departemen tersebut terkait pengembangan aplikasi *website* ini. Gambar 3.3 menampilkan dokumentasi kegiatan wawancara yang dilakukan sebagai bagian dari proses identifikasi kebutuhan departemen HR. Topik yang dibahas dalam wawancara mencakup aplikasi *website* yang sebelumnya digunakan, kelebihan dan kekurangannya, serta harapan dan kebutuhan HR terhadap aplikasi *website* yang akan dikembangkan.



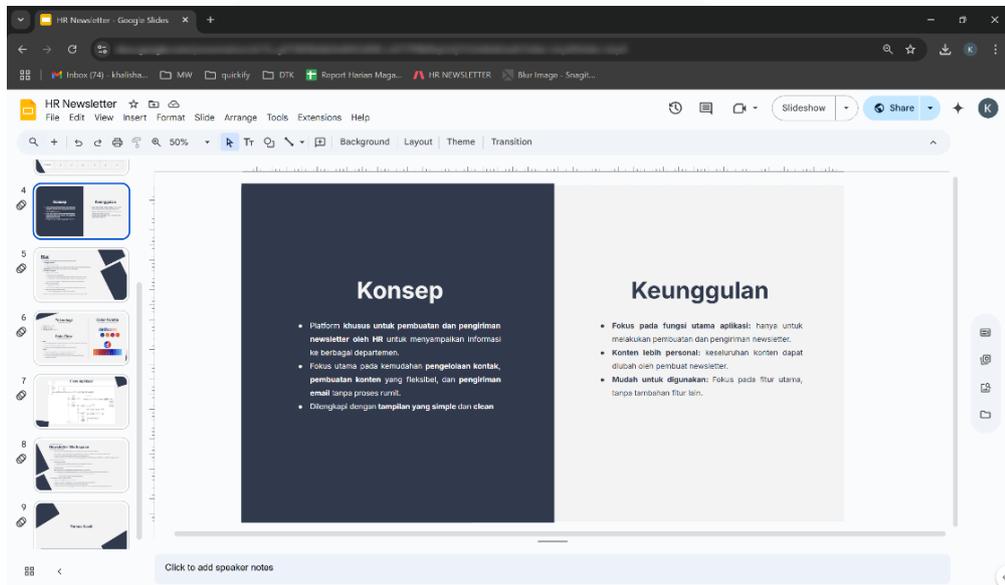
Gambar 3.3 Wawancara mengenai kebutuhan HR

Berdasarkan hasil wawancara, diketahui bahwa departemen HR sebelumnya telah menggunakan sebuah aplikasi *website* untuk membuat dan mengirimkan *newsletter*. Namun, aplikasi tersebut memiliki beberapa kekurangan, seperti jumlah fitur yang terlalu banyak sehingga banyak di antaranya tidak dimanfaatkan, antarmuka yang sulit dinavigasi oleh pengguna, serta keterbatasan dalam menyesuaikan konten sesuai dengan kebutuhan. Hal-hal tersebut menjadi alasan utama perlunya pengembangan atau peningkatan aplikasi *website* untuk pembuatan dan pengiriman HR *Newsletter* yang lebih sesuai dengan kebutuhan pengguna.

3.2.2 Pembuatan Konsep dan Fitur Aplikasi *Website*

Setelah melakukan riset dan analisis kebutuhan HR, tahap selanjutnya adalah merancang konsep dan fitur aplikasi *website* dengan mengumpulkan hasil riset dan analisis yang telah diperoleh. Hasil riset dan analisis kebutuhan HR menunjukkan bahwa aplikasi *website* sebelumnya memiliki terlalu banyak fitur tambahan seperti analitik, sehingga banyak di antaranya tidak dimanfaatkan. Konsep aplikasi *website* yang digunakan adalah platform khusus untuk pembuatan dan pengiriman *newsletter* oleh HR untuk menyampaikan informasi ke berbagai departemen, sebagai solusi untuk

mengatasi masalah tersebut. Gambar 3.4 menunjukkan konsep aplikasi yang sudah dimasukkan ke dalam Google Slides.



Gambar 3.4 Konsep Aplikasi pada Google Slides

Fokus utama aplikasi ini adalah kemudahan dalam pengelolaan kontak, pembuatan konten yang fleksibel, serta pengiriman email yang sederhana tanpa proses yang rumit. Tampilan aplikasi ini juga dirancang simpel dan bersih agar aplikasi *website* lebih mudah untuk dinavigasi oleh pengguna. Berikut adalah fitur-fitur yang akan ada pada aplikasi *website* HR Newsletter.

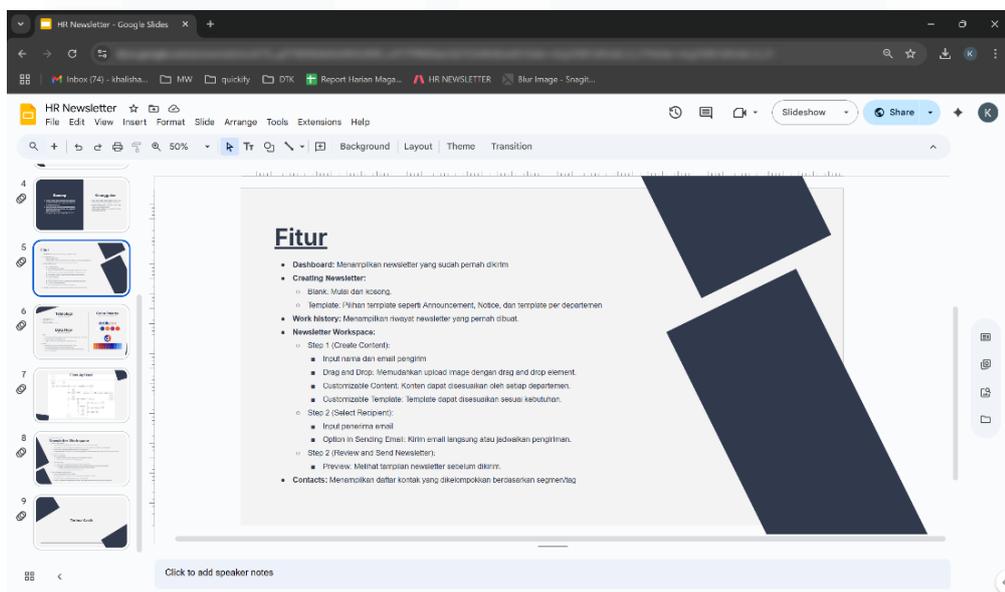
Pertama, terdapat fitur *Dashboard*, yaitu fitur yang menampilkan tabel untuk melihat *newsletter* yang telah dikirim. Tabel ini dilengkapi dengan informasi seperti nama *newsletter* yang dibuat, tanggal pengiriman, penerima *newsletter*, jumlah yang sudah terkirim, serta status pengiriman *newsletter*. Status pengiriman dapat mencakup beberapa kondisi, yaitu: "*Completed*" jika semua *newsletter* berhasil terkirim, "*Partially Sent*" jika hanya sebagian *newsletter* yang berhasil terkirim, "*Failed*" jika semua *newsletter* gagal terkirim, "*Scheduled*" jika tanggal pengiriman sudah dijadwalkan namun belum mencapai tanggal tersebut, dan "*Pending*" jika pengiriman masih menunggu atau berada dalam antrian pengiriman *newsletter* lainnya.

Kedua, terdapat fitur *Create Newsletter*, yaitu fitur yang menyediakan halaman kosong (*blank page*) dan beberapa *template* yang bisa digunakan oleh pengguna. *Template* tersebut disusun dalam bentuk *grid* dengan *preview* untuk memudahkan pengguna melihat secara visual pilihan yang tersedia. Pengguna dapat memilih halaman kosong untuk membuat *newsletter* dari awal sesuai keinginan, atau memilih *template* yang sudah disediakan dan menyesuaikan *layout* serta isinya sesuai kreativitas dan kebutuhan mereka.

Ketiga, terdapat fitur *Newsletter Workshop*, yaitu fitur yang dirancang untuk memfasilitasi pengguna dalam menyusun dan mengatur *newsletter* mereka. Fitur ini terbagi menjadi tiga bagian, yaitu *Create Content*, *Select Recipient*, dan *Preview and Send Newsletter*. Pada bagian pertama (*Create Content*), pengguna dapat mengisi informasi pengiriman seperti *subject*, nama pengirim, email pengirim, dan email balasan. Selain itu, pengguna juga dapat memilih *template* yang ingin digunakan serta melakukan pengeditan pada *template* tersebut sesuai kebutuhan. Pada bagian kedua (*Select Recipient*), pengguna dapat mengisi detail informasi penerima *newsletter*, dengan dukungan untuk lebih dari satu penerima. Kemudian, pada bagian ketiga (*Preview and Send Newsletter*), pengguna dapat melihat *preview* dari *newsletter* yang telah dibuat, meninjau ulang informasi pengiriman yang telah diisi sebelumnya, serta menentukan metode pengiriman, yaitu mengirim langsung atau menjadwalkan pengiriman. Selain itu, tersedia juga fitur untuk mengirim email percobaan ke alamat yang ditentukan oleh pengguna.

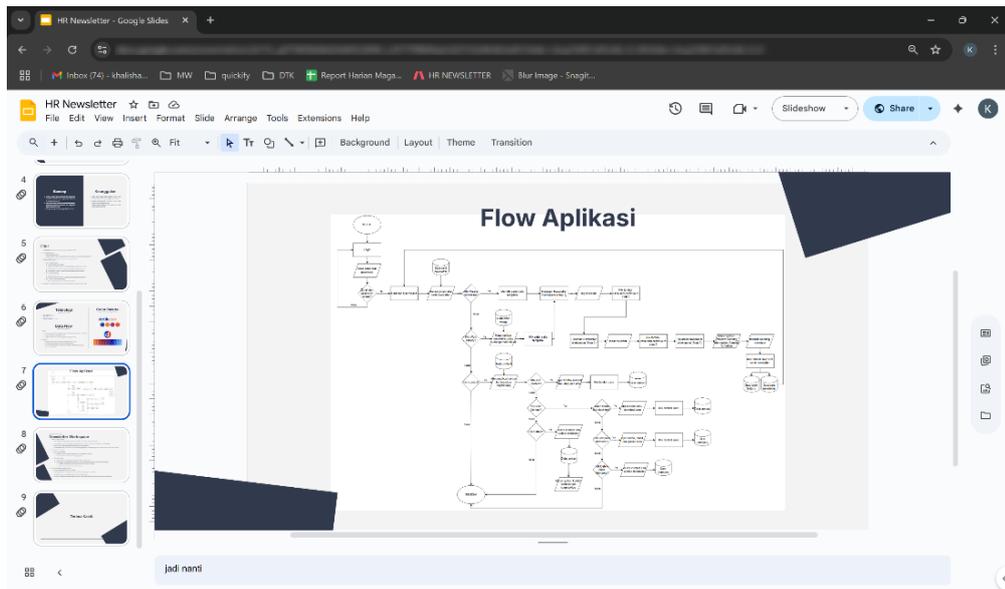
Keempat, terdapat fitur *Work History*, yaitu fitur yang menampilkan seluruh *newsletter* yang telah dibuat oleh pengguna. Fitur ini memiliki tampilan yang mirip dengan *Create Newsletter*, yang menggunakan *grid* untuk memudahkan navigasi dalam aplikasi. Perbedaannya terletak pada konten yang ditampilkan, di mana halaman ini hanya menampilkan *template* yang telah digunakan dan dimodifikasi oleh pengguna, yang kemudian dapat dimodifikasi kembali sesuai dengan kebutuhan pengguna.

Terakhir, terdapat fitur *Contact*, yaitu fitur yang menampilkan daftar kontak yang dikelompokkan berdasarkan segmen atau *tag*. Daftar kontak ini disajikan dalam bentuk tabel, dengan informasi mengenai nama kontak, deskripsi kontak, jumlah anggota yang dimiliki, serta ikon untuk menghapus atau mengedit kontak tersebut. Pada halaman ini, pengguna dapat mengelola kontak dengan cara menambahkan, memodifikasi, atau menghapus kontak beserta anggotanya, atau hanya mengelola anggota di dalam kontak tersebut.



Gambar 3.5 Presentasi Hasil Pembuatan Konsep dan Fitur Aplikasi

Seluruh konsep dan fitur yang telah dirancang dirangkum ke dalam presentasi menggunakan Google Slides, seperti ditunjukkan pada Gambar 3.4 dan Gambar 3.5. Presentasi tersebut juga mencakup teknologi yang akan digunakan, alur data (*data flow*), palet warna (*color palette*), serta alur aplikasi (*flow aplikasi*) yang akan dikembangkan. Diagram alur aplikasi (*flow aplikasi*) HR *Newsletter* ditampilkan dalam Gambar 3.6.



Gambar 3.6 *Flow Aplikasi HR Newsletter*

Presentasi ini dipaparkan kepada tim HR, supervisor, dan mentor magang sebagai bagian dari proses evaluasi. Melalui presentasi yang dilakukan, diperoleh berbagai masukan dan tanggapan yang berguna untuk menyempurnakan rancangan. Hasil diskusi tersebut menjadi dasar untuk memperoleh persetujuan dan melanjutkan ke tahap pengembangan berikutnya.

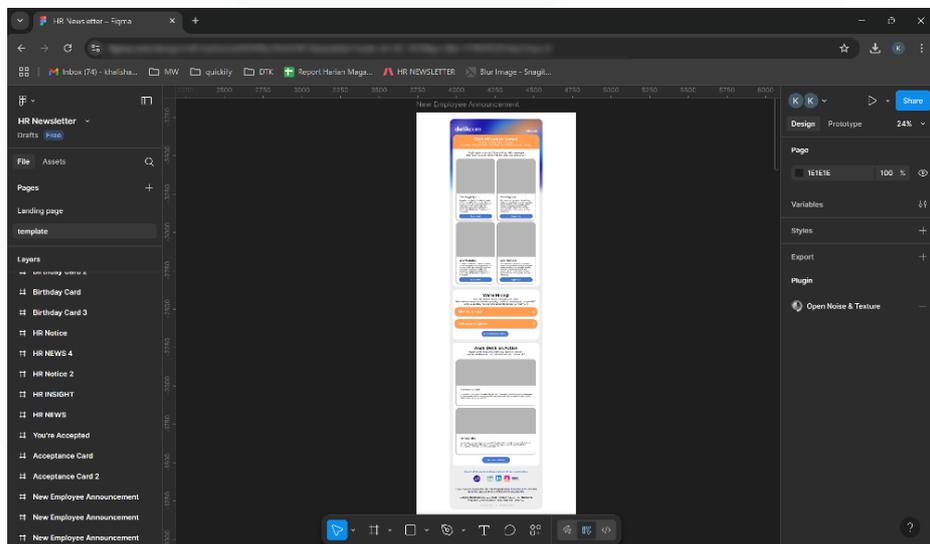
3.2.3 Pembuatan Desain *Template Newsletter* dan Prototipe Aplikasi *Website*

Setelah dilakukan riset, analisis, perumusan konsep dan fitur, serta presentasi kepada tim HR, langkah selanjutnya adalah pembuatan desain *template newsletter* dan prototipe aplikasi. Proses perancangan ini dilakukan menggunakan alat desain berbasis web, yaitu Figma. Pemilihan Figma didasarkan pada fakta bahwa aplikasi ini merupakan salah satu *tools* yang umum digunakan di perusahaan detikcom. Selain itu, Figma menyediakan berbagai fitur yang memudahkan dalam pembuatan *template* dan prototipe, serta mendukung akses mudah terhadap ikon, simbol, dan penyusunan alur (*flow*) dalam pengembangan aplikasi web.

3.2.3.1 Desain *Template Newsletter*

Template newsletter yang dibuat dirancang agar dapat digunakan dan disesuaikan oleh pengguna sesuai dengan kebutuhan dan kreativitas mereka, termasuk pengeditan *layout* dan isi. Setelah mencari referensi dan mendapatkan contoh dari tim HR, dilakukan perancangan sebanyak 13 *template newsletter*. *Template* tersebut dibagi ke dalam enam kategori, yaitu *Welcoming Card*, *Acceptance Card*, *HR Notice*, *HR Insight*, *HR News*, dan *Birthday Card*.

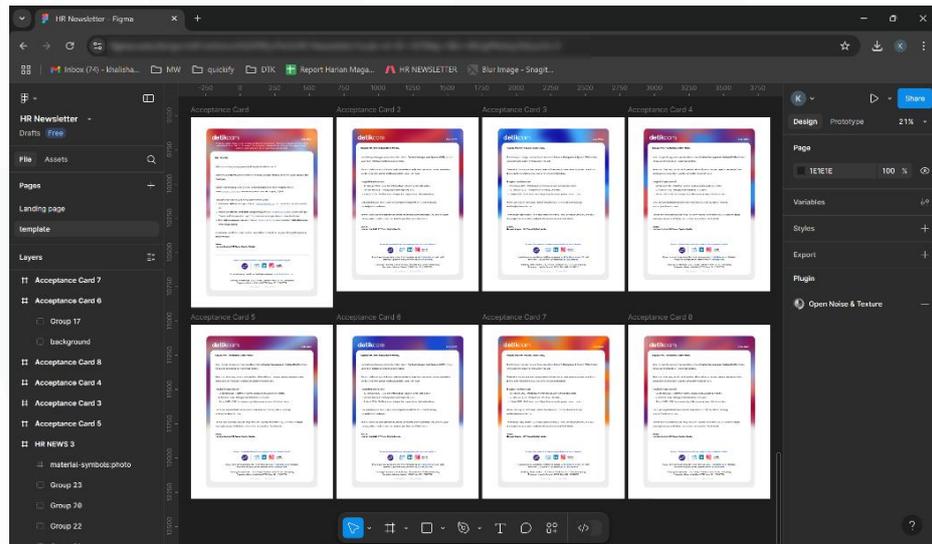
Pertama, terdapat *template Welcoming Card*, yaitu *newsletter* yang berfungsi untuk menyambut karyawan baru dan dapat dikirimkan kepada seluruh karyawan detikcom. *Template* ini dilengkapi dengan foto, nama, dan deskripsi singkat mengenai karyawan baru tersebut. Selain itu, *newsletter* ini juga memuat informasi tambahan seperti lowongan kerja di detikcom, akun media sosial resmi, serta berbagai aktivitas yang dilakukan di lingkungan detikcom. Gambar 3.7 menunjukkan desain *Welcoming Card* yang dibuat menggunakan Figma.



Gambar 3.7 Desain *Welcoming Card* pada Figma

Kedua, terdapat *template Acceptance Card*, yaitu *newsletter* yang berfungsi untuk menginformasikan penerimaan karyawan baru dan dikirimkan kepada calon karyawan yang diterima di detikcom. *Template* ini memuat ucapan selamat, tanggal mulai bekerja, serta beberapa

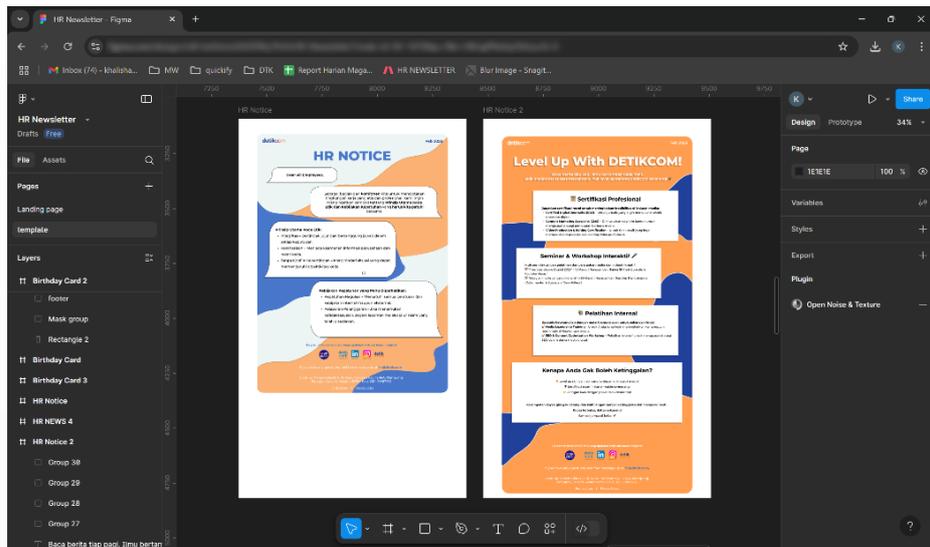
ketentuan atau persyaratan yang harus dipatuhi selama masa kerja. *Acceptance Card* ini memiliki dua tipe dengan total delapan desain latar belakang yang dirancang menggunakan Figma, seperti ditampilkan pada Gambar 3.8.



Gambar 3.8 Desain *Acceptance Card* pada Figma

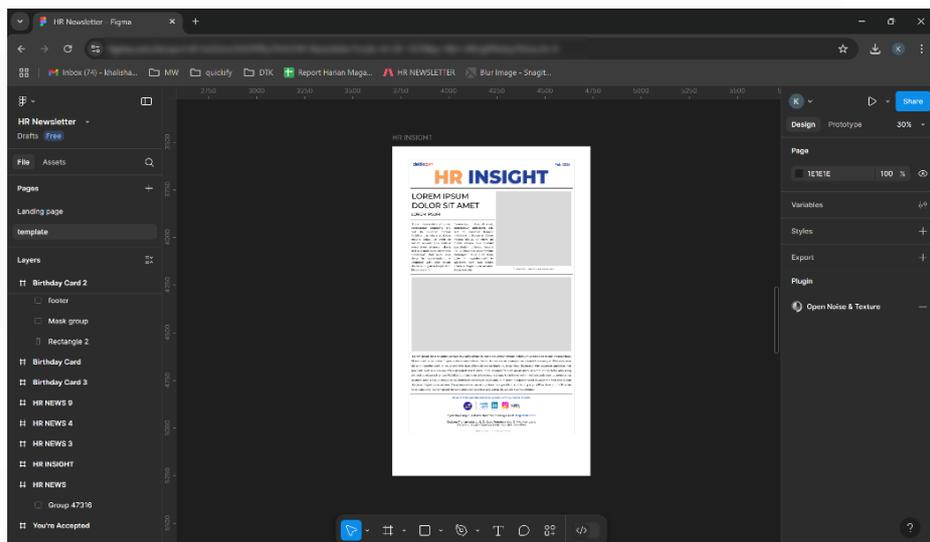
Ketiga, terdapat *template* HR Notice, yaitu *newsletter* yang berfungsi untuk menyampaikan informasi terkait pengumuman terbaru di perusahaan. *Template* ini dirancang dengan tampilan yang menarik agar penerima tertarik untuk membaca setiap informasi yang disampaikan oleh tim HR. HR Notice memiliki dua tipe desain dengan latar belakang dan tampilan yang berbeda, seperti yang ditampilkan pada Gambar 3.9.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



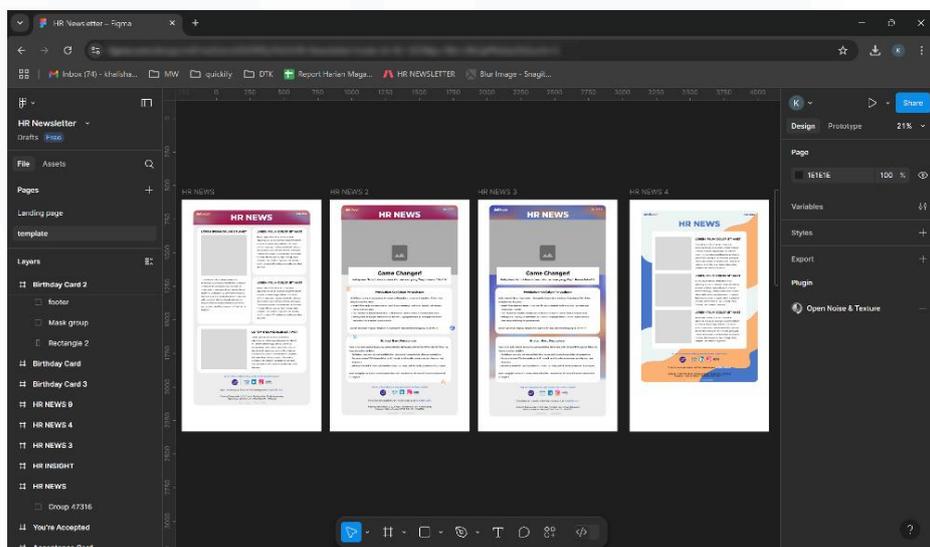
Gambar 3.9 Desain HR Notice pada Figma

Keempat, terdapat *template HR Insight*, yaitu *newsletter* yang berfungsi untuk membagikan *insight* atau wawasan terbaru seputar perusahaan. *Template* ini dilengkapi dengan teks dan gambar yang disusun secara menarik agar mudah dibaca oleh penerima *newsletter*. *HR Insight* hanya memiliki satu tipe desain dengan gaya tampilan yang unik, seperti yang ditunjukkan pada Gambar 3.10.



Gambar 3.10 Desain HR Insight pada Figma

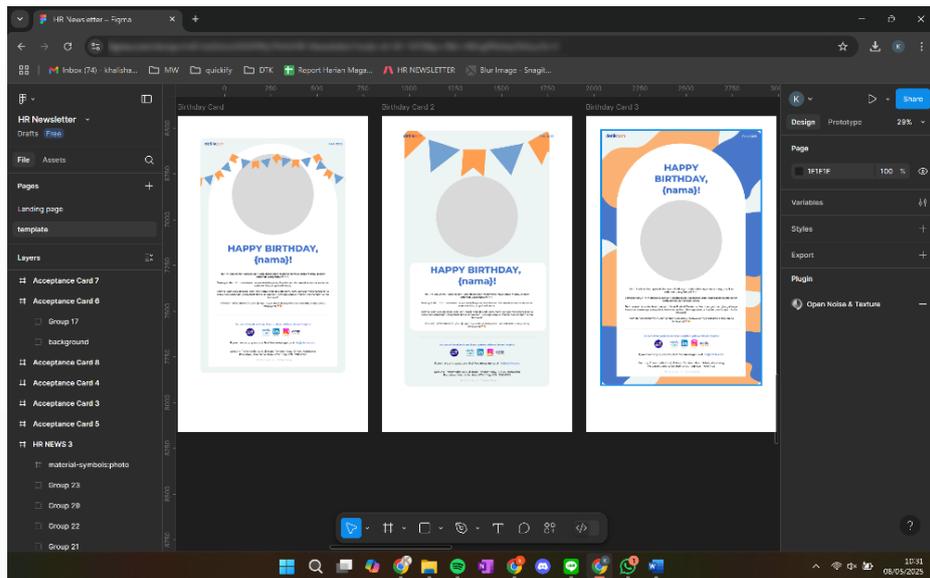
Kelima, terdapat *template HR News*, yaitu *newsletter* yang memiliki fungsi serupa dengan *HR Notice*, yaitu untuk menyampaikan berita terbaru mengenai perusahaan. *Template* ini dilengkapi dengan teks, gambar, dan dekorasi yang dirancang untuk membuatnya menarik dan mudah dibaca oleh penerima *newsletter*. *HR News* memiliki empat tipe desain dengan tampilan, latar belakang, dan gaya yang berbeda, seperti yang ditunjukkan pada Gambar 3.11.



Gambar 3.11 Desain *HR News* pada Figma

Keenam, terdapat *template Birthday Card*, yaitu *newsletter* yang berfungsi untuk menyampaikan ucapan selamat ulang tahun beserta harapan dan doa yang menyertainya. *Template* ini dilengkapi dengan gambar karyawan yang berulang tahun, teks ucapan, harapan, dan doa, serta dekorasi yang menarik. *Birthday Card* memiliki tiga tipe desain dengan tampilan dan latar belakang yang berbeda-beda, seperti yang ditunjukkan pada Gambar 3.12.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

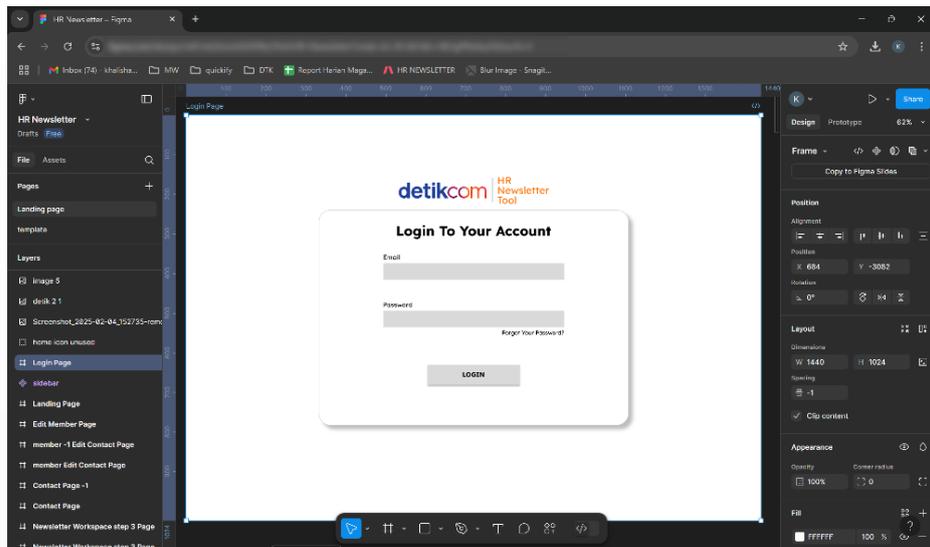


Gambar 3.12 Desain *Birthday Card* pada Figma

3.2.3.2 Desain Prototipe Aplikasi *Website HR Newsletter*

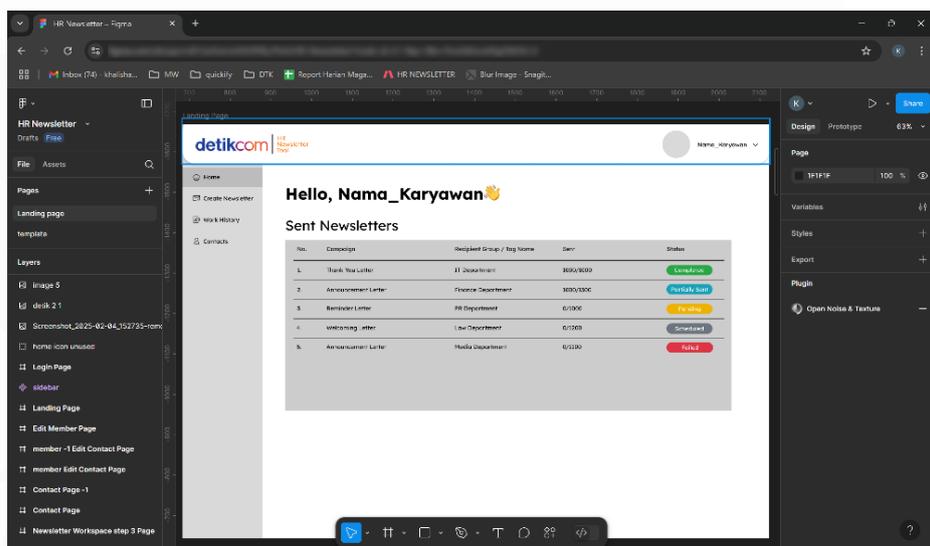
Prototipe aplikasi *website HR Newsletter* dibuat untuk mempermudah pengembang aplikasi dan tim HR dalam menggambarkan aplikasi yang ingin dibuat. Prototipe ini akan digunakan sebagai referensi dan panduan dalam pengembangan aplikasi *website HR Newsletter*. Prototipe ini mencakup enam halaman, terdiri atas satu halaman *Login* dan lima halaman fitur utama yang telah dijelaskan sebelumnya, yaitu *Dashboard*, *Create Newsletter*, *Newsletter Workspace*, *Work History*, dan *Contacts*.

Pertama, halaman *Login* terdiri dari satu halaman yang sederhana. Halaman ini berisi logo perusahaan, nama *website* aplikasi, serta kotak yang menampilkan *textbox* untuk pengguna memasukkan email dan *password*, dan tombol untuk *forgot password* serta *login*. Gambar 3.13 menunjukkan tampilan desain halaman *login* yang dibuat menggunakan Figma.



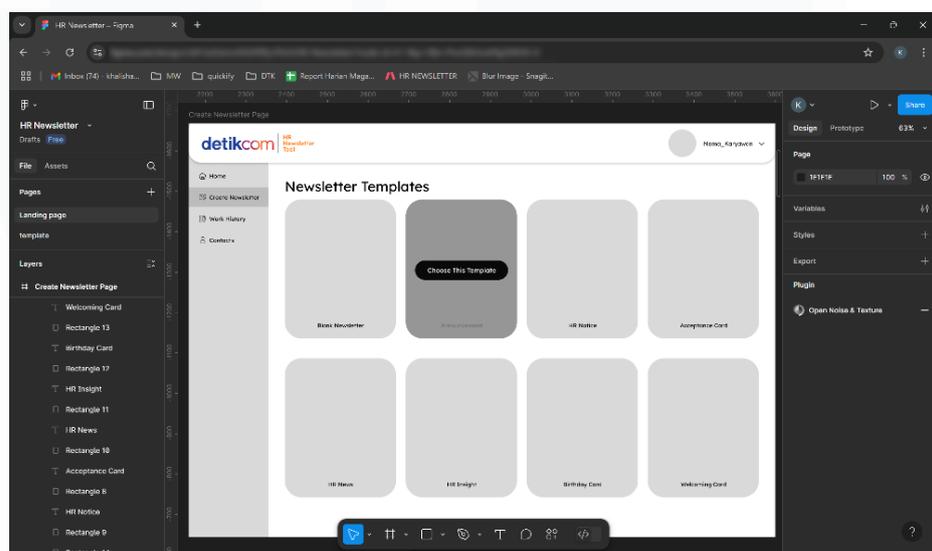
Gambar 3.13 Desain Halaman *Login* pada Figma

Kedua, halaman *Dashboard* terdiri atas satu halaman yang menampilkan tabel berisi informasi mengenai *newsletter* yang telah dikirim. Tabel ini mencakup nama *newsletter*, tanggal pengiriman, penerima, jumlah yang sudah terkirim, serta status pengiriman. Gambar 3.14 menunjukkan tampilan desain halaman *Dashboard* yang dibuat menggunakan Figma.



Gambar 3.14 Desain Halaman *Dashboard* pada Figma

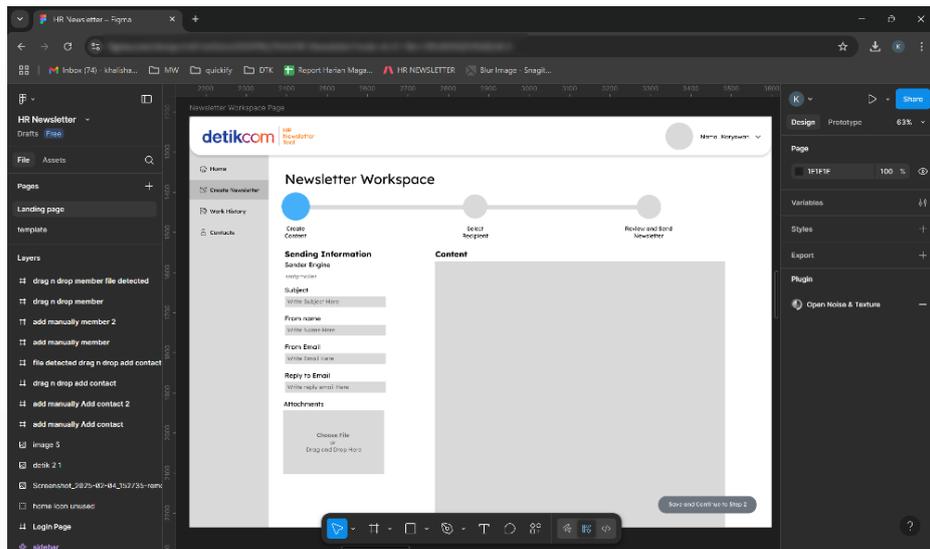
Ketiga, halaman *Create Newsletter* terdiri atas satu halaman yang menampilkan pilihan *template newsletter* dalam bentuk *grid*. Halaman ini dilengkapi dengan gambar *preview* dari setiap *template* untuk mempermudah pengguna dalam memilih secara visual. Saat pengguna mengarahkan kursor ke salah satu *template* (*hover*), akan muncul tombol '*Choose this template*' di atas gambar *preview*, seperti ditunjukkan pada Gambar 3.15.



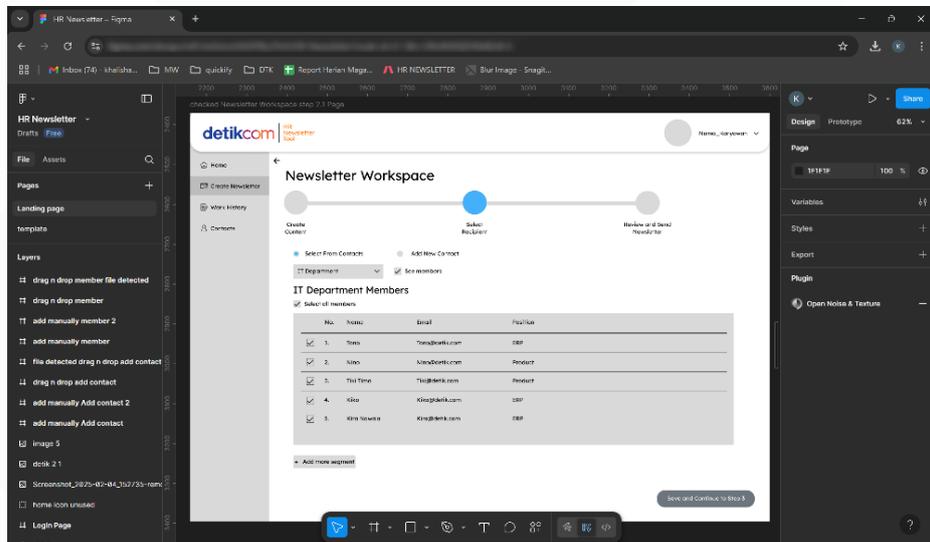
Gambar 3.15 Desain Halaman *Create Newsletter* pada Figma

Keempat, halaman *Newsletter Workspace* terdiri atas tiga bagian utama. Bagian pertama, yaitu *Create Content*, digunakan untuk membuat dan memodifikasi isi *newsletter* serta mengisi informasi terkait pengiriman. Gambar 3.16 menunjukkan desain bagian pertama (*Create Content*) yang dibuat menggunakan Figma.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

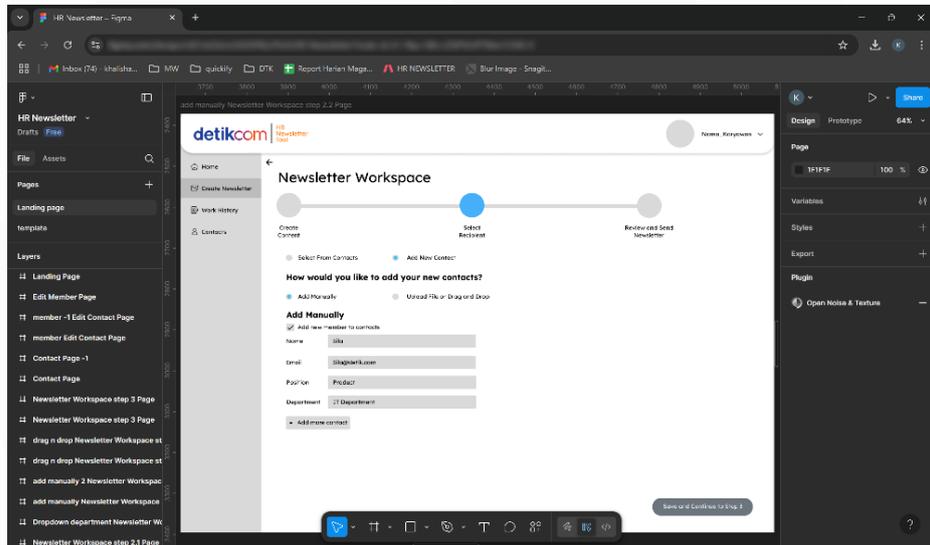


Gambar 3.16 Desain Bagian Pertama (*Create Content*) dari *Newsletter Workspace* pada Figma
 Bagian kedua, yaitu *Select Recipient*, berfungsi untuk memilih penerima *newsletter*, baik dari daftar kontak yang sudah tersedia, seperti terlihat pada Gambar 3.17, maupun dengan menambahkan kontak baru.

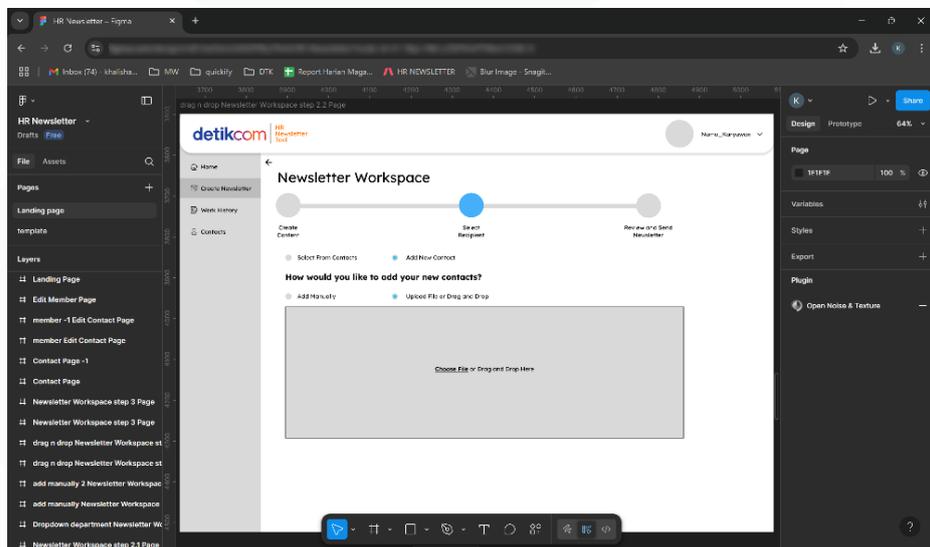


Gambar 3.17 Desain Bagian Kedua (*Select Recipient*) dari *Newsletter Workspace* dengan Pemilihan *Recipient* dari Daftar Kontak pada Figma

Penambahan kontak baru dapat dilakukan melalui metode input manual, seperti terlihat pada Gambar 3.18, atau menggunakan metode *drag and drop*, seperti terlihat pada Gambar 3.19.



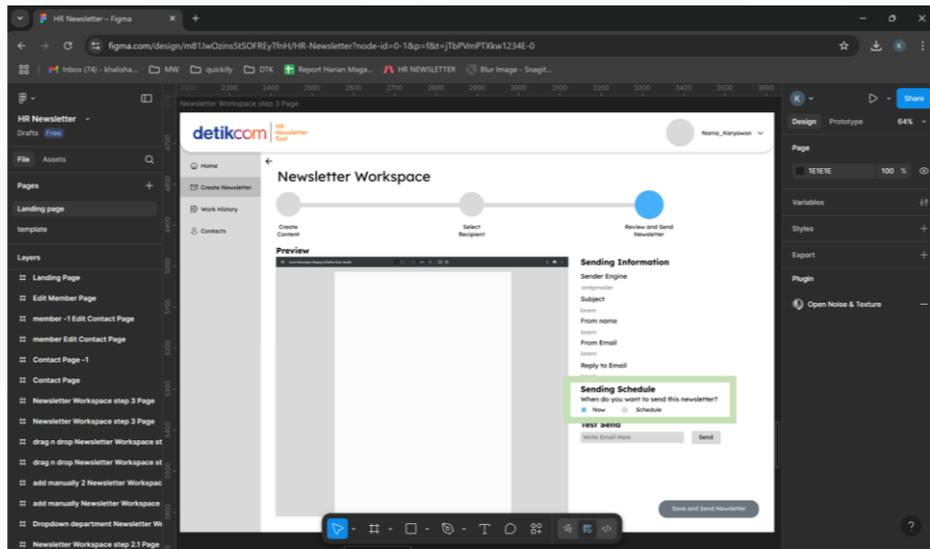
Gambar 3.18 Desain Bagian Kedua (*Select Recipient*) dari *Newsletter Workspace* dengan Pemilihan *Recipient* melalui Penambahan Kontak Baru secara Manual pada Figma



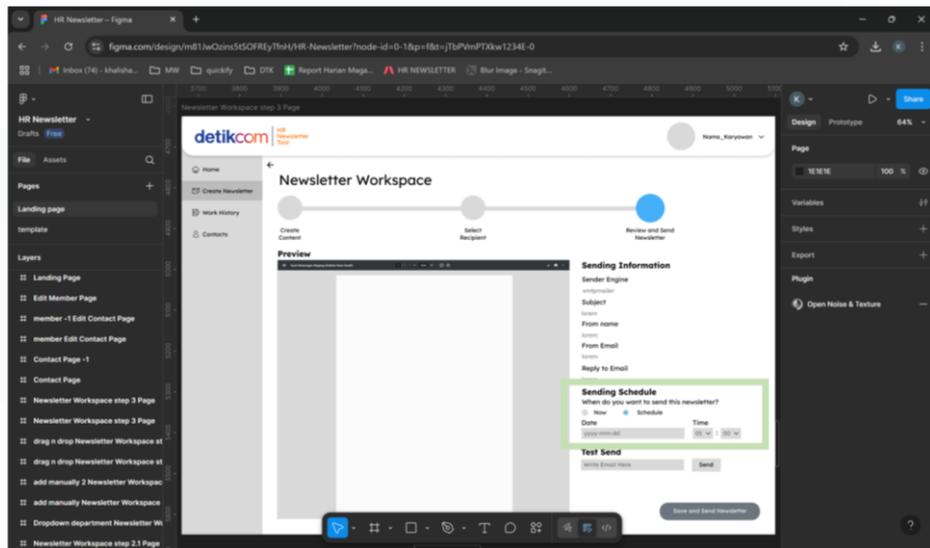
Gambar 3.19 Desain Bagian Kedua (*Select Recipient*) dari *Newsletter Workspace* dengan Pemilihan *Recipient* melalui Penambahan Kontak Baru secara *Drag and Drop* pada Figma

Bagian ketiga, yaitu *Preview and Send Newsletter*, memungkinkan pengguna untuk melihat pratinjau hasil *newsletter* yang telah dibuat, meninjau ulang informasi pengiriman, serta melakukan uji coba

pengiriman. Bagian ini juga dilengkapi dengan dua opsi pengiriman, yaitu mengirim secara langsung, seperti ditunjukkan pada Gambar 3.20, atau menjadwalkan pengiriman untuk waktu tertentu di kemudian hari, sebagaimana terlihat pada Gambar 3.21.



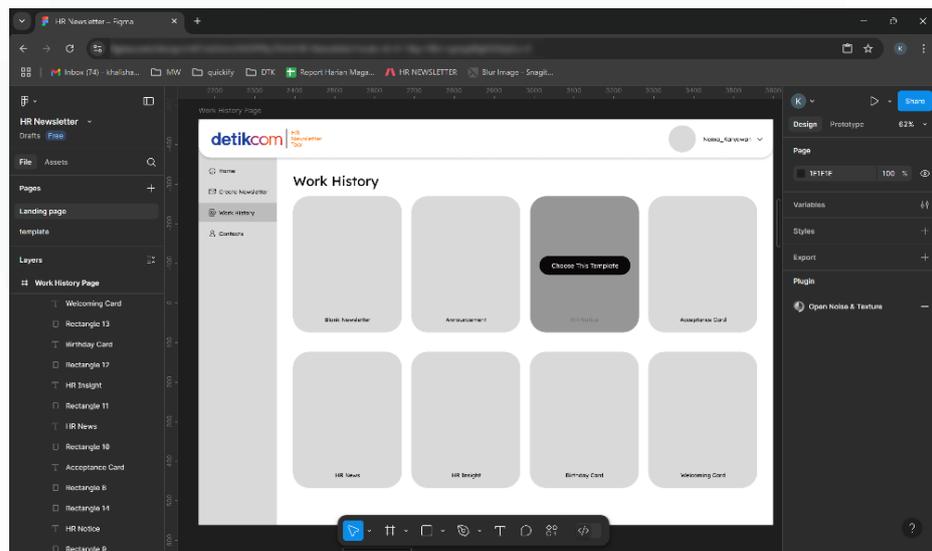
Gambar 3.20 Desain Bagian Ketiga (*Preview and Send Newsletter*) dari *Newsletter Workspace* dengan Opsi Pengiriman *Newsletter* secara Langsung pada Figma



Gambar 3.21 Desain Bagian Ketiga (*Preview and Send Newsletter*) dari *Newsletter Workspace* dengan Opsi Menjadwalkan Pengiriman *Newsletter* pada Figma

Kelima, halaman *Work History* terdiri atas satu halaman yang menampilkan gambar *preview template newsletter* dalam bentuk *grid*,

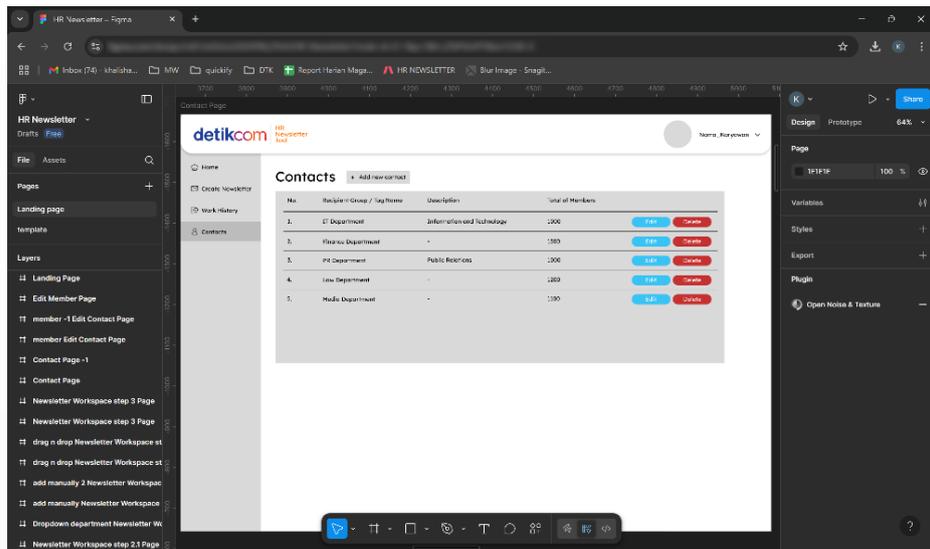
yang telah dimodifikasi oleh pengguna sebelumnya. Fungsionalitas halaman ini serupa dengan halaman *Create Newsletter*, di mana ketika pengguna mengarahkan kursor ke salah satu *template* (*hover*), akan muncul tombol '*Choose this template*' di atas gambar pratinjau. Gambar 3.22 menunjukkan desain halaman *Work History* yang dibuat menggunakan Figma.



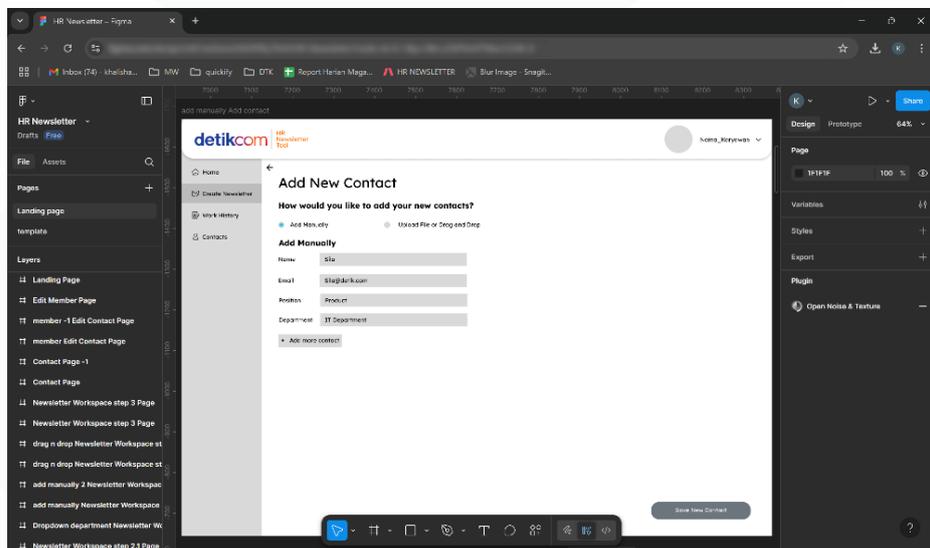
Gambar 3.22 Desain Halaman *Work History* pada Figma

Keenam, halaman *Contacts* terdiri atas lima bagian. Bagian pertama adalah *Contact List* yang menampilkan daftar kontak dalam bentuk tabel, menampilkan informasi seperti nama kontak, deskripsi kontak, jumlah anggota, serta ikon untuk mengedit atau menghapus kontak. Pada halaman ini juga terdapat tombol untuk menambahkan kontak baru, seperti yang ditunjukkan pada Gambar 3.23.

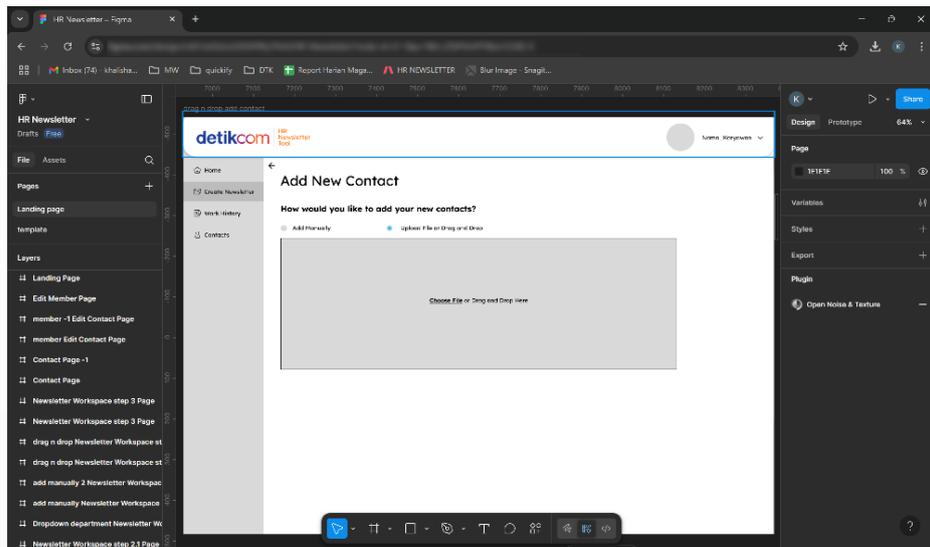
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.23 Desain Bagian Pertama (*Contact List*) dari Halaman *Contacts* pada Figma
 Bagian kedua adalah *Add Contact*, yang menampilkan dua opsi penambahan kontak, yaitu secara input manual atau melalui metode *drag and drop*, seperti terlihat pada Gambar 3.24 dan Gambar 3.25.

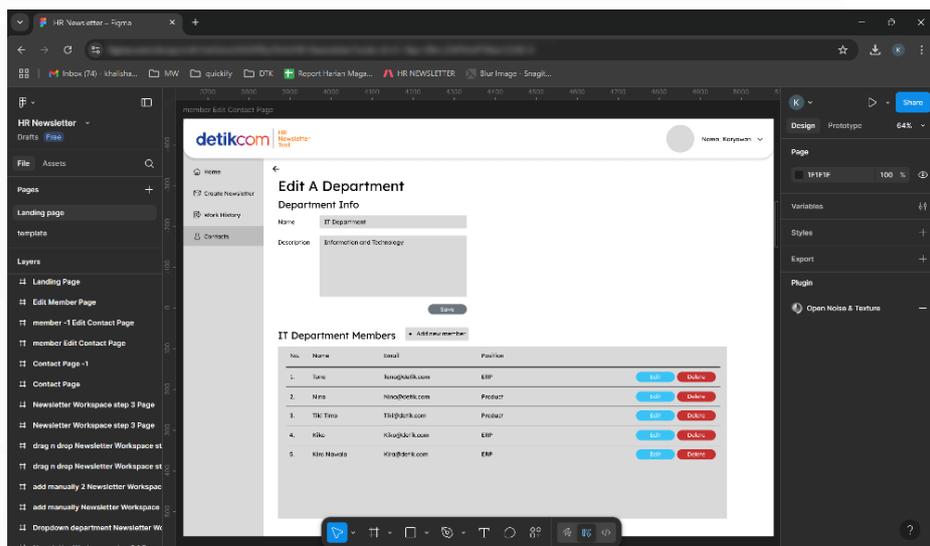


Gambar 3.24 Desain Bagian Kedua (*Add Contact*) dari Halaman *Contacts* melalui Input Manual pada Figma



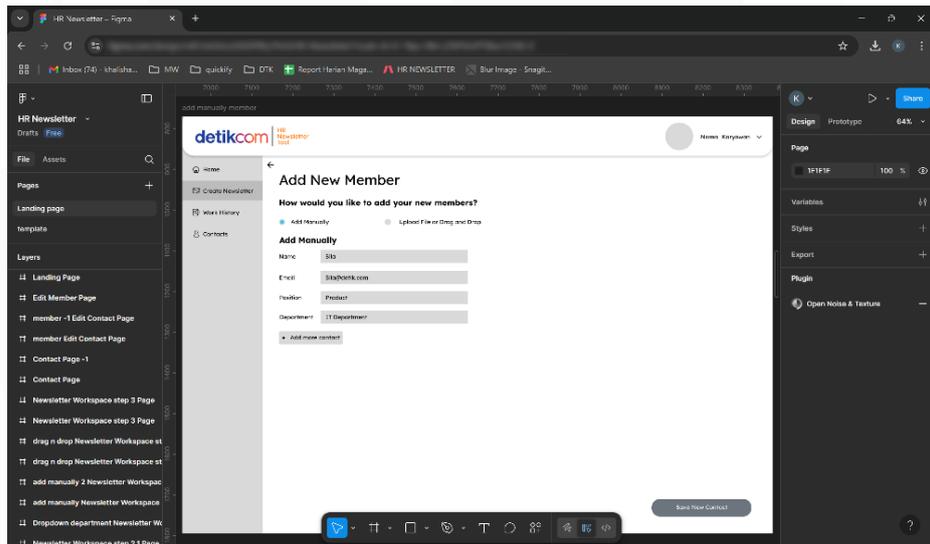
Gambar 3.25 Desain Bagian Kedua (*Add Contact*) dari Halaman *Contacts* melalui *Drag and Drop* pada Figma

Bagian ketiga adalah *Edit Contact*, yang menampilkan *textbox* untuk mengubah nama dan deskripsi kontak. Selain itu, terdapat tabel berisi daftar anggota dalam kontak tersebut yang mencakup informasi nama, email, posisi, serta ikon untuk mengedit atau menghapus anggota. Terdapat pula tombol untuk menambahkan anggota baru, seperti yang ditunjukkan pada Gambar 3.26.

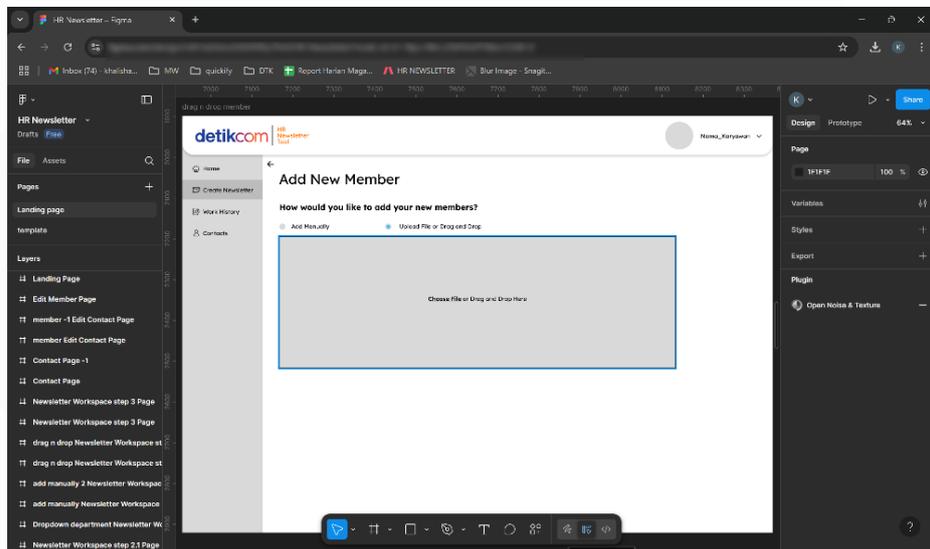


Gambar 3.26 Desain Bagian Ketiga (*Edit Contact*) dari Halaman *Contacts* pada Figma

Bagian keempat adalah *Add Member*, yang memiliki tampilan dan fungsionalitas serupa dengan halaman *Add Contact*, yaitu memberikan dua pilihan penambahan anggota: secara input manual atau dengan *drag and drop*, seperti pada Gambar 3.27 dan Gambar 3.28.



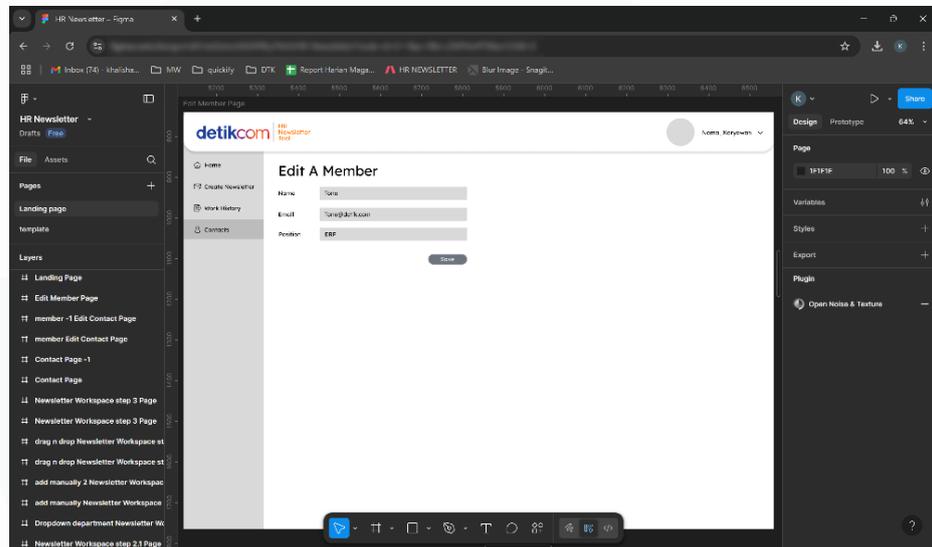
Gambar 3.27 Desain Bagian Keempat (*Add Member*) dari Halaman *Contacts* melalui Input Manual pada Figma



Gambar 3.28 Desain Bagian Keempat (*Add Member*) dari Halaman *Contacts* melalui *Drag and Drop* pada Figma

Bagian kelima adalah *Edit Member*, yaitu halaman yang memungkinkan pengguna untuk mengubah informasi nama, posisi, dan email dari

anggota dalam suatu kontak, sebagaimana ditampilkan pada Gambar 3.29.



Gambar 3.29 Desain Bagian Kelima (Edit Member) dari Halaman Contacts pada Figma

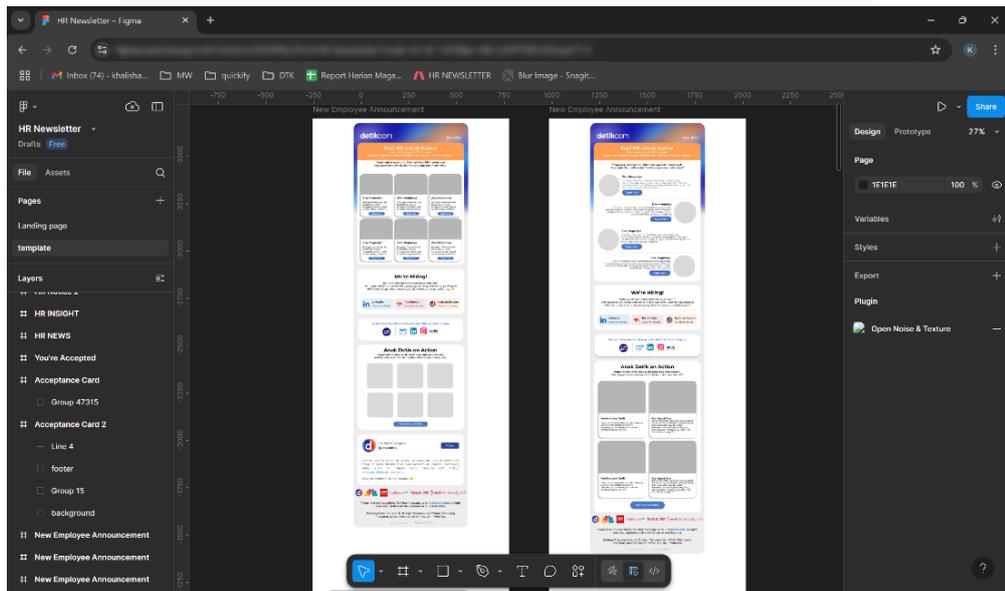
3.2.4 Review dan Finalisasi Desain

Setelah proses pembuatan desain *template newsletter* dan prototipe aplikasi *website* selesai dilakukan, kedua hasil tersebut dipresentasikan kepada tim HR untuk mendapatkan persetujuan dan dapat dilanjutkan ke tahap pengembangan. Presentasi ini bertujuan untuk mendapatkan peninjauan ulang serta evaluasi dari tim HR terhadap desain *template* dan prototipe yang telah dibuat, sehingga dapat dilakukan finalisasi desain dengan mempertimbangkan masukan yang diberikan. Selain itu, presentasi ini juga menjadi kesempatan untuk memastikan bahwa desain yang dibuat sesuai dengan standar dan preferensi tim HR.

Pada saat presentasi, tim HR meninjau hasil desain dan memberikan beberapa masukan. Pertama, untuk desain *template*, tim HR mengusulkan penambahan satu *template welcome card* yang serupa dengan desain sebelumnya yang pernah dibuat oleh UI Developer. Kedua, pada prototipe aplikasi *website*, tim HR menginginkan agar di halaman *newsletter workspace* disediakan tombol untuk memilih *template* yang ingin digunakan. Selain itu,

pada fitur *newsletter workspace*, tim HR juga mengusulkan agar wadah konten dapat dibuat *drag and drop* untuk memudahkan pengguna dalam menyusun isi konten.

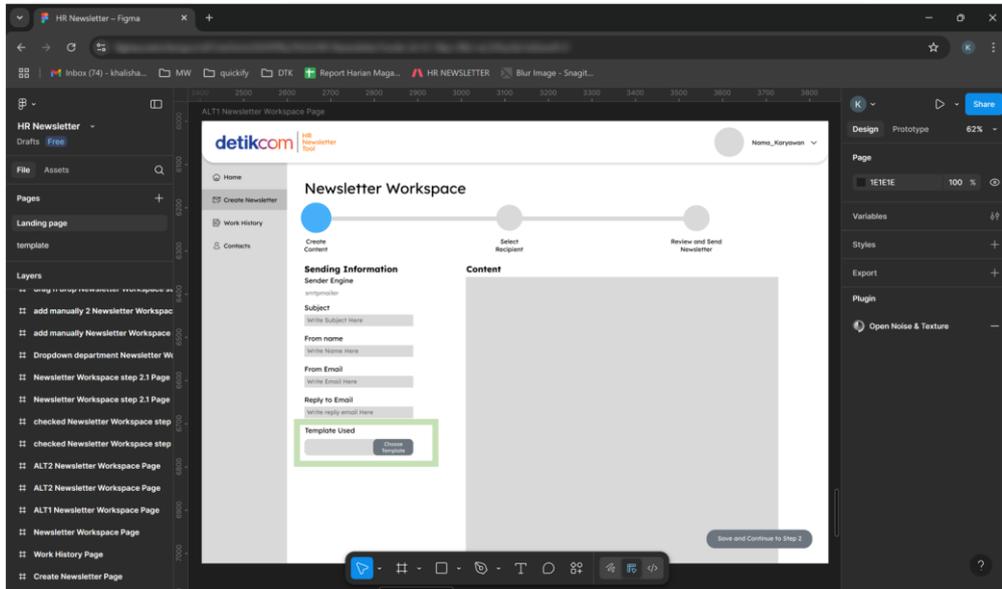
Menindaklanjuti masukan tersebut, dua tambahan *template welcome card* dibuat dengan desain yang serupa dengan versi sebelumnya. *Template* tambahan tersebut dapat dilihat pada Gambar 3.30.



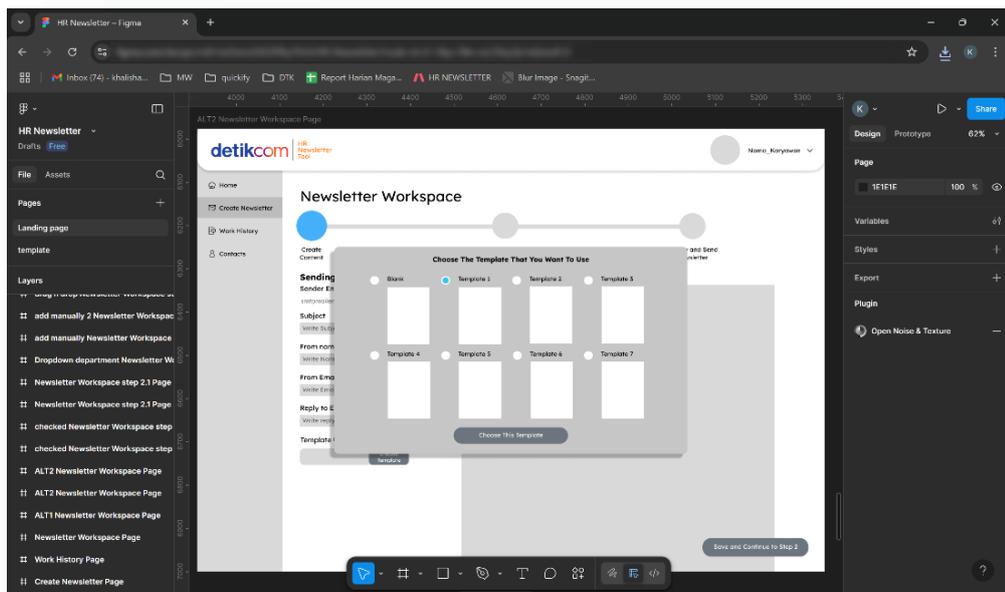
Gambar 3.30 Desain Tambahan *Welcoming Card* pada Figma

Untuk prototipe aplikasi *website*, revisi dilakukan dengan menambahkan tombol pemilih *template*, seperti yang ditampilkan pada Gambar 3.31. Ketika tombol tersebut diklik, sebuah *pop-up* akan muncul menampilkan beberapa pilihan *template* yang tersedia, sebagaimana ditunjukkan pada Gambar 3.32. Pengguna kemudian dapat memilih salah satu *template*, yang akan ditampilkan dalam kotak editor untuk memungkinkan pengguna mengedit isi kontennya.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.31 Tombol Pemilih *Template* pada *Newsletter Workspace*



Gambar 3.32 *Pop-up* Beberapa Pilihan *Template* yang Tersedia pada *Newsletter Workspace*

Sebagai tindak lanjut atas usulan tim HR, dilakukan riset untuk menindaklanjuti usulan dari tim HR terkait pembuatan wadah konten yang mendukung fitur *drag and drop*, untuk memudahkan pengguna dalam menyusun isi konten. Untuk itu, dibutuhkan sebuah WYSIWYG editor (*What You See Is What You Get*) yang mendukung antarmuka visual interaktif. Setelah dilakukan riset, ditemukan beberapa solusi potensial, seperti

Summernote, GrapesJS, TinyMCE, CKEditor, Craft.js, dan Reka.js. Tabel 3.2 menunjukkan perbandingan dari keenam WYSIWYG editor tersebut.



Tabel 3.2 Perbandingan beberapa WYSIWYG Editor

Kriteria	Summernote	GrapesJS	TinyMCE	CKEditor	Craft.js	Reka.js
<i>Open Source</i>	Ya	Ya	Ya	Ya	Ya	Ya
Fitur <i>Drag and Drop</i>	Terbatas	Ya	Terbatas	Terbatas	Ya	Ya
Mudah Diimplementasikan (API)	Ya	Ya	Ya	Ya	Tidak	Tidak
<i>Custom</i> Komponen	Terbatas	Ya	Ya	Ya	Ya	Ya
Dukungan & Dokumentasi	Cukup	Lengkap	Lengkap	Lengkap	Terbatas	Terbatas
Fitur Pembuatan & Pengiriman <i>Newsletter</i>	Dasar (teks & gambar)	Lengkap (<i>builder & export HTML</i>)	Dasar	Dasar (butuh <i>plugin</i>)	Tidak tersedia langsung	Tidak tersedia langsung

(Sumber olahan peneliti, 2025)

Berdasarkan hasil perbandingan pada Tabel 3, GrapesJS dipilih sebagai solusi yang paling sesuai dengan kebutuhan aplikasi HR *Newsletter* dan preferensi tim HR. Dengan demikian, GrapesJS kemudian diimplementasikan dalam pengembangan *frontend* dan *backend* aplikasi *website* sebagai langkah nyata dari hasil riset yang telah dilakukan.

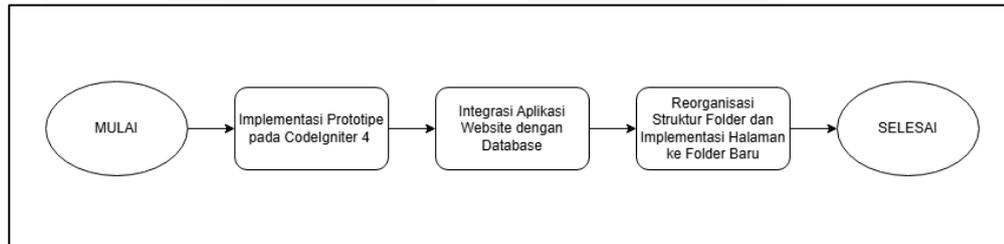
3.2.5 Pengembangan *Frontend* dan *Backend* Aplikasi *Website*

Setelah dilakukan proses *review* dan finalisasi desain, proses dilanjutkan ke tahap berikutnya, yaitu pengembangan *frontend* dan *backend* aplikasi *website*. Pengembangan *frontend* mencakup pembuatan tampilan antarmuka pengguna seperti halaman *login*, *dashboard*, dan halaman fungsional lainnya, serta memastikan tampilan responsif di berbagai perangkat. Sementara itu, pengembangan *backend* mencakup pengelolaan *database*, pembuatan logika bisnis, pengaturan *routing*, serta integrasi antara *frontend* dan *database* menggunakan API. Fungsi-fungsi seperti autentikasi pengguna, validasi data, dan operasi CRUD (*Create, Read, Update, Delete*) juga diterapkan.

Pengembangan *frontend* dan *backend* dilaksanakan menggunakan *framework* PHP, yaitu CodeIgniter 4, sesuai arahan dari mentor di detikcom. Pada awalnya, *database* yang digunakan adalah PostgreSQL sehingga perlu mengunduh dan menggunakan pgAdmin 4 sebagai aplikasi pengelola *database* sesuai instruksi mentor. Namun, saat proses pemindahan kode ke folder baru yang diberikan mentor, *database* beralih ke MySQL, sehingga dilakukan penyesuaian.

Seiring dengan penggunaan CodeIgniter 4, dilakukan pengunduhan beberapa *tools* pendukung yang umum digunakan di perusahaan detikcom, antara lain Visual Studio Code (VSCode) sebagai *code* editor, Composer untuk manajemen dependensi, Git Bash untuk menjalankan dan mengelola aplikasi, serta XAMPP sebagai server lokal untuk pengujian aplikasi. Namun, penggunaan Git Bash kemudian tidak dilanjutkan dan digantikan dengan

terminal bawaan VSCode sesuai arahan mentor. Gambar 3.33 menunjukkan *flowchart* alur kerja pengembangan *frontend* dan *backend* pada aplikasi *website* ini.



Gambar 3.33 *Flowchart* Pengembangan *Frontend* dan *Backend* Aplikasi *Website* HR *Newsletter*

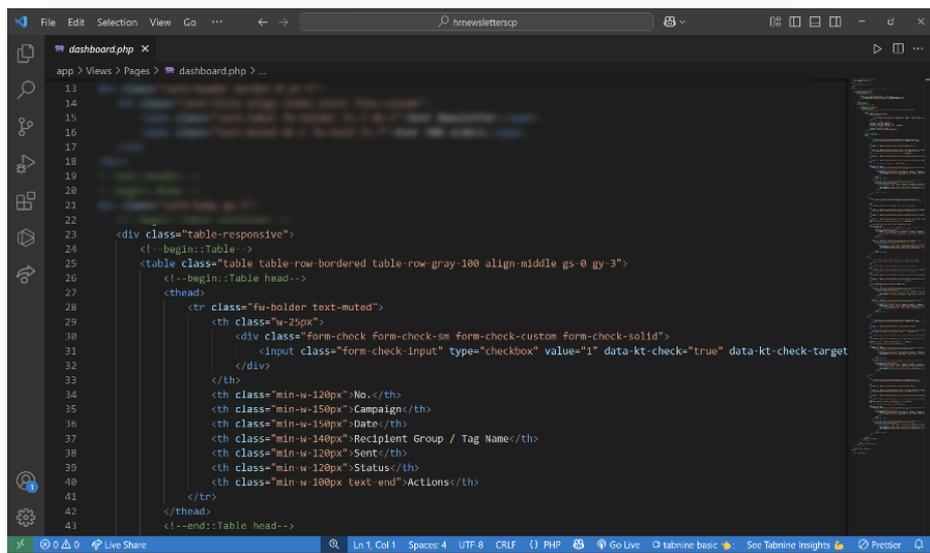
3.2.5.1 Implementasi Prototipe pada CodeIgniter 4

Implementasi prototipe dilakukan dengan menggunakan *framework* CodeIgniter 4 yang mendukung arsitektur *Model-View-Controller* (MVC) untuk memudahkan pengembangan aplikasi secara terstruktur. Tahap awal dimulai dengan instalasi CodeIgniter 4 menggunakan Composer, dilanjutkan dengan pembuatan struktur proyek dan konfigurasi dasar. Proses implementasi mencakup pembuatan halaman *Dashboard*, *Create Newsletter*, *Newsletter Workspace*, *Work History*, *Contacts*, dan *Login* sesuai dengan desain prototipe yang telah disusun. Dalam proses pengembangan, *framework* Metronic digunakan sebagai *template* antarmuka pengguna (*User Interface/UI*) untuk mempermudah pembuatan desain yang konsisten, responsif, dan modern. Pemilihan *framework* ini dilakukan berdasarkan arahan mentor untuk memastikan kualitas tampilan serta efisiensi dalam pengembangan.

3.2.5.1.1 Halaman *Dashboard*

Pertama, halaman *Dashboard* dirancang untuk menampilkan tabel berisi daftar *newsletter* yang telah dikirim oleh pengguna. Tabel ini dibuat menggunakan elemen `<table>` yang dibungkus dalam div dengan *class* `table-responsive`, sehingga memungkinkan tampilan tabel dapat di *scroll* secara horizontal jika tampilan tidak muat pada layar. Bagian

kepala tabel menggunakan elemen `<thead>` dengan tujuh kolom utama, yaitu kolom *checkbox* untuk memilih baris, nomor, nama *campaign*, tanggal pengiriman, grup penerima atau nama *tag*, jumlah *newsletter* yang telah terkirim, status pengiriman, dan kolom aksi. Seluruh kolom pada kepala tabel ditata dengan *class* `min-w-*` dari bootstrap, untuk mengatur lebar minimum kolom, seperti yang terlihat pada Gambar 3.34.



```
13
14
15
16
17
18
19
20
21
22
23 <div class="table-responsive">
24 <!-- begin:Table -->
25 <table class="table table-row-bordered table-row-gray-100 align-middle gs-0 gy-3">
26 <!-- begin:Table head-->
27 <thead>
28 <tr class="fu-holder text-muted">
29 <th class="fu-holder text-muted">
30 <div class="form-check form-check-sm form-check-custom form-check-solid">
31 <input class="form-check-input" type="checkbox" value="1" data-kt-check="true" data-kt-check-target
32 </div>
33 </th>
34 <th class="min-w-120px">No.</th>
35 <th class="min-w-150px">Campaign</th>
36 <th class="min-w-150px">Date</th>
37 <th class="min-w-140px">Recipient Group / Tag Name</th>
38 <th class="min-w-120px">Sent</th>
39 <th class="min-w-120px">Status</th>
40 <th class="min-w-100px text-end">Actions</th>
41 </tr>
42 </thead>
43 <!-- end:Table head-->
```

Gambar 3.34 Kode Halaman *Dashboard* (1)

Selanjutnya, bagian isi tabel diisi secara manual menggunakan elemen `<tbody>` yang dengan elemen `<td>`, yang berisi konten seperti *link* judul *newsletter*, teks tambahan (seperti kode kampanye), jumlah *newsletter* yang telah terkirim, status pengiriman, serta dua buah tombol

aksi yang menggunakan ikon SVG untuk representasi edit dan *delete*. Isi tabel ini dapat dilihat pada Gambar 3.35.

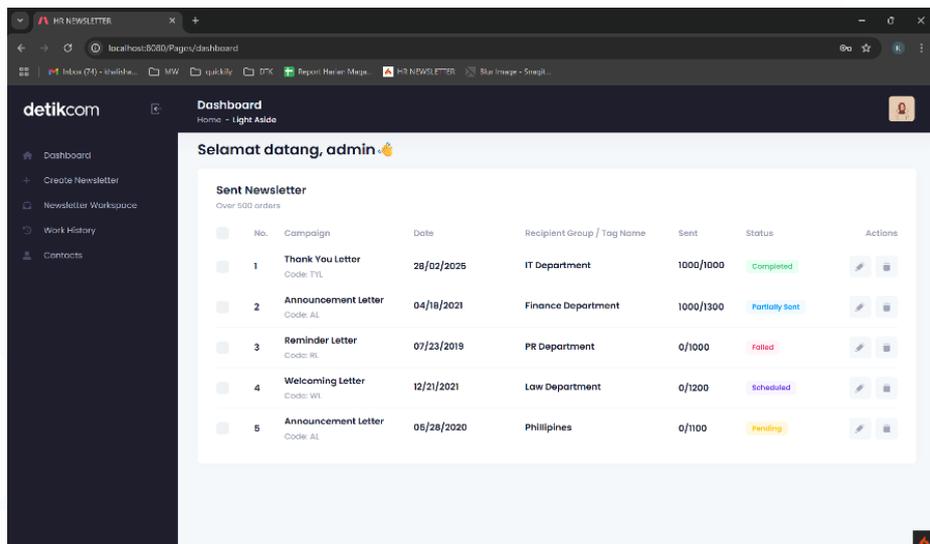
```

52 <td>
53 <a href="#" class="text-dark fw-bold text-hover-primary fs-4">/a>
54 </td>
55 </td>
56 <a href="#" class="text-dark fw-bold text-hover-primary d-block mb-1 fs-6">Thank You Letter</a>
57 <span class="text-muted fw-bold text-muted d-block fs-7">Code: TYL</span>
58 </td>
59 <td class="text-dark fw bold text hover primary fs 6">28/02/2025</td>
60 </td>
61 <a href="#" class="text-dark fw-bold text-hover-primary d-block mb-1 fs-6">IT Department</a>
62 </td>
63 </td>
64 <a href="#" class="text-dark fw bold text hover primary d block mb 1 fs 6">1000/1000</a>
65 </td>
66 </td>
67 <span class="badge badge-light-success">Completed</span>
68 </td>
69 </td>
70 <td class="text-end">
71 <a href="#" class="btn btn-icon btn-bg-light btn-active-color-primary btn-sm me-1">
72 <!--begin:SVG Icon -->
73 <span class="svg-icon svg icon">
74 <!--begin:SVG Icon -->
75 <!--end:SVG Icon -->
76 </span>
77 </td>
78 </td>
79 <a href="#" class="btn btn-icon btn-bg-light btn-active-color-primary btn-sm">
80 <!--begin:SVG Icon -->
81 <span class="svg-icon svg icon">
82 <!--begin:SVG Icon -->
83 <!--end:SVG Icon -->
84 </span>
85 </td>
86 </td>
87 </td>
88 </td>
89 </td>
90 </td>

```

Gambar 3.35 Kode Halaman *Dashboard* (2)

Setelah seluruh proses pembuatan, pengujian, dan perapian kode selesai dilakukan, tampilan akhir yang akan disajikan kepada pengguna ditunjukkan pada Gambar 3.36.

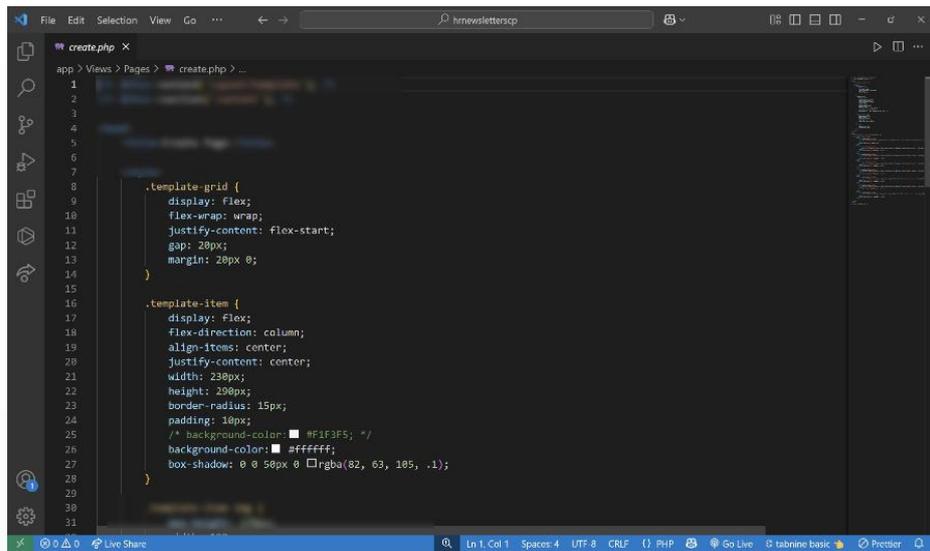


Gambar 3.36 Tampilan Halaman *Dashboard*

3.2.5.1.2 Halaman *Create Newsletter*

Kedua, halaman *Create Newsletter* dirancang untuk menampilkan daftar *template* yang tersedia dalam format *grid*, sehingga memudahkan pengguna dalam memilih *template* yang diinginkan sebagai dasar dalam pembuatan *newsletter*. Daftar *template* ini ditata dengan menggunakan *class* bernama *template-grid*, yang membuat tampilan halaman ini menjadi *flexible*. Hal ini dilakukan dengan menggunakan properti CSS *wrap* pada *flex-wrap* untuk membungkus baris secara otomatis, *gap* untuk memberi jarak antar item, dan *margin* atas-bawah untuk memberikan jarak antara judul halaman dengan *grid* yang dibuat.

Kemudian, setiap *template* yang tersedia pada *grid* dibungkus dalam elemen `<div>` dengan *class* *template-item*, yang dibuat agar tampil rapi dan menarik. *Class* ini juga menggunakan beberapa properti CSS, yaitu *width* dan *height* untuk mengatur ukuran tetap, *justify-content* dan *align-items-center* untuk membuat *template* di tengah secara vertikal dan horizontal, serta *box-shadow* untuk memberikan efek bayangan pada tiap kotak *template*. Struktur dan implementasi *class* *template-item* serta *template-grid* dapat diamati pada Gambar 3.37, yang menggambarkan bagaimana kedua *class* tersebut digunakan dalam membentuk tampilan antarmuka pemilihan *template*.

A screenshot of a code editor window titled 'create.php'. The editor shows CSS code for a newsletter template. The code is as follows:

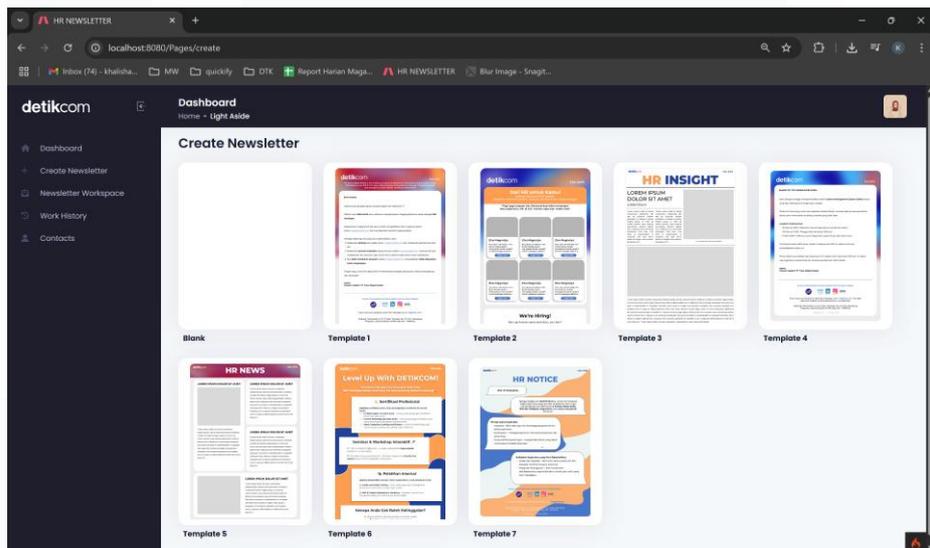
```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
  
.template-grid {  
  display: flex;  
  flex-wrap: wrap;  
  justify-content: flex-start;  
  gap: 28px;  
  margin: 28px 0;  
}  
  
.template-item {  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
  justify-content: center;  
  width: 238px;  
  height: 208px;  
  border-radius: 15px;  
  padding: 18px;  
  /* background-color: #F1F3F5; */  
  background-color: #ffffff;  
  box-shadow: 0 0 58px 0 rgba(82, 63, 105, .1);  
}
```

Gambar 3.37 Kode Halaman *Create Newsletter* (1)

Selain itu, pada Gambar 3.38, dapat dilihat bahwa di dalam elemen `<div>` dengan *class* `template-item`, terdapat elemen gambar yang berfungsi sebagai pratinjau *template*. Gambar ini dapat diklik untuk mengarahkan pengguna ke halaman *Newsletter Workspace* menggunakan atribut *event handler onclick* pada JavaScript. Setiap *template* juga disertai dengan label yang menjelaskan nama *template*, seperti "*Blank*" dan "*Template 1*". Hasil tampilan halaman *Create Newsletter* ini dapat dilihat pada Gambar 3.39.

```
File Edit Selection View Go ... hnewsletterscp
create.php X
app > Views > Pages > create.php > ...
35
36
37
38
39
40
41
42
43
44
45
46 <body>
47 <h1 class="ml-4 mb-5">Create Newsletter</h1>
48 <div class="template-grid">
49 <div class="template-item">
50  site_url('Pages/workspace'); '>
51 </div>
52 <label for="blank">Blank</label>
53 </div>
54 <div class="template-item">
55  site_url('Pages/worksp
56 </div>
57 <label for="template1">template 1</label>
58 </div>
59
60
61
62
63
64
65
```

Gambar 3.38 Kode Halaman *Create Newsletter* (2)

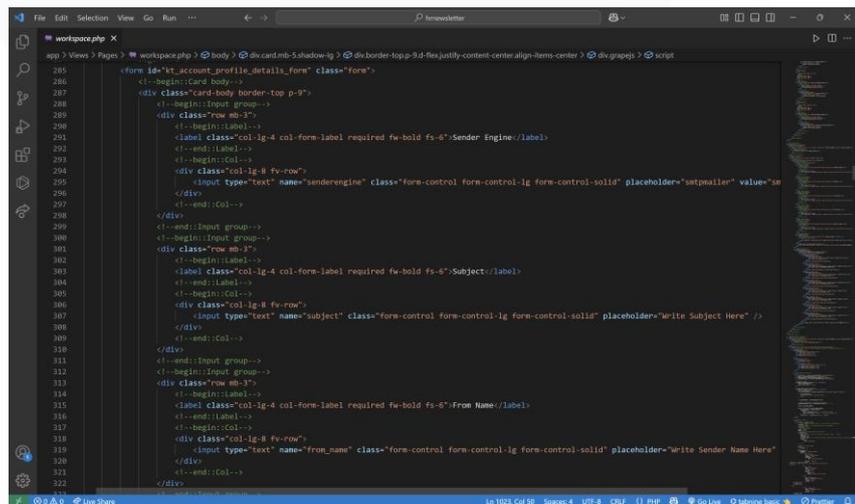


Gambar 3.39 Tampilan Halaman *Create Newsletter*

3.2.5.1.3 Halaman *Newsletter Workspace*

Ketiga, halaman *Newsletter Workspace* dibuat untuk memudahkan pengguna dalam mengelola proses pengiriman *newsletter*, mulai dari pengisian informasi pengiriman, pembuatan konten, pemilihan penerima, hingga peninjauan ulang dan pengiriman akhir. Halaman ini terbagi menjadi tiga bagian utama:

1. Bagian pertama, yaitu *Create Content*, berfungsi sebagai wadah bagi pengguna untuk mengisi informasi pengiriman serta membuat konten *newsletter* sesuai kebutuhan. Pada bagian ini, terdapat dua sub-bagian utama, yaitu *Sending Details* dan *Newsletter Editor*. Pada sub-bagian *Sending Details*, pengguna dapat mengisi informasi terkait pengiriman *email*, seperti *subject email*, nama pengirim, *email* pengirim, dan *email* untuk balasan (*reply-to*). Sub-bagian ini dibuat dalam bentuk *form*, yang terdiri dari elemen-elemen HTML seperti `<label>` untuk memberikan keterangan pada setiap kolom isian, serta `<input>` sebagai wadah bagi pengguna untuk memasukkan data. Contoh struktur kode dari bagian ini dapat dilihat pada Gambar 3.40, yang menampilkan potongan kode HTML untuk elemen *form* tersebut.

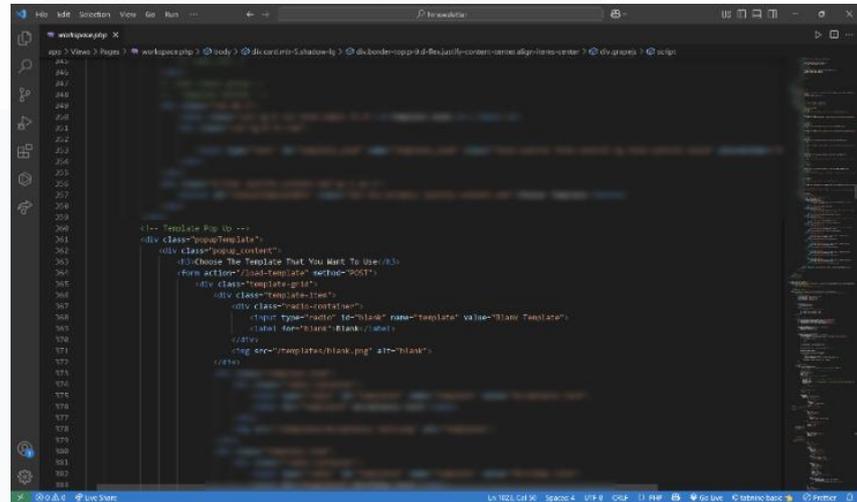


```
285 <form id="kt_account_profile_details_form" class="form">
286 <!--begin::card body-->
287 <div class="card-body border-top p-9">
288 <!--begin::input group-->
289 <div class="row mb-3">
290 <!--begin::label-->
291 <label class="col-lg-4 col-form-label required fw-bold fs-6">Sender Engine</label>
292 <!--end::label-->
293 <!--begin::input-->
294 <div class="col-lg-8 form-row">
295 <input type="text" name="senderengine" class="form-control form-control-lg form-control-solid" placeholder="smtpmail" value="" />
296 </div>
297 <!--end::col-->
298 </div>
299 <!--end::input group-->
300 <!--begin::input group-->
301 <div class="row mb-3">
302 <!--begin::label-->
303 <label class="col-lg-4 col-form-label required fw-bold fs-6">Subject</label>
304 <!--end::label-->
305 <!--begin::col-->
306 <div class="col-lg-8 form-row">
307 <input type="text" name="subject" class="form-control form-control-lg form-control-solid" placeholder="write subject here" />
308 </div>
309 <!--end::col-->
310 </div>
311 <!--end::input group-->
312 <!--begin::input group-->
313 <div class="row mb-3">
314 <!--begin::label-->
315 <label class="col-lg-4 col-form-label required fw-bold fs-6">From Name</label>
316 <!--end::label-->
317 <!--begin::col-->
318 <div class="col-lg-8 form-row">
319 <input type="text" name="from_name" class="form-control form-control-lg form-control-solid" placeholder="write sender name here" />
320 </div>
321 <!--end::col-->
322 </div>
```

Gambar 3.40 Kode Bagian Pertama (*Create Content*) dari *Newsletter Workspace* (1)

Selain itu, pengguna dapat memilih *template* yang akan digunakan dengan menekan tombol “*Choose Template*”, yang akan memunculkan sebuah *pop-up* berisi beberapa opsi *template*. Pilihan *template* ditampilkan menggunakan radio *button*, sehingga hanya satu *template* dapat dipilih. Pengimplementasian radio *button* ini dilakukan dengan menggunakan elemen `<div>` class Bootstrap *radio-container*, yang berisi elemen `<input>` dengan

atribut `type="radio"` untuk mengaktifkan radio *button*, serta elemen `<label>` yang menunjukkan judul *template*. Struktur ini dapat dilihat pada Gambar 3.41.

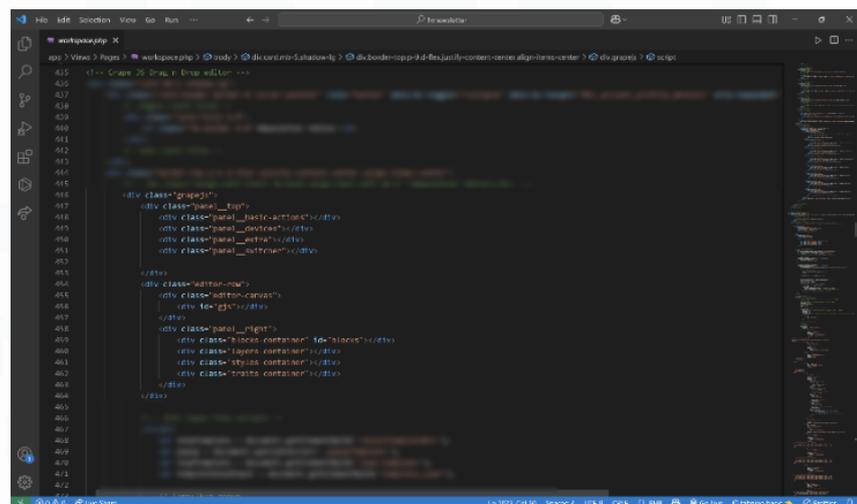


```
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383

<!-- Template Pop Up -->
<div class="popup-template">
  <div class="popup-content">
    <!-- Choose The Template That You Want To Use -->
    <form action="" method="POST">
      <div class="template-grid">
        <div class="template-item">
          <div class="radio-container">
            <input type="radio" id="blank" name="template" value="Blank Template"/>
            <label for="blank" Blank />
          </div>
          
        </div>
      </div>
    </form>
  </div>
</div>
```

Gambar 3.41 Kode Bagian Pertama (*Create Content*) dari *Newsletter Workspace* (2)

Setelah *template* dipilih, tampilan akan beralih ke bagian *newsletter* editor, yaitu area untuk membuat dan mengedit konten *newsletter* menggunakan GrapesJS. Penggunaan GrapesJS diimplementasikan dengan membangun tampilan melalui beberapa elemen `<div>` yang memiliki *class* khusus dari GrapesJS, seperti yang ditunjukkan pada Gambar 3.42.

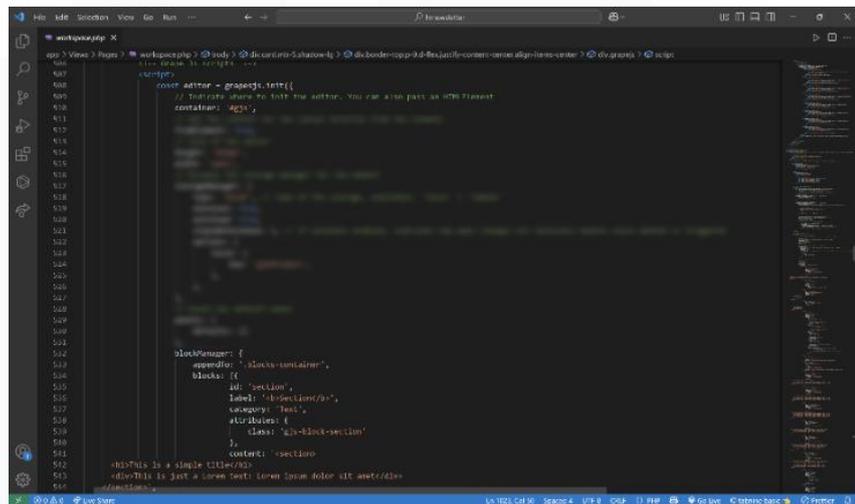


```
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

<!-- Grapes JS Editor -->
<div class="grapes">
  <div class="panel_top">
    <div class="panel__back-arrow"></div>
    <div class="panel__devices"></div>
    <div class="panel__extra"></div>
    <div class="panel__outline"></div>
  </div>
  <div class="editor-main">
    <div class="editor-container">
      <div id="js"></div>
    </div>
    <div class="panel__right">
      <div class="blocks-container" id="blocks"></div>
      <div class="layers-container"></div>
      <div class="styles-container"></div>
      <div class="props-container"></div>
    </div>
  </div>
</div>
```

Gambar 3.42 Kode Bagian Pertama (*Create Content*) dari *Newsletter Workspace* (3)

Selanjutnya, dilakukan *inisialisasi* pada JavaScript menggunakan *grape.init*, kemudian ditambahkan beberapa *custom blocks* pada BlockManager GrapesJS, seperti yang ditunjukkan pada Gambar 3.43.

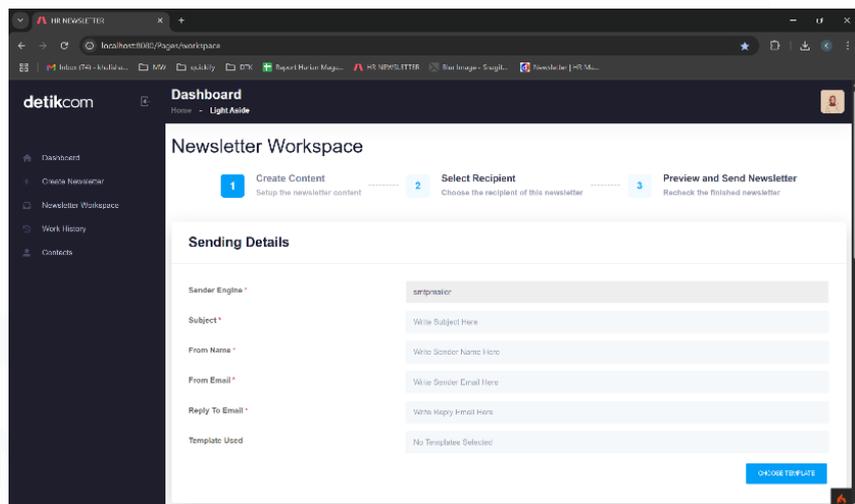


```
workspace.gjs({
  // Initialize GrapesJS to init the editor. You can also pass an HTML element
  container: '#editor',
  // ...
});

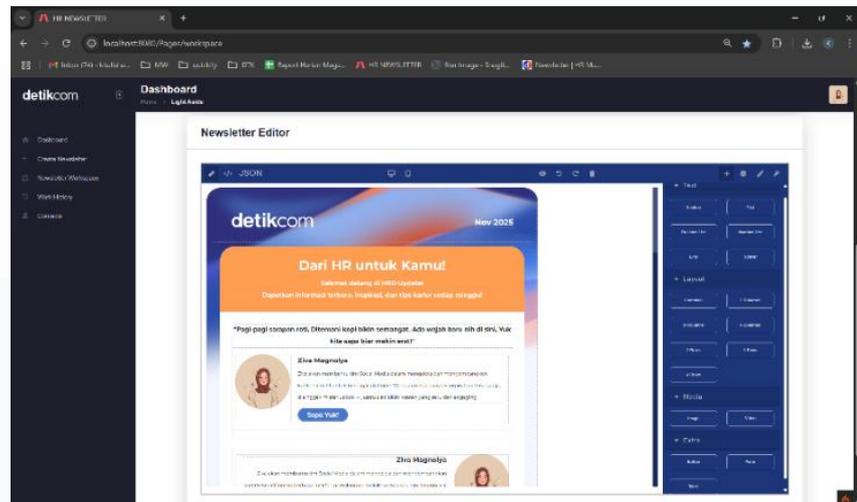
BlockManager: {
  namespace: 'block-manager',
  blocks: [
    {
      id: 'section',
      label: 'BlockManager/33',
      category: 'text',
      attributes: {
        class: 'gjs-block-section'
      },
      content: '<section>'
    }
  ]
}
```

Gambar 3.43 Kode Bagian Pertama (*Create Content*) dari *Newsletter Workspace* (4)

Setelah dilakukan beberapa perbaikan dan pengujian kode, hasil tampilan bagian pertama dari halaman *Newsletter Workspace* dapat dilihat pada Gambar 3.44 dan Gambar 3.45.

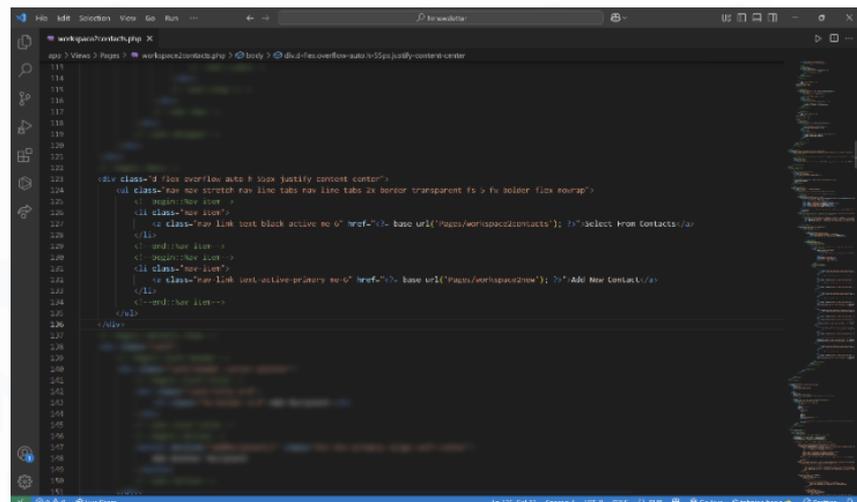


Gambar 3.44 Tampilan Bagian Pertama (*Create Content*) dari *Newsletter Workspace* (1)



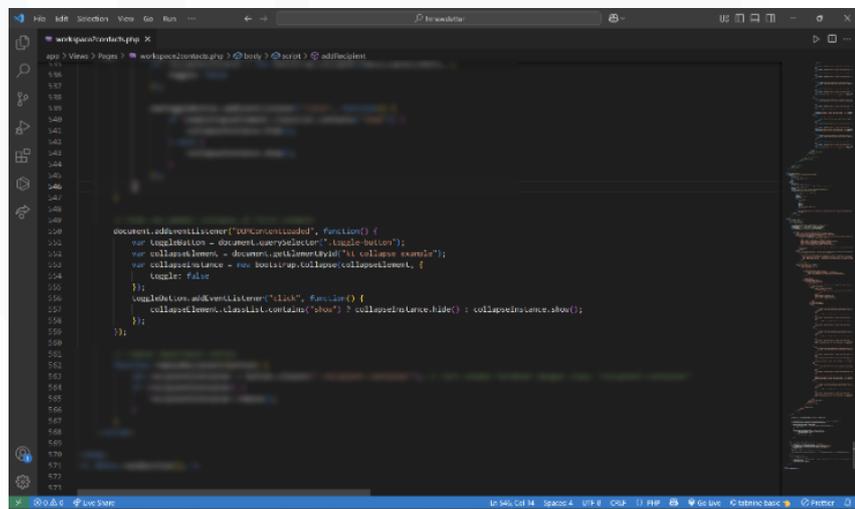
Gambar 3.45 Tampilan Bagian Pertama (*Create Content*) dari *Newsletter Workspace* (2)

2. Bagian kedua, yaitu *Select Recipient*, merupakan area untuk memilih penerima *newsletter*. Pemilihan dapat dilakukan dengan dua cara: memilih kontak yang sudah tersedia atau menambahkan kontak baru sebagai penerima. Kedua opsi tersebut ditampilkan menggunakan komponen antarmuka dari *framework* Metronic, yang mengadaptasi beberapa *class* Bootstrap seperti *nav-item* dan *nav-link* untuk membentuk *navigation bar*, seperti yang ditunjukkan pada Gambar 3.46.



Gambar 3.46 Kode Bagian Kedua (*Select Recipient*) dari *Newsletter Workspace* dengan Pemilihan *Recipient* dari Daftar Kontak (1)

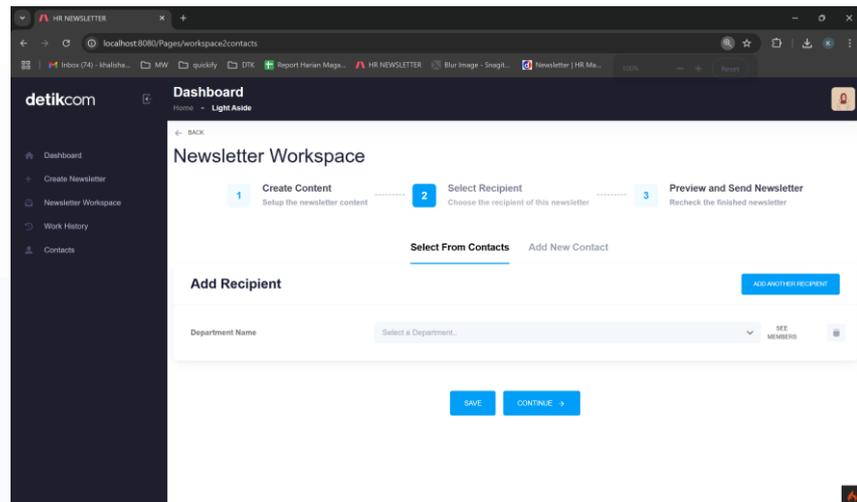
Pada bagian pemilihan penerima *newsletter* melalui kontak yang sudah tersedia, ditampilkan sebuah kotak yang memungkinkan pengguna memilih nama kontak atau departemen sebagai penerima. Setelah pilihan dilakukan, pengguna dapat menampilkan daftar anggota dari kontak atau departemen tersebut dengan menekan tombol *See Members*. Tampilan daftar anggota ini menggunakan fitur *collapse* dari Bootstrap yang diaktifkan melalui JavaScript pada tombol tersebut, seperti terlihat pada Gambar 3.47.



```
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

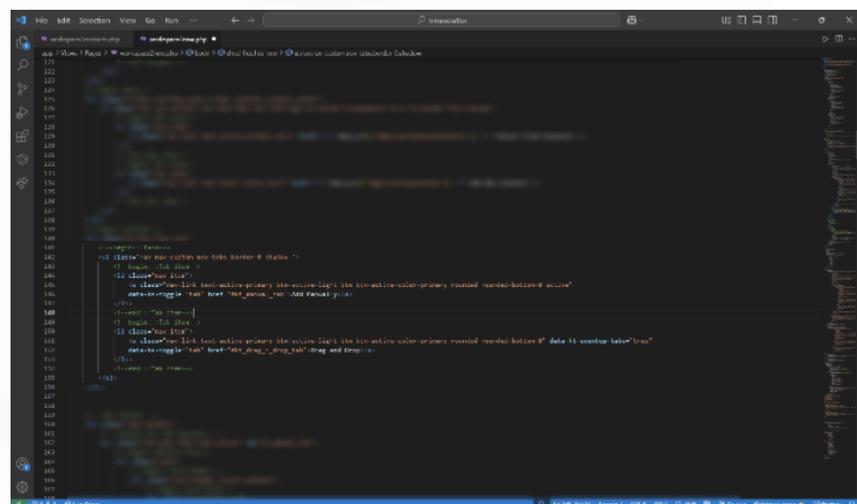
Gambar 3.47 Kode Bagian Kedua (*Select Recipient*) dari *Newsletter Workspace* dengan Pemilihan *Recipient* dari Daftar Kontak (2)

Hasil tampilan dari bagian *Select Recipient* melalui pemilihan kontak dapat dilihat pada Gambar 3.48, yang juga menunjukkan keberadaan tombol *Add Another Recipient*. Tombol ini memungkinkan pengguna menambahkan lebih dari satu kontak atau departemen sebagai penerima *newsletter*. Selain itu, setiap kotak pemilihan kontak atau departemen dilengkapi dengan tombol dengan ikon *trashcan* yang dapat digunakan untuk menghapus kontak yang tidak diinginkan.



Gambar 3.48 Tampilan Bagian Kedua (*Select Recipient*) dari *Newsletter Workspace* dengan Pemilihan *Recipient* dari Daftar Kontak

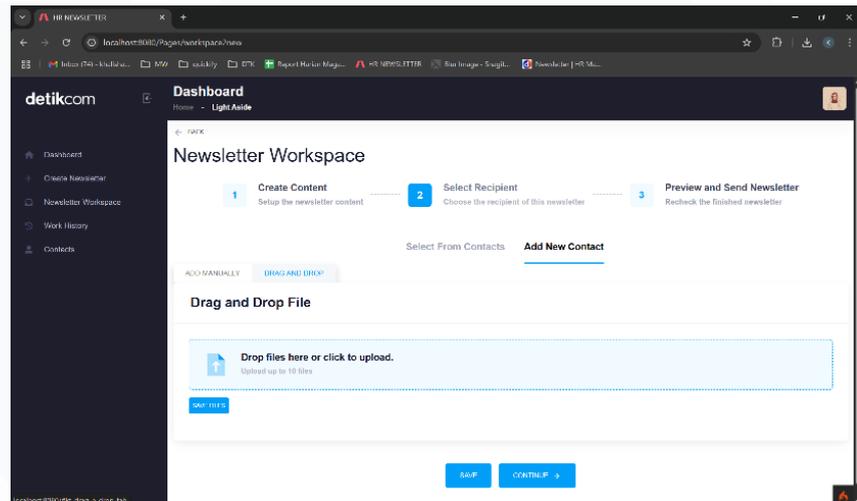
Selanjutnya, pada bagian pemilihan penerima *newsletter* dengan menambahkan kontak baru, tersedia dua opsi, yaitu pengisian kontak secara manual dan metode *drag and drop*. Kedua opsi ini dibuat menggunakan *class* Bootstrap tab yang telah dimodifikasi oleh *framework* Metronic, seperti terlihat pada Gambar 3.49.



Gambar 3.49 Kode Bagian Kedua (*Select Recipient*) dari *Newsletter Workspace* dengan Pemilihan *Recipient* melalui Penambahan Kontak Baru secara Manual (1)

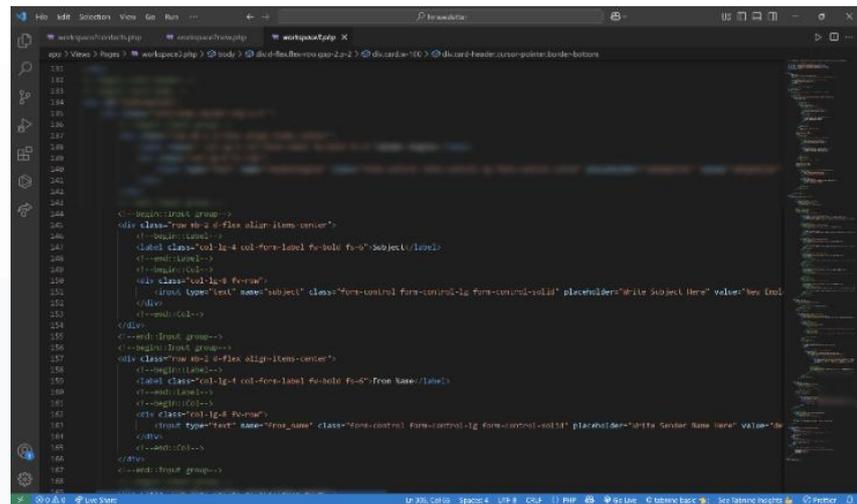
Isi dari tab yang telah dimodifikasi tersebut dibuat menggunakan elemen `<div>` dengan *class* `tab-content`. Di dalamnya terdapat elemen `<div>` lain dengan *class* `tab-pane` yang memisahkan konten

Sedangkan jika pengguna memilih menambahkan kontak melalui metode *drag and drop*, pengguna dapat langsung menarik (*drag*) *file* dari file explorer dan meletakkannya (*drop*) ke area kotak *drag and drop* yang telah disediakan, seperti yang terlihat pada Gambar 3.53 .



Gambar 3.53 Kode Bagian Kedua (*Select Recipient*) dari *Newsletter Workspace* dengan Pemilihan *Recipient* melalui Penambahan Kontak Baru secara *Drag and Drop*

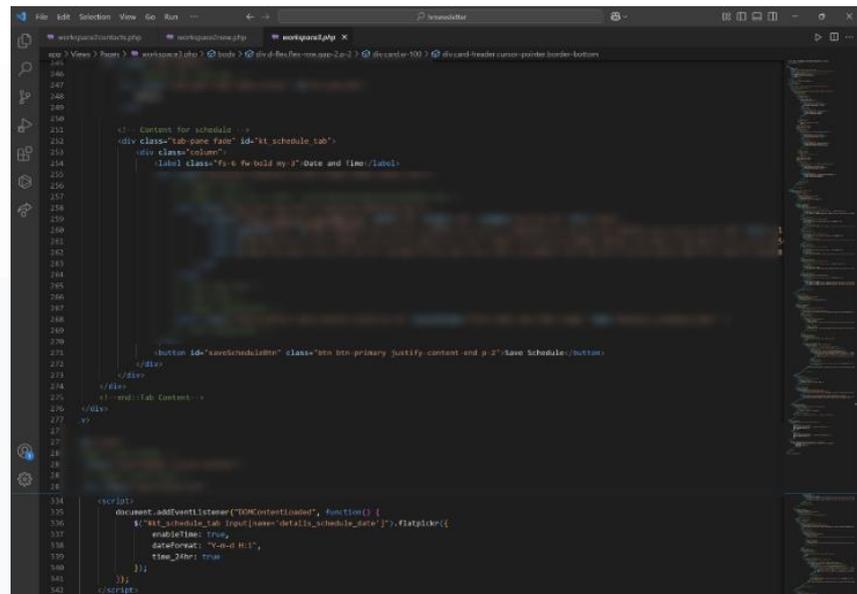
3. Bagian ketiga, *Preview and Send Newsletter*, berfungsi untuk meninjau ulang seluruh informasi yang telah diisi oleh pengguna sebelum *newsletter* dikirimkan. Informasi tersebut mencakup data pengiriman, konten *newsletter*, serta daftar penerima. Seluruh data ini ditampilkan di sisi kiri halaman dalam bentuk elemen-elemen `<input>` yang masing-masing disertai dengan elemen `<label>` sebagai penanda atau keterangan nama *field*, sebagaimana ditunjukkan pada Gambar 3.54.



```
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000
```

Gambar 3.54 Kode Bagian Ketiga (*Preview and Send Newsletter*) dari *Newsletter Workspace* (1)

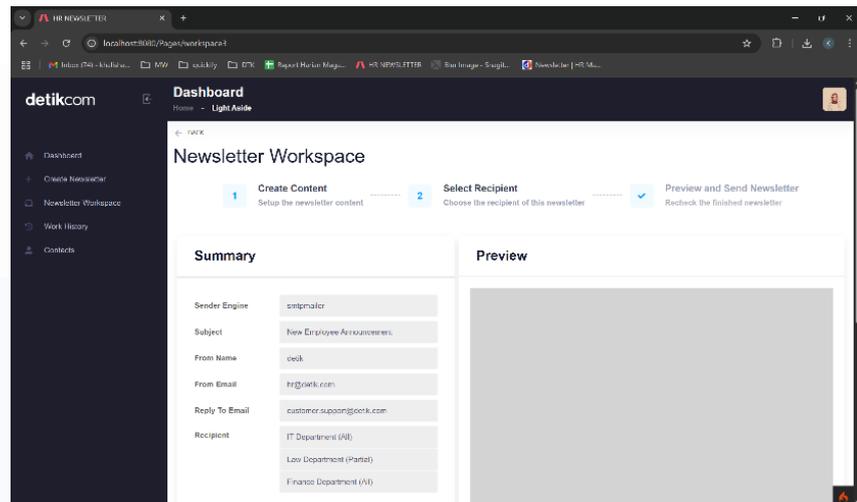
Pada sisi kiri halaman ini juga terdapat fitur penjadwalan pengiriman atau *sending schedule*. Fitur ini memungkinkan pengguna untuk menentukan waktu pengiriman *newsletter* secara spesifik. Jika pengguna memilih opsi penjadwalan, akan muncul sebuah ikon kalender. Ketika ikon tersebut diklik, pengguna dapat memilih tanggal, bulan, tahun, dan waktu pengiriman sesuai kebutuhan. Komponen kalender ini dibuat menggunakan *library* JavaScript Flatpickr, yang juga digunakan pada Metronic sesuai arahan dari mentor. Implementasi fitur ini dapat dilihat pada Gambar 3.55.



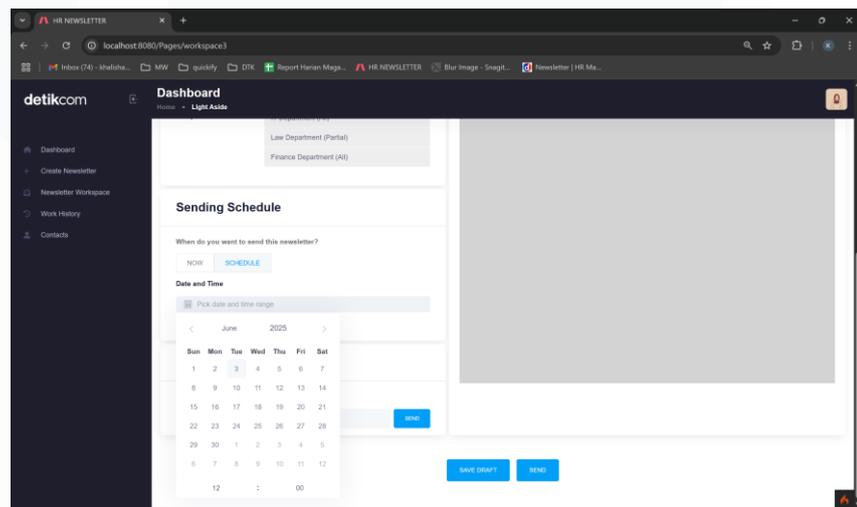
```
211 <!-- Content for schedule -->
212 <div class="tab-pane fade" id="tt-schedule-tab">
213 <div class="col-sm-12">
214 <label class="fr-fw-bc12 my-2">Date and Time/Label:
215
216
217
218
219
220
221
222 <button id="saveScheduleBtn" class="btn btn-primary justify-content-end p-2">Save Schedule/Buttons
223
224 </div>
225 </div>
226 </div>
227 </div>
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Gambar 3.55 Kode Bagian Ketiga (*Preview and Send Newsletter*) dari *Newsletter Workspace* (2)

Selain itu, disediakan pula kolom input untuk melakukan pengiriman uji coba atau *test send newsletter* ke alamat email tertentu yang dimasukkan secara manual oleh pengguna. Fitur ini memberikan fleksibilitas agar pengguna dapat memastikan bahwa konten dan format *newsletter* sudah sesuai sebelum benar-benar dikirimkan. Sementara itu, di sisi kanan halaman ditampilkan konten *newsletter* yang telah dirancang oleh pengguna. Bagian ini menyajikan pratinjau visual dari hasil akhir *newsletter*, sehingga pengguna dapat melihat secara langsung tampilan yang akan diterima oleh penerima. Gambar 3.56 dan Gambar 3.57 menunjukkan tampilan halaman *Preview and Send Newsletter* dari *Newsletter Workspace*.



Gambar 3.56 Tampilan Bagian Ketiga (*Preview and Send Newsletter*) dari *Newsletter Workspace* (1)



Gambar 3.57 Bagian Ketiga (*Preview and Send Newsletter*) dari *Newsletter Workspace* (2)

3.2.5.1.4 Halaman *Work History*

Keempat, halaman *Work History* dirancang untuk menampilkan daftar *template newsletter* yang sebelumnya telah dibuat dan diedit oleh pengguna. *Template* ini disusun dalam format tampilan *grid*, sehingga secara visual memiliki kemiripan dengan halaman *Create Newsletter*. Penggunaan *grid* ini bertujuan untuk memudahkan pengguna dalam menelusuri riwayat pekerjaan mereka secara cepat dan efisien.

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Gambar 3.58 Kode Halaman *Work History* (1)

Sama halnya dengan halaman *Create Newsletter*, tata letak daftar *template* pada halaman ini diatur menggunakan *class* *template-grid* dan *template-item*, sebagaimana ditunjukkan pada Gambar 3.58. *Class* *template-grid* berfungsi untuk mengelola susunan elemen agar membentuk tampilan *grid* yang responsif terhadap berbagai ukuran layar perangkat, sementara *class* *template-item* digunakan untuk memastikan setiap item tampil secara teratur, seragam, dan menarik secara visual.

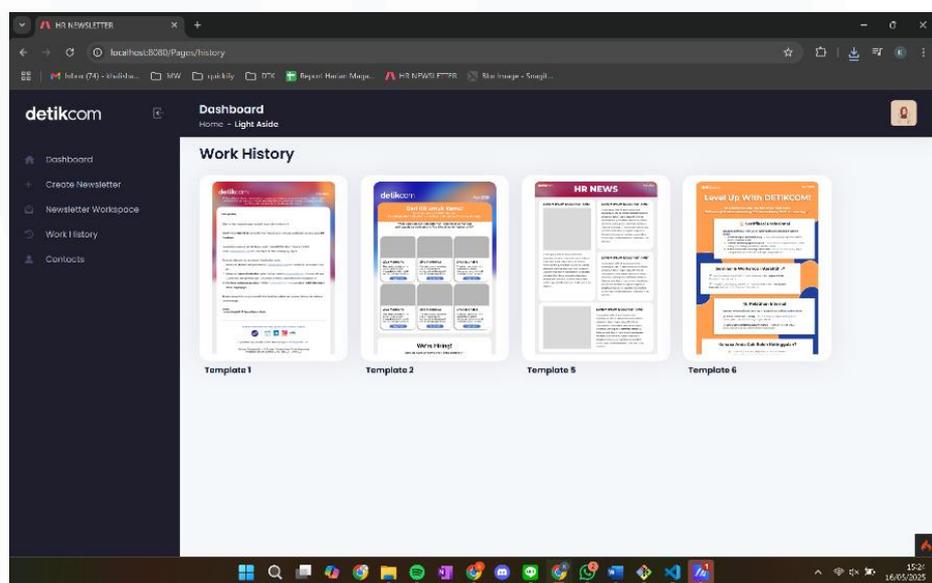
```

41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Gambar 3.59 Kode Halaman *Work History* (2)

Implementasi dari kedua *class* tersebut dapat dilihat pada Gambar 3.59, yang memperlihatkan bahwa setiap *template* dibungkus dalam elemen `<div>` dengan *class* `template-grid`. Di dalamnya terdapat elemen `<div>` lain dengan *class* `template-item`, yang selanjutnya memuat elemen `` sebagai pratinjau visual dari masing-masing *template newsletter*. Tampilan akhir dari halaman *Work History* disajikan pada Gambar 3.60.



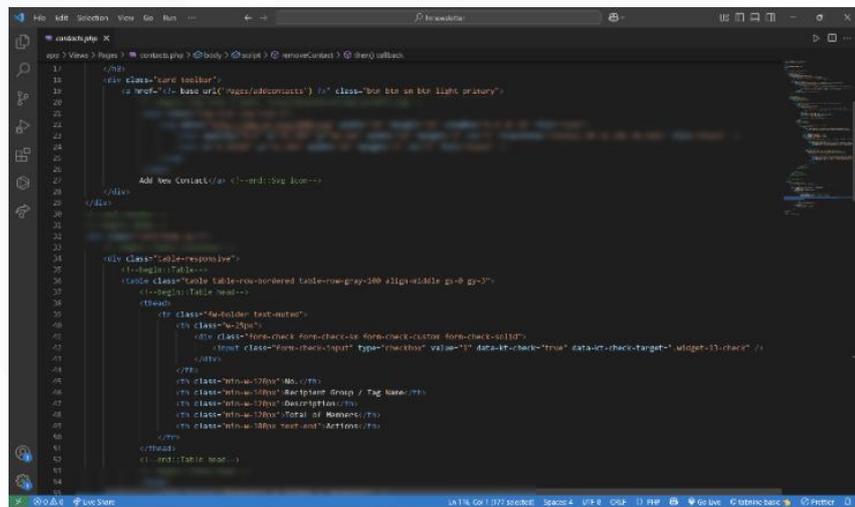
Gambar 3.60 Tampilan Halaman *Work History*

3.2.5.1.5 Halaman *Contacts*

Kelima, halaman *Contacts* dibuat untuk menampilkan tabel yang berisi daftar kontak yang ada pada akun pengguna. Halaman ini terbagi menjadi 5 bagian, yaitu:

1. Bagian pertama, yaitu *Contact List*, dirancang untuk menampilkan daftar kontak yang tersimpan dalam akun pengguna. Seluruh kontak disajikan dalam bentuk tabel untuk memudahkan pengguna dalam mengelola, menelusuri, dan mencari informasi kontak yang telah disimpan. Tabel ini memuat beberapa kolom penting, seperti *Recipient Group* atau *Tag Name*, *Description*, *Total of Members*, serta kolom *Action* yang menyediakan opsi *Edit* dan *Delete*. Tabel

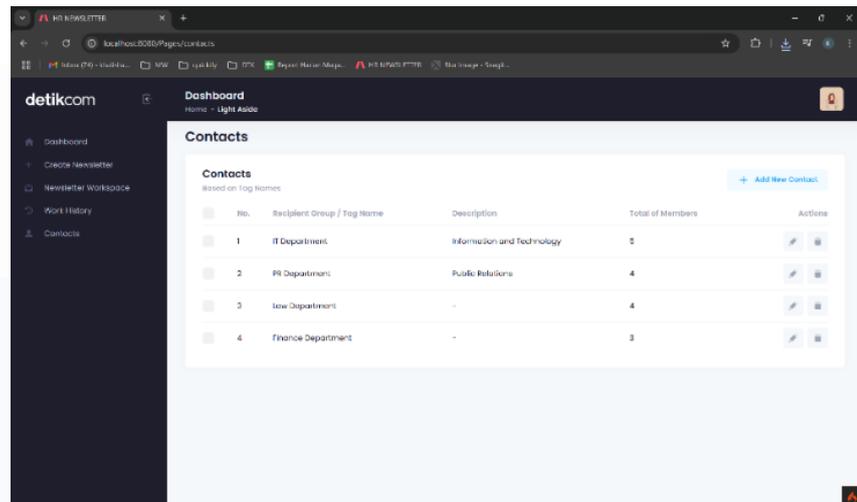
pada bagian ini dikembangkan dengan menggunakan *class* *table-responsive*, yang memungkinkan tampilan tabel menyesuaikan dengan berbagai ukuran perangkat. *Class* ini merupakan bagian dari *framework* Metronic, dan penggunaannya mengacu pada arahan dari mentor. Untuk memperjelas struktur data, ditambahkan pula *class* *table-row-bordered* agar setiap baris pada tabel memiliki garis pembatas, sebagaimana dapat dilihat pada Gambar 3.61.



```
11 </tbody>
12 </table>
13 </div>
14 </div>
15 </div>
16 </div>
17 </div>
18 </div>
19 </div>
20 </div>
21 </div>
22 </div>
23 </div>
24 </div>
25 </div>
26 </div>
27 </div>
28 </div>
29 </div>
30 </div>
31 </div>
32 </div>
33 </div>
34 </div>
35 </div>
36 </div>
37 </div>
38 </div>
39 </div>
40 </div>
41 </div>
42 </div>
43 </div>
44 </div>
45 </div>
46 </div>
47 </div>
48 </div>
49 </div>
50 </div>
51 </div>
52 </div>
53 </div>
54 </div>
55 </div>
56 </div>
57 </div>
58 </div>
59 </div>
60 </div>
61 </div>
62 </div>
63 </div>
64 </div>
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </div>
```

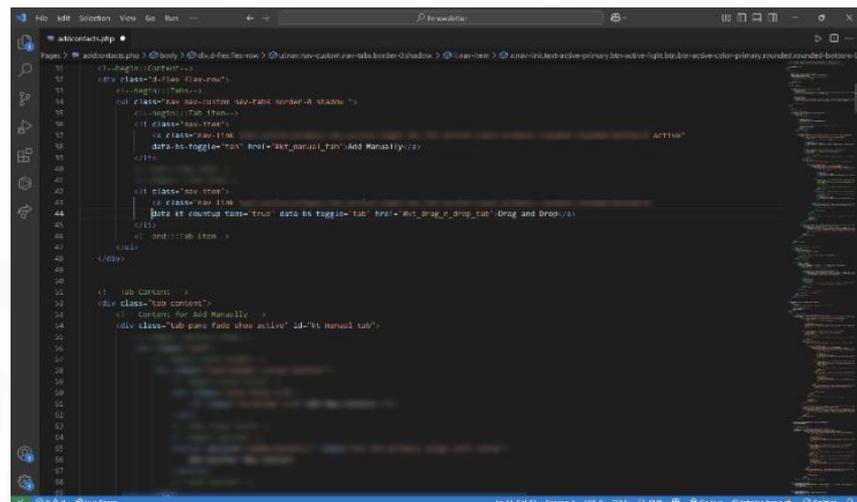
Gambar 3.61 Kode Bagian Pertama (*Contact List*) dari Halaman *Contacts*

Selain tabel, bagian ini juga dilengkapi dengan tombol *Add New Contact* yang berfungsi untuk mengarahkan pengguna ke bagian kedua (*Add Contact*) dari halaman *Contacts*. Bagian kedua tersebut memungkinkan pengguna menambahkan kontak baru ke dalam daftar kontak yang dimiliki. Tampilan akhir dari bagian pertama *Contacts* disajikan pada Gambar 3.62.



Gambar 3.62 Tampilan Bagian Pertama (*Contact List*) dari Halaman *Contacts*

2. Bagian kedua, yaitu *Add Contact*, berfungsi untuk menambah kontak, dengan menyediakan dua metode dalam menambahkan kontak, yaitu secara manual dan melalui fitur *drag and drop*. Kedua metode ini dipisahkan dengan menggunakan *class* tab pada Bootstrap, yang kemudian tiap konten dari metode tersebut dibungkus dengan `<div>` dengan *class* `tab-content`, dan `tab-pane` untuk tiap isi kontennya, seperti yang dapat dilihat pada Gambar 3.63.



Gambar 3.63 Kode Bagian Kedua (*Add Contact*) dari Halaman *Contacts* (1)

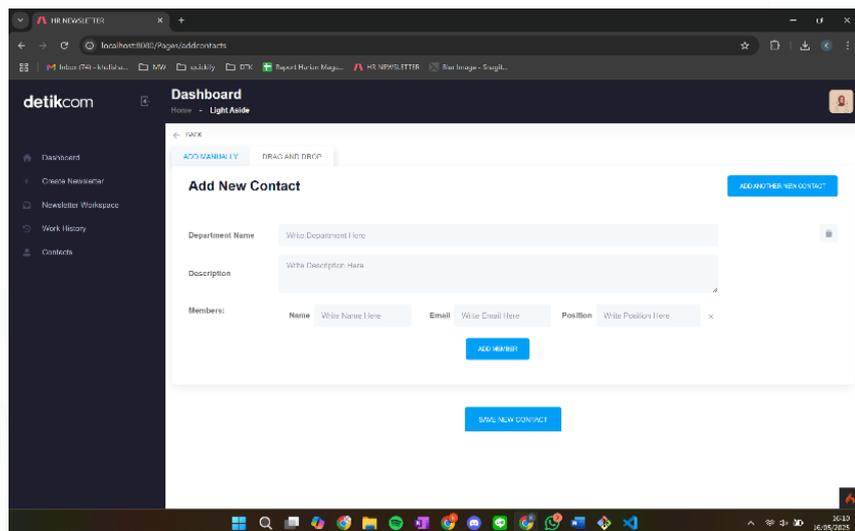

```

177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

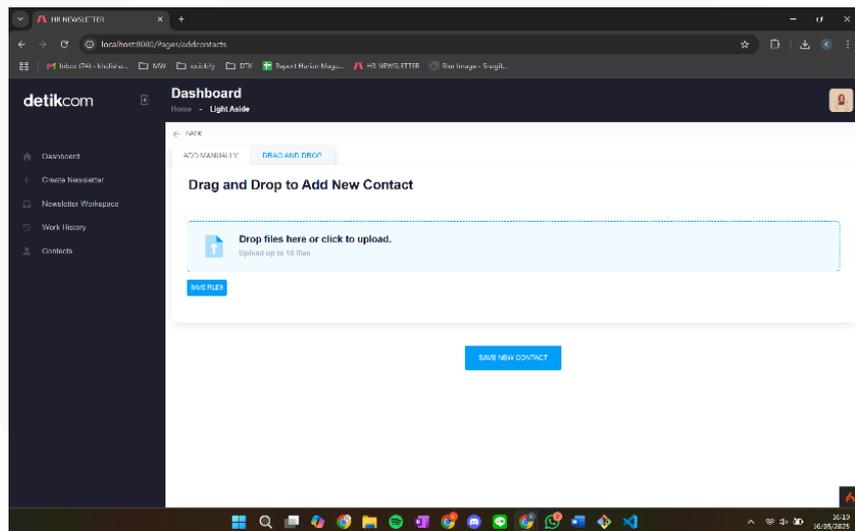
Gambar 3.65 Kode Bagian Kedua (*Add Contact*) dari Halaman *Contacts* melalui *Drag and Drop*

Pada Dropzone ini, juga dilakukan beberapa modifikasi pada elemen, pengaturan tipe *file*, serta fungsi-fungsi Dropzone.js seperti *success*, *removedfile*, dan *error*, sebagaimana yang diterapkan pada bagian kedua halaman *Newsletter Workspace*, yaitu pemilihan *recipient* dengan menambahkan kontak baru melalui *drag and drop*. Gambar 3.65 menunjukkan kode *layout* Dropzone pada bagian kedua (*Add Contact*).



Gambar 3.66 Tampilan Bagian Kedua (*Add Contact*) dari Halaman *Contacts* melalui *Input Manual*

Setelah berbagai penyesuaian dan pengujian kode dilakukan, metode penambahan kontak secara manual telah berhasil diimplementasikan. Detail implementasi metode manual dapat dilihat pada Gambar 3.66. Sementara itu, pengaturan Dropzone untuk metode *drag and drop* juga telah diterapkan dan dapat dilihat pada Gambar 3.67.



Gambar 3.67 Tampilan Bagian Kedua (*Add Contact*) dari Halaman *Contacts* melalui *Drag and Drop*

3. Bagian ketiga, yaitu *Edit Contact*, berfungsi untuk mengubah informasi dari kontak yang sudah ada. Pada bagian ini tersedia *form*, seperti yang ditampilkan pada Gambar 3.68, yang memungkinkan pengguna untuk memperbarui data kontak. Selain itu, terdapat pula tabel yang menampilkan daftar anggota (*member*) yang termasuk dalam kontak tersebut, sehingga pengguna dapat dengan mudah melakukan pembaruan data.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

```
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Gambar 3.68 Kode Bagian Ketiga (Edit Contact) dari Halaman Contacts (1)

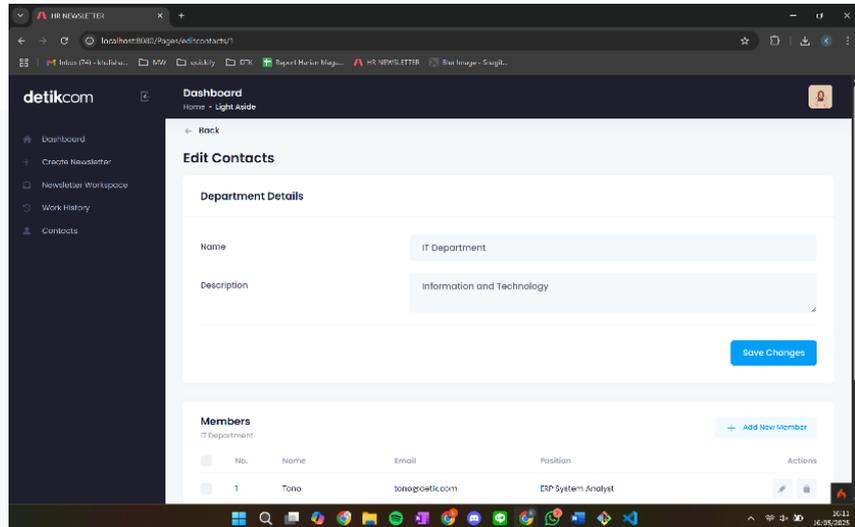
Tampilan anggota dari kontak yang dipilih akan ditampilkan menggunakan tabel dengan class table-responsive dari Bootstrap yang digunakan oleh Metronic. Tabel ini menyajikan informasi anggota seperti nama, email, dan posisi, sebagaimana ditunjukkan pada Gambar 3.69.

```
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

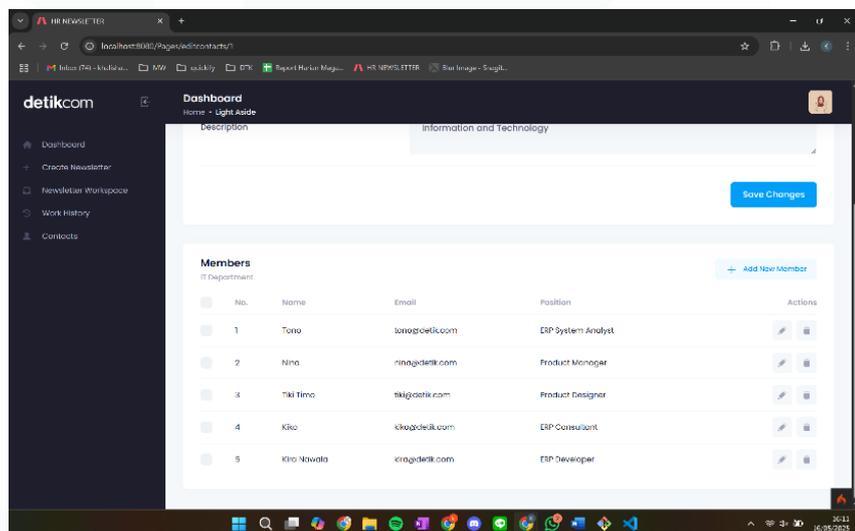
Gambar 3.69 Kode Bagian Ketiga (Edit Contact) dari Halaman Contacts (2)

Hasil tampilan bagian ketiga (Edit Contact) dari halaman Contacts dapat dilihat pada Gambar 3.70 yang menunjukkan form untuk melakukan perubahan nama dan deskripsi kontak yang dipilih oleh pengguna, dan Gambar 3.71 yang menampilkan daftar anggota

yang ada dalam kontak tersebut, beserta dengan tombol *Add New Member* untuk mengarahkan pengguna ke bagian keempat (*Add Member*) dari halaman *Contacts*.



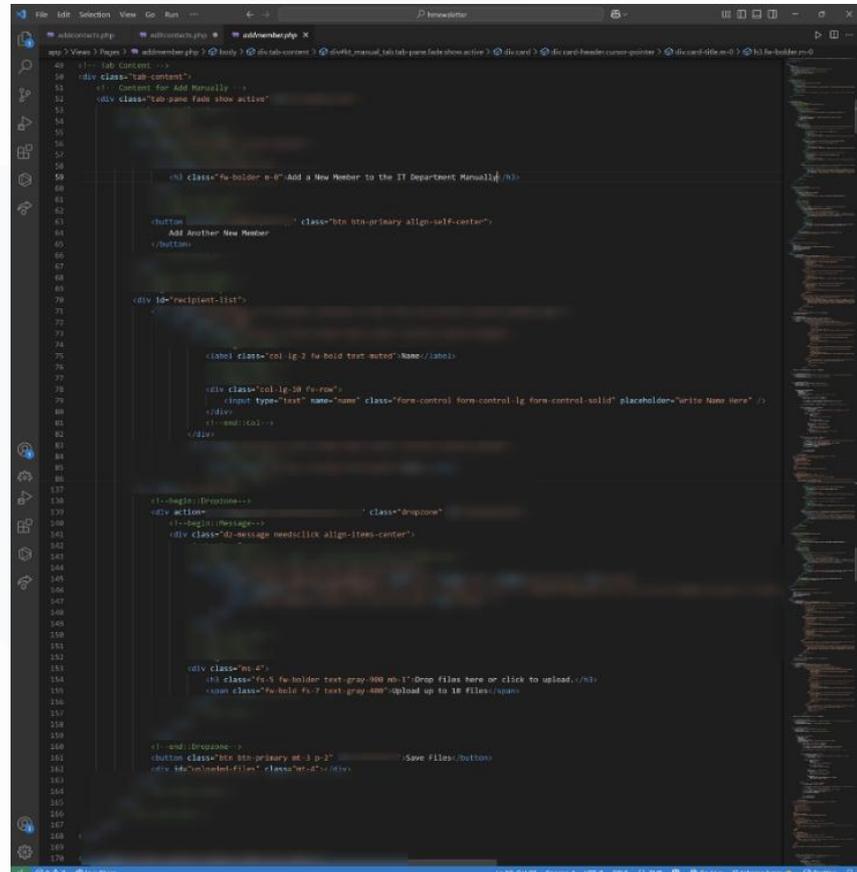
Gambar 3.70 Tampilan Bagian Ketiga (Edit *Contact*) dari Halaman *Contacts* (1)



Gambar 3.71 Tampilan Bagian Ketiga (Edit *Contact*) dari Halaman *Contacts* (2)

4. Bagian keempat, yaitu *Add Member*, memungkinkan pengguna untuk menambahkan anggota ke dalam kontak. Fitur ini disediakan dalam dua metode, serupa dengan bagian kedua (*Add Contact*) pada halaman *Contacts*, yaitu melalui input manual dan *drag and drop*. Pada metode input manual, pengguna dapat mengisi

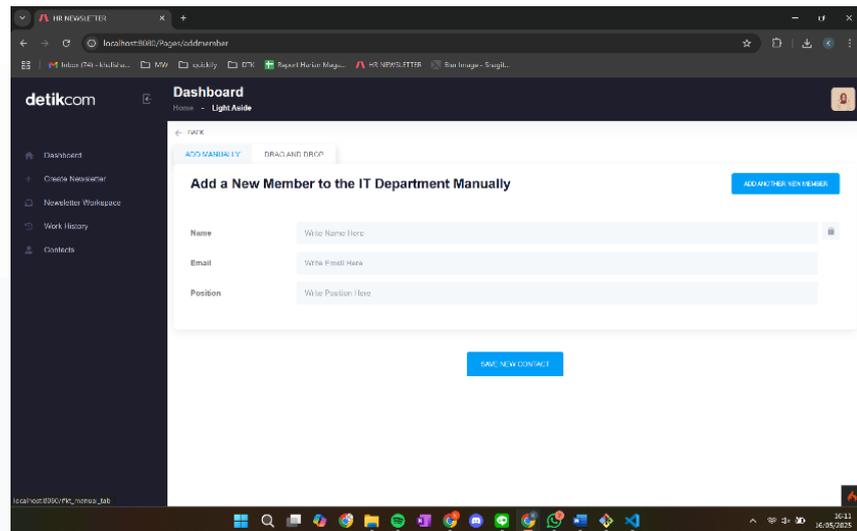
informasi anggota, seperti nama, email, dan posisi, melalui elemen `<input>` yang tersedia.



```
49 <!-- Tab Content -->
50 <div class="tab-content">
51 <!-- Content for Add Manually -->
52 <div class="tab-pane fade show active">
53
54
55
56
57
58
59 <div class="fu-holder m-0">Add a New Member to the IT Department Manually (0).
60
61
62
63 <button class="btn btn-primary align-self-center">
64 Add Another New Member
65 </button>
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

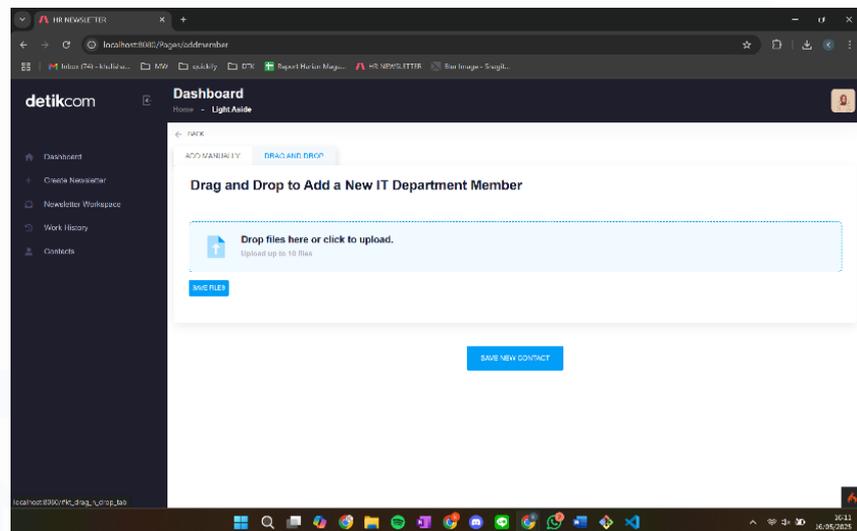
Gambar 3.72 Kode Bagian Keempat (*Add Member*) dari Halaman *Contacts* melalui Input Manual dan *Drag and Drop*

Sementara itu, pada metode *drag and drop*, pengguna dapat mengunggah *file* berisi data anggota secara langsung ke area yang telah disediakan. Metode ini menggunakan *library* JavaScript *Dropzone.js* yang sudah terintegrasi dalam *framework* *Metronic*, sama seperti pada bagian kedua (*Add Contact*) dari halaman *Contacts*. Struktur *layout* dari kedua metode ini dapat dilihat pada Gambar 3.72.



Gambar 3.73 Tampilan Bagian Keempat (*Add Member*) dari Halaman *Contacts* melalui Input Manual

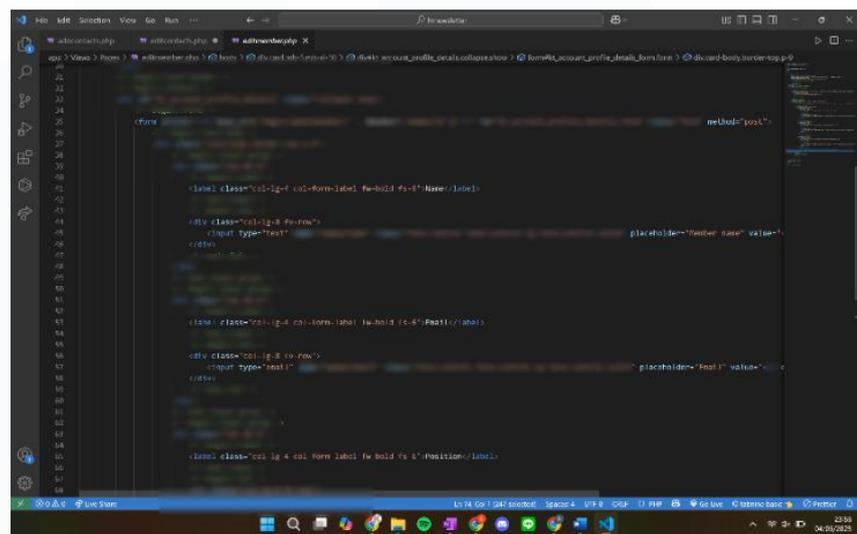
Hasil tampilan metode pertama, yaitu menambahkan anggota melalui pengisian secara manual untuk mengisi data anggota tersedia pada Gambar 3.73. Sedangkan, hasil tampilan metode kedua, yaitu menambahkan anggota melalui *drag and drop* dapat dilihat pada Gambar 3.74.



Gambar 3.74 Tampilan Bagian Keempat (*Add Member*) dari Halaman *Contacts* melalui *Drag and Drop*

5. Bagian kelima, yaitu *Edit Member*, berfungsi untuk memperbarui data anggota yang telah ditambahkan sebelumnya. Tampilan pada

bagian ini menyerupai halaman Edit *Contact*, yaitu dengan menyediakan *form* yang memungkinkan pengguna untuk mengubah informasi anggota secara langsung agar data tetap akurat dan relevan.



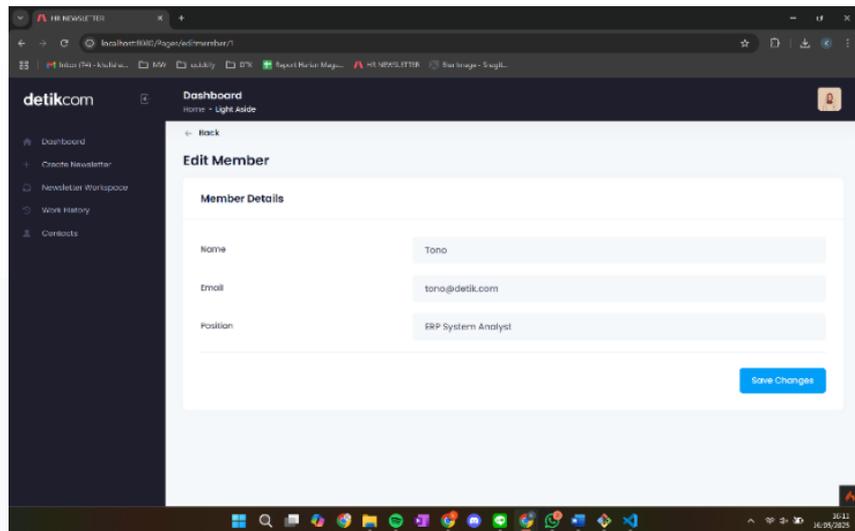
```

<form method="post">
  <label class="col-lg-4 col-form-label fw-bold fs-6" Name: />Name: />
  <input class="col-lg-8 form-control" type="text" placeholder="Member name" value="" />
  <label class="col-lg-4 col-form-label fw-bold fs-6" Position: />Position: />
  <input class="col-lg-8 form-control" type="text" placeholder="Full" value="" />
  <input class="col-lg-4 col-form-label fw-bold fs-6" Position: />Position: />

```

Gambar 3.75 Kode Bagian Kelima (Edit *Member*) dari Halaman *Contacts*

Form ini terdiri dari elemen `<label>` yang berfungsi sebagai penanda nama isian, dan elemen `<input>` yang digunakan untuk memasukkan perubahan data anggota. Struktur *layout* dari *form* ini dapat dilihat pada Gambar 3.75, sedangkan hasil tampilan lengkap dari bagian Edit Member ditunjukkan pada Gambar 3.76.



Gambar 3.76 Tampilan Bagian Kelima (*Edit Member*) dari Halaman *Contacts*

3.2.5.1.6 Halaman *Login*

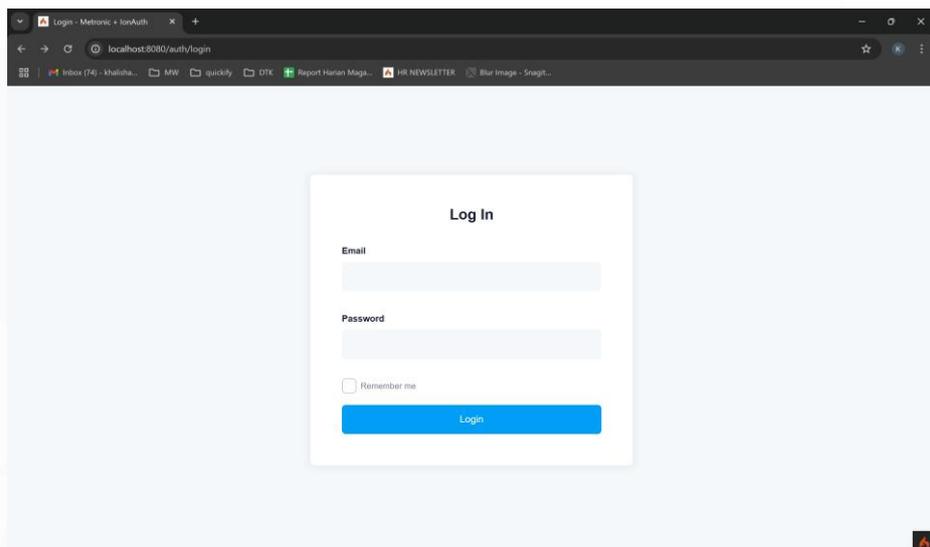
Terakhir, halaman *Login* dibuat untuk memastikan bahwa hanya pengguna yang memiliki hak akses yang dapat masuk ke dalam aplikasi *website*. Berbeda dengan halaman lain yang dibuat secara manual dan disimpan dalam folder *Views* aplikasi *website*, halaman *Login* dibuat dengan menggunakan *Ion Auth* untuk menangani proses autentikasi. *Ion Auth* merupakan *library* autentikasi untuk *framework* *CodeIgniter*. Sesuai saran dari mentor, *Ion Auth* digunakan karena sudah menyediakan fitur autentikasi lengkap dan mudah diintegrasikan.

Selain itu, *Ion Auth* juga telah menyediakan tampilan *login* bawaan (*built-in*), sehingga proses pembuatan tampilan *login* tidak perlu dilakukan secara manual. Untuk memastikan keamanan akses, diterapkan filter bernama *CheckLogin*, yang berfungsi untuk memeriksa apakah pengguna sudah *login* atau belum sebelum mengakses halaman tertentu.

```
app > Filters > CheckLogin.php > ...
3 namespace App\Filters;
4
5 use CodeIgniter\HTTP\RequestInterface;
6 use CodeIgniter\HTTP\ResponseInterface;
7 use CodeIgniter\Filters\FilterInterface;
8
9 class CheckLogin implements FilterInterface
10 {
11     public function before(RequestInterface $request, $arguments = null)
12     {
13         $ionAuth = new \IonAuth\Libraries\IonAuth();
14
15         if (!$ionAuth->loggedIn()) {
16             return redirect()->to('/auth/login');
17         }
18     }
19
20     public function after(RequestInterface $request, ResponseInterface $response, $arguments = null)
21     {
22     }
23 }
24
```

Gambar 3.77 Kode Halaman *Login*

Gambar 3.77 menunjukkan kode dari filter *CheckLogin*, di mana sistem akan secara otomatis mengarahkan (*redirect*) pengguna ke halaman *Login* (*/auth/login*) apabila pengguna belum melakukan *login*. Setelah pengguna berhasil *login*, sistem akan mengarahkan pengguna ke halaman yang sesuai dengan hak aksesnya. Hasil tampilan halaman *Login* dapat dilihat pada Gambar 3.78.



Gambar 3.78 Tampilan Halaman *Login*

3.2.5.1.7 Evaluasi

Setelah menyelesaikan implementasi prototipe pada CodeIgniter, perkembangan aplikasi disampaikan kepada mentor. Setelah *layout* dan alur kerja aplikasi *website* dijelaskan kepada mentor, mentor memberikan persetujuan awal dan selanjutnya mengadakan pertemuan dengan tim HR untuk menindaklanjuti pengembangan lebih lanjut. Dari hasil pertemuan tersebut, disampaikan bahwa secara umum *layout* aplikasi sudah sesuai, namun tim HR memberikan beberapa masukan, antara lain agar tampilan pada halaman *Work History* dapat membedakan antara *template* yang masih berstatus *draft* dan yang sudah dikirim. Selain itu, pada halaman *Dashboard* yang menampilkan tabel *Sent Newsletter*, tim HR menyarankan agar status pengiriman yang ditandai sebagai *failed* disertai dengan informasi atau keterangan yang menjelaskan alasan kegagalan tersebut.

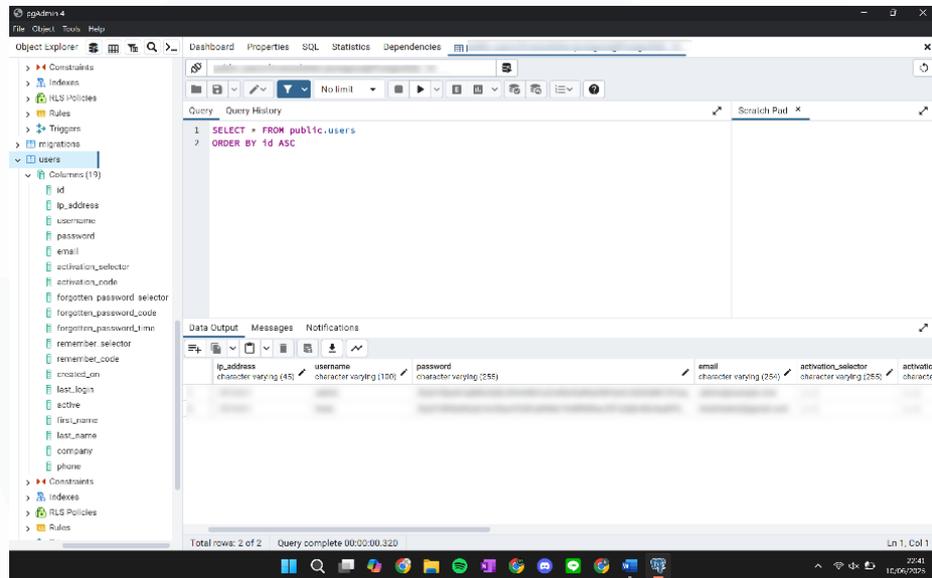
3.2.5.2 Integrasi Aplikasi *Website* dengan *Database*

Setelah menyelesaikan implementasi prototipe pada CodeIgniter 4, dilakukan integrasi aplikasi *website* yang telah dikembangkan dengan *database* yang telah dipilih sebelumnya, yaitu PostgreSQL. Pemilihan PostgreSQL didasarkan pada arahan dari mentor, dan pengelolaannya dilakukan menggunakan aplikasi pgAdmin 4. Sebelum pembuatan tabel dan pengisian data, terlebih dahulu dilakukan analisis kebutuhan sistem. Berdasarkan hasil diskusi dengan mentor, ditentukan bahwa terdapat tiga tabel utama yang diperlukan dalam aplikasi *website* ini.

3.2.5.2.1 Tabel Pengguna (*User*)

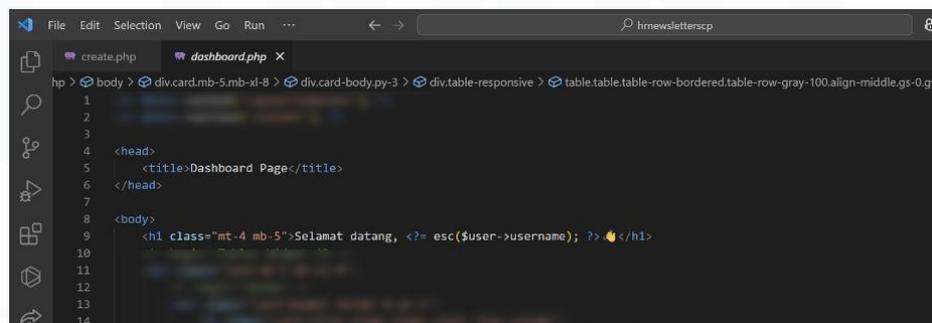
Tabel pertama yang dibutuhkan adalah tabel untuk menyimpan seluruh data pengguna. Data ini nantinya digunakan dalam proses autentikasi melalui halaman *Login*, serta sebagai identifikasi pengguna dalam sistem. Dengan menggunakan Ion Auth sebagai *library* autentikasi, tabel penyimpanan data pengguna telah tersedia secara

default oleh *library* tersebut, sehingga struktur data bawaan dari Ion Auth dimanfaatkan tanpa perlu membuat tabel baru secara manual.



Gambar 3.79 Tabel Pengguna pada pgAdmin 4

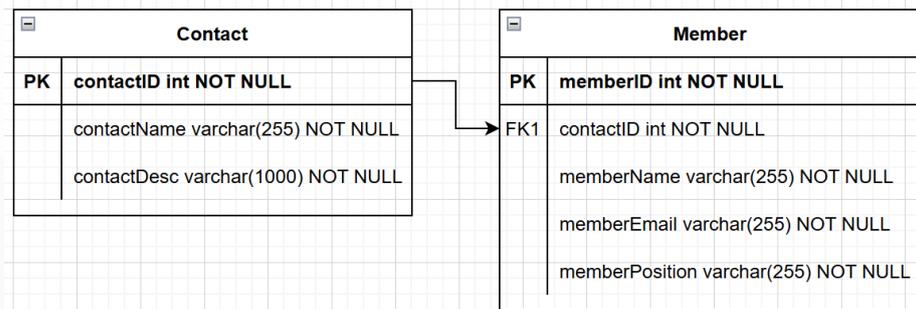
Pada Gambar 3.79, terlihat bahwa tabel *default* ini berisi beberapa kolom penting, seperti nama, *username*, *password*, dan kolom lainnya, yang ditampilkan di bagian sebelah kiri gambar. Selain untuk autentikasi, tabel ini juga digunakan untuk menampilkan informasi *username* di halaman *Dashboard*, sebagaimana ditunjukkan melalui pengimplementasian kode pada Gambar 3.80.



Gambar 3.80 Contoh Implementasi Kode untuk Integrasi *Database* pada Tabel Pengguna

3.2.5.2.2 Tabel *Contacts* dan *Members*

Tabel kedua yang dibutuhkan adalah *Contacts* dan *Members*, yang berfungsi untuk menyimpan seluruh data kontak pengguna beserta anggota dari masing-masing kontak. Gambar 3.81 menunjukkan struktur relasi antara kedua tabel tersebut.



Gambar 3.81 Struktur Relasi Tabel *Contacts* dan *Members*

Tabel *Contacts* berisi data seperti *id*, nama, dan deskripsi dari setiap kontak. Sementara itu, tabel *Members* menyimpan informasi berupa *id*, nama, email, posisi, serta *contactId* yang menghubungkan anggota dengan kontak terkait. Relasi ini memungkinkan sistem untuk mengetahui anggota mana yang termasuk dalam kontak tertentu. Kedua tabel ini digunakan untuk menampilkan data kontak beserta anggotanya pada seluruh bagian halaman *Contacts*, serta untuk memilih penerima *newsletter* pada halaman *Newsletter Workspace*, khususnya pada bagian kedua, yaitu *Select Recipient*.

```

57
58
59 <input type="text" name="contactName" class="form-control form-control-lg form-control-valid"
60 placeholder="Department name" value="<%= contact[ 'contactName' ] %>"/>
61
62
63
64
65
66
67
68 <label class="col-lg-4 col-form-label form-label" for="description">Description</label>
69
70
71
72 <input type="text" name="contactDescription" class="form-control form-control-lg form-control-valid"
73 placeholder="Description" value="<%= contact[ 'contactDesc' ] %>"/>
74
75
76
77
78
79
80
81 <div class="label-responsive">
82
83
84
85
86
87
88 <input type="checkbox" form check form check on form check custom form check select"
89 class="form-check-input widget-13-check" type="checkbox" value="1" />
90
91 <div>
92 <input checked="" type="checkbox" />
93 <input type="checkbox" />
94 <input checked="" type="checkbox" />
95 <input type="checkbox" />
96
97
98
99 <input type="checkbox" class="form-check-input widget-13-check" value="1" />
100
101
102
103

```

Gambar 3.82 Contoh Implementasi Kode untuk Integrasi Database pada Tabel *Contacts* dan *Members*

Contoh implementasi penggunaan tabel *Contacts* dan *Members* dapat dilihat pada bagian Edit *Contacts* (bagian ketiga dari halaman *Newsletter Workspace*). Gambar 3.82 menampilkan bagaimana tabel *Contacts* digunakan untuk menampilkan nama dan deskripsi kontak, sedangkan tabel *Members* digunakan untuk menampilkan seluruh anggota yang terkait dalam bentuk tabel.

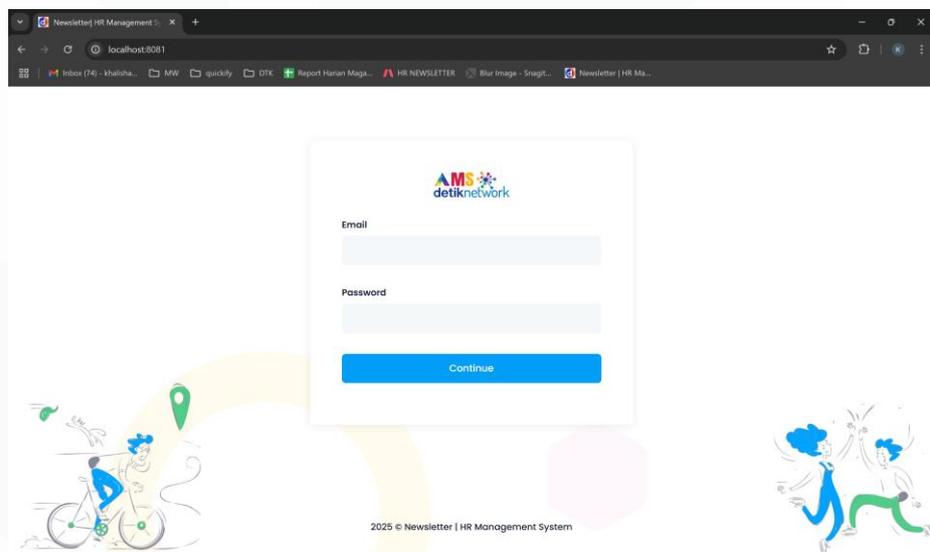
3.2.5.2.3 Tabel *Template*

Tabel ketiga yang dibutuhkan adalah tabel *Template*, yang berfungsi untuk menyimpan seluruh *template newsletter* yang dapat digunakan oleh pengguna. Tabel ini dirancang untuk memuat informasi seperti nama *template*, gambar pratinjau (*preview*), serta HTML dari masing-masing *template*. Data dari tabel ini digunakan pada halaman *Create Newsletter* untuk menampilkan pratinjau *template*, serta pada bagian pertama halaman *Newsletter Workspace*, yang memungkinkan pengguna memilih dan memodifikasi *template* sesuai kebutuhan tim HR. Namun, hingga tahap ini, pembuatan dan integrasi tabel *Template* belum dilakukan karena, sesuai arahan mentor, pengembangan diprioritaskan

pada tabel *Contacts* dan *Members* untuk menyelesaikan fitur utama seperti *Newsletter Workspace* dan *Contacts*.

3.2.5.2.4 Evaluasi

Setelah melakukan integrasi *database*, perkembangan terbaru disampaikan kepada mentor, dalam bentuk hasil pengembangan yang mencakup implementasi prototipe dan integrasi *database*. Setelah dilakukan pengecekan, mentor memberikan sejumlah masukan, termasuk perbaikan dan reorganisasi struktur proyek, seperti pembuatan folder baru yang di dalamnya sudah mencakup dua halaman, yaitu *Login* dan *Dashboard*, yang telah dipindahkan dan disesuaikan oleh mentor. Gambar 3.83 menunjukkan tampilan halaman *Login* setelah dipindahkan. Selain itu, mentor juga mengarahkan agar sistem *database* yang semula menggunakan PostgreSQL diganti menjadi MySQL.

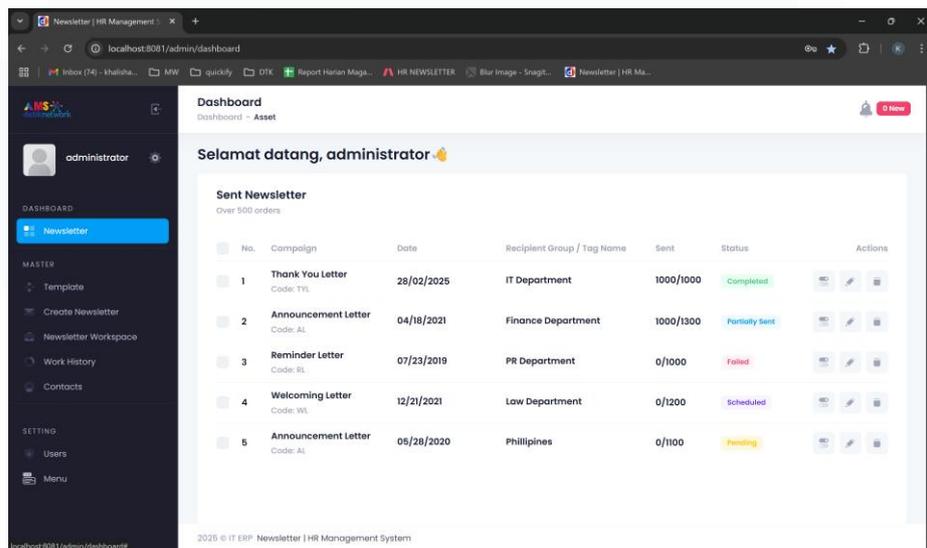


Gambar 3.83 Tampilan Halaman *Login* setelah Reorganisasi Struktur Folder

3.2.5.3 Reorganisasi Struktur Folder dan Implementasi Halaman ke Folder Baru

Setelah mentor selesai melakukan reorganisasi struktur folder, folder baru diserahkan untuk dilakukan penyesuaian implementasi. Salah satu halaman yang telah ditangani oleh mentor adalah halaman

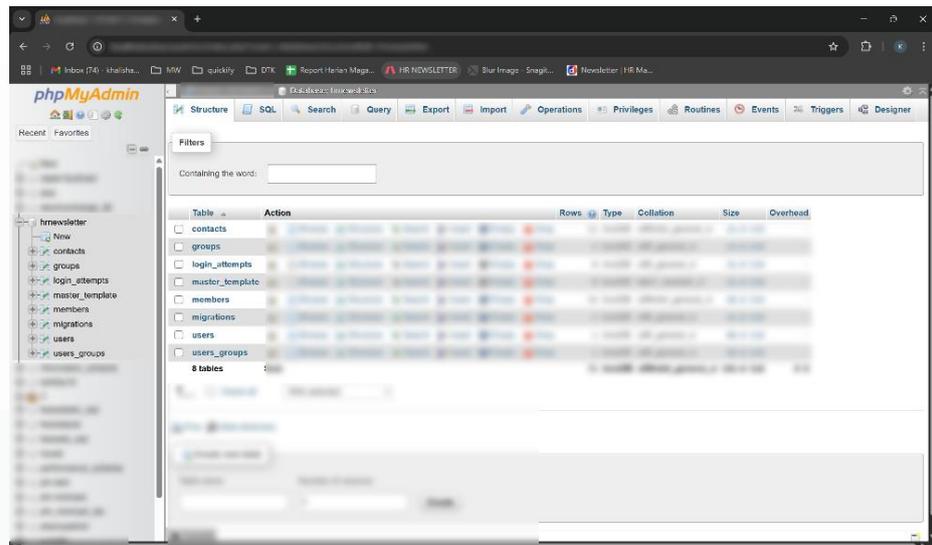
Dashboard. Meskipun demikian, modifikasi terhadap halaman tersebut tetap dilakukan, termasuk penambahan sapaan (*greetings*) yang menampilkan nama pengguna di bagian atas tabel *Sent Newsletter*. Data *username* ini diambil langsung dari tabel *user* dalam *database* sebagai hasil proses autentikasi saat *login*. Gambar 3.84 menunjukkan tampilan halaman *Dashboard* setelah dilakukan modifikasi.



Gambar 3.84 Tampilan Halaman *Dashboard* setelah Reorganisasi Struktur Folder

Selain itu, sesuai arahan mentor, sistem *database* yang semula menggunakan PostgreSQL diganti menjadi MySQL. Aplikasi XAMPP digunakan untuk menjalankan server lokal, sedangkan phpMyAdmin untuk mengelola *database* secara visual, sesuai dengan rekomendasi mentor untuk mempermudah proses pengelolaan dan pengecekan data. Gambar 3.85 menunjukkan tabel-tabel yang telah dipindahkan dan diatur ulang melalui phpMyAdmin.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

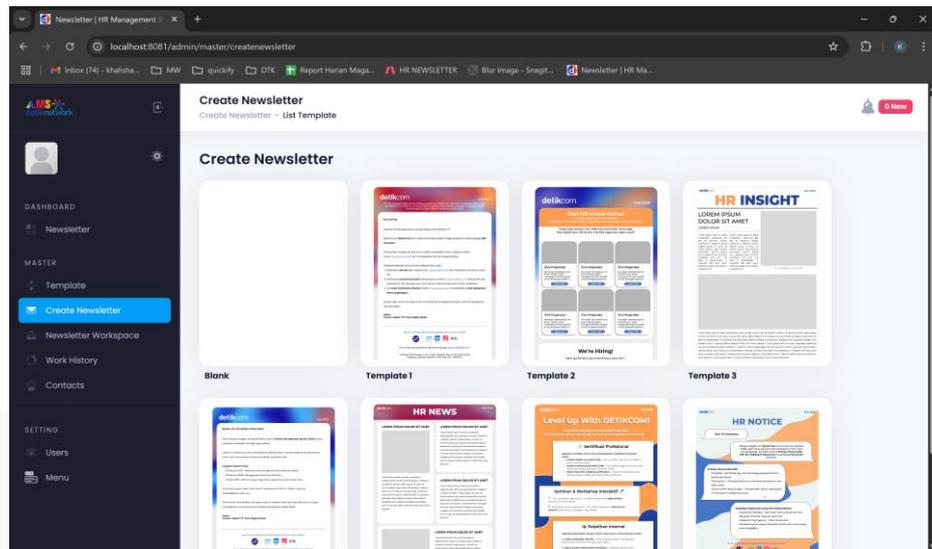


Gambar 3.85 Tampilan Tabel setelah Pemindahan dan Penataan Ulang melalui phpMyAdmin

Setelah struktur folder dan sistem *database* telah diperbarui, implementasi dilanjutkan pada empat halaman utama lainnya, yaitu: *Create Newsletter*, *Newsletter Workspace*, *Work History*, dan *Contacts*. Proses ini mencakup pemindahan *file* tampilan (*view*), pengaturan *controller*, serta penyesuaian konfigurasi agar halaman-halaman tersebut dapat berjalan sesuai dengan struktur proyek yang baru.

3.2.5.3.1 Halaman *Create Newsletter*

Halaman *Create Newsletter* telah dipindahkan ke dalam struktur folder baru tanpa mengalami perubahan pada tampilan maupun fungsionalitasnya. Halaman ini tetap menggunakan susunan *grid* untuk menampilkan daftar *template*, dengan gambar pratinjau yang ditambahkan secara manual. Kemudian, sesuai arahan mentor, implementasi halaman ini hanya dilakukan hingga tahap prototipe tanpa integrasi dengan *database*, karena pengembangan diarahkan untuk difokuskan terlebih dahulu pada fitur-fitur yang berkaitan langsung dengan proses pembuatan dan pengiriman email *newsletter*, yaitu halaman *Newsletter Workspace* dan *Contacts*. Gambar 3.86 memperlihatkan hasil pemindahan halaman ini ke dalam folder baru.



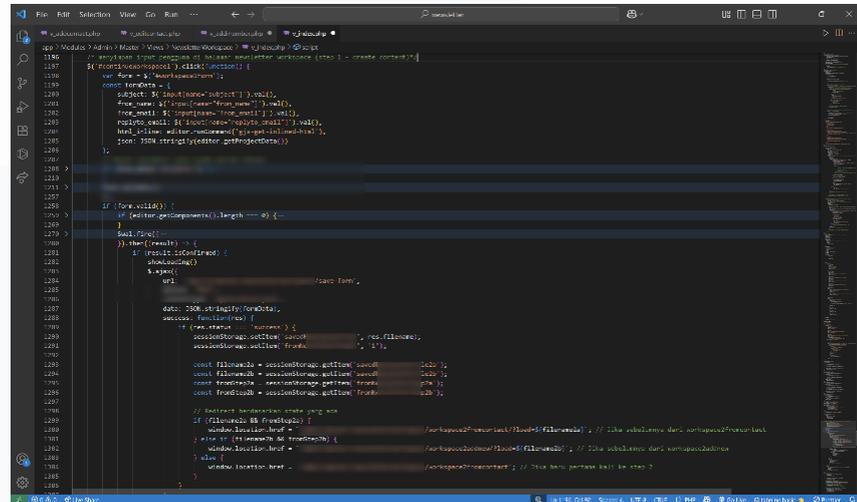
Gambar 3.86 Tampilan Halaman *Create Newsletter* setelah Reorganisasi Struktur Folder

3.2.5.3.2 Halaman *Newsletter Workspace*

Pemindahan halaman *Newsletter Workspace* ke dalam struktur folder baru disertai dengan sejumlah penyesuaian pada setiap komponennya. Halaman ini terdiri atas tiga bagian utama yang masing-masing mengalami modifikasi sesuai kebutuhan pengembangan.

1. Bagian pertama, yaitu *Create Content*, merupakan tampilan awal yang memungkinkan pengguna mengisi informasi pengiriman serta membuat konten *newsletter* sesuai kebutuhan. Pada bagian ini, dilakukan pemindahan ke struktur folder baru yang disertai beberapa modifikasi. Salah satunya adalah penambahan *plugin* *gis-preset-newsletter* ke dalam inisialisasi GrapesJS, sehingga hasil *output* HTML yang dihasilkan memiliki struktur utama berbasis elemen `<table>` dan menggunakan CSS *inline*. Penyesuaian ini penting karena email *client* seperti Gmail, yang digunakan dalam pengujian aplikasi, tidak mendukung penggunaan CSS terpisah dan memerlukan struktur HTML berbasis tabel agar konten dapat ditampilkan dengan semestinya. Selain itu, elemen *block* yang disediakan oleh *plugin* GrapesJS tersebut juga menyesuaikan

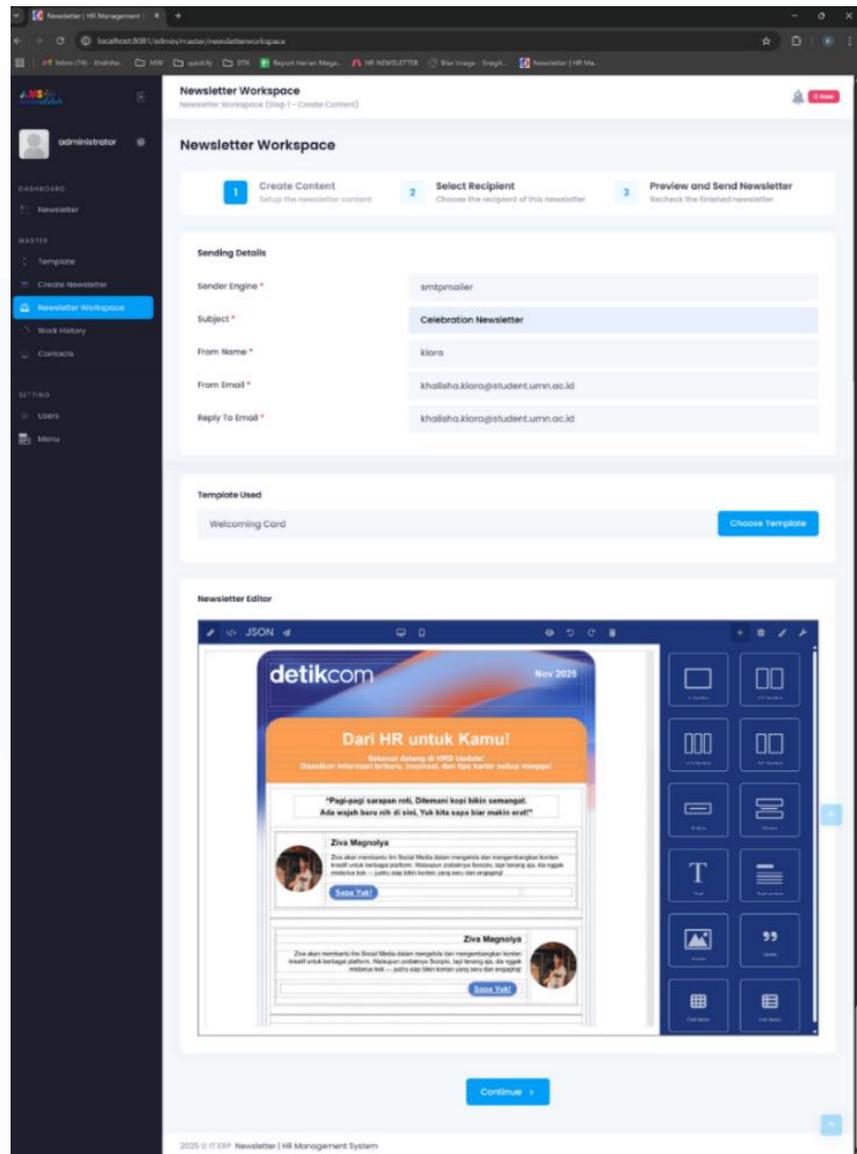
dengan struktur tersebut, di mana komponen-komponen dasar kini dibangun menggunakan format tabel.



```
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
```

Gambar 3.87 Kode Bagian Pertama (*Create Content*) dari *Newsletter Workspace* setelah Reorganisasi Struktur Folder

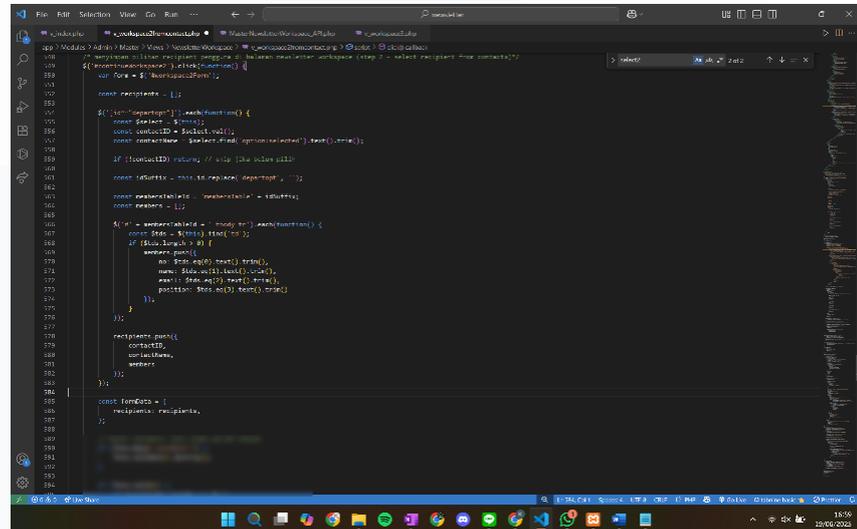
Modifikasi lainnya ditunjukkan pada Gambar 3.87, yaitu penambahan fungsi pada tombol *Continue* untuk menyimpan input pengguna secara lokal. Dengan adanya fungsi ini, data yang telah diisi akan tetap tersimpan ketika pengguna berpindah halaman, dan dapat ditampilkan kembali saat pengguna kembali ke halaman ini maupun pada bagian akhir, yaitu *Preview and Send Newsletter*. Selain itu, dilakukan penyesuaian tata letak dengan memisahkan bagian pemilihan *template* ke dalam subbagian tersendiri agar lebih terstruktur dan tidak tercampur dengan informasi pengiriman. Gambar 3.88 memperlihatkan perubahan tata letak serta hasil akhir dari bagian pertama halaman *Newsletter Workspace*.



Gambar 3.88 Tampilan Bagian Pertama (*Create Content*) dari *Newsletter Workspace* setelah Reorganisasi Struktur Folder

2. Bagian kedua, yaitu *Select Recipient*, merupakan tampilan untuk memilih penerima email (*recipient*), yang dapat dilakukan melalui dua metode, yaitu *Select from Contacts* dan *Add New Contact*. Pada bagian ini, dilakukan pemindahan struktur ke folder baru disertai dengan beberapa modifikasi. Salah satu modifikasi pada metode *Select from Contacts* adalah dilakukannya integrasi dengan *database* untuk menampilkan daftar kontak beserta anggota-

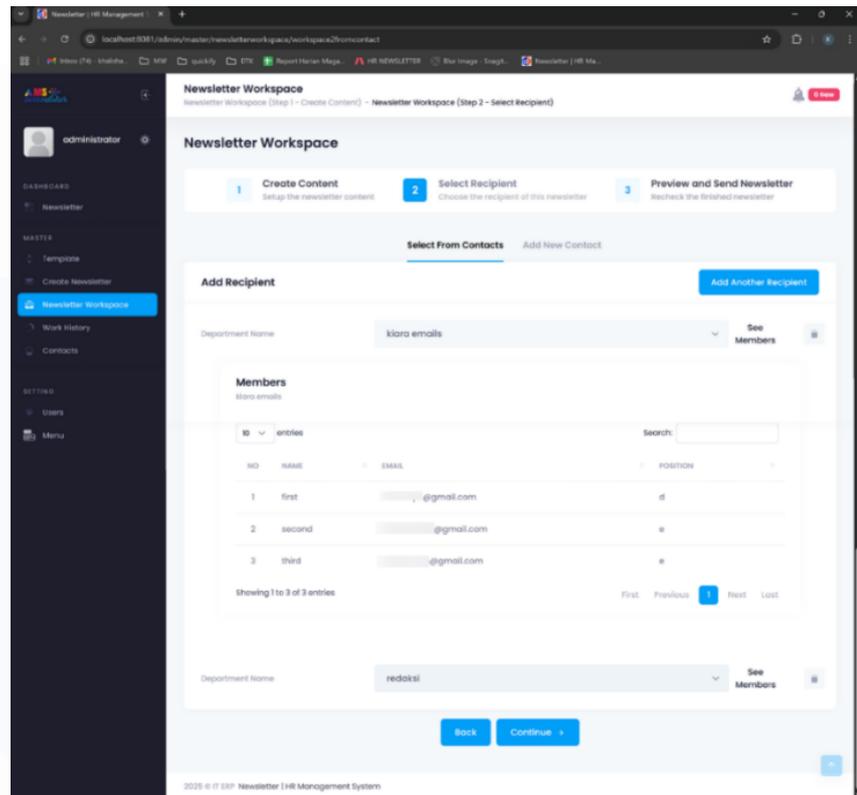
anggotanya melalui *dropdown*, yang ditampilkan menggunakan *plugin* DataTables dari jQuery.



```
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

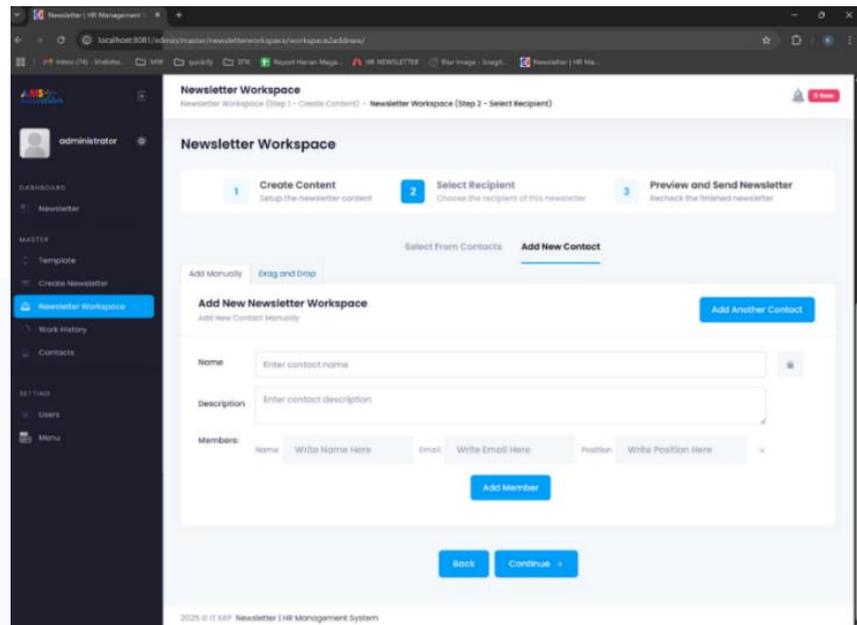
Gambar 3.89 Kode Bagian Kedua (*Select Recipient*) dari *Newsletter Workspace* dengan Pemilihan *Recipient* melalui Metode *Select From Contacts* setelah Reorganisasi Struktur Folder

Selain itu, pada metode *Select from Contacts*, dilakukan juga penambahan fungsi pada tombol *Continue*, yang memungkinkan penyimpanan lokal terhadap kontak yang dipilih oleh pengguna. Proses penyimpanan ini menggunakan pendekatan yang serupa dengan bagian pertama, yakni menyimpan data ke dalam *local storage*, namun berbeda pada jenis data yang disimpan karena fokus pada data kontak yang dipilih. Implementasi fungsi ini ditunjukkan pada Gambar 3.89, sedangkan hasil akhir dari metode *Select from Contacts* dapat dilihat pada Gambar 3.90.

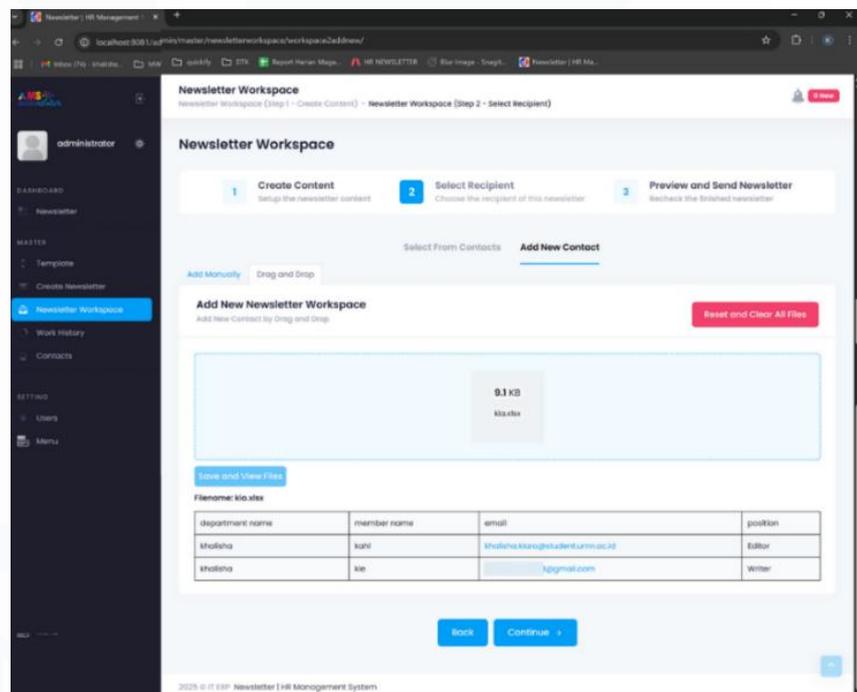


Gambar 3.90 Tampilan Halaman Bagian Kedua (*Select Recipient*) dari *Newsletter Workspace* dengan Pemilihan *Recipient* melalui Metode *Select From Contacts* setelah Reorganisasi Struktur Folder

Pada metode *Add New Contact*, dilakukan modifikasi pada proses penyimpanan data yang dimasukkan pengguna, baik melalui opsi *Add Manually* maupun *Drag and Drop*. Mekanisme penyimpanan pada kedua opsi tersebut serupa dengan metode *Select from Contacts*, yaitu menyimpan data ke *local storage*. Pada opsi *Drag and Drop*, proses penyimpanan lebih kompleks karena melibatkan pengambilan data dari *file Excel*, seperti nama dan deskripsi kontak, serta informasi anggota berupa nama, email, dan posisi. Data dari Excel ini perlu diorganisasi dan disimpan secara terstruktur, karena format kolomnya dapat beragam dan tidak berurutan. Gambar 3.91 menunjukkan implementasi kode untuk penyimpanan data dari opsi *Drag and Drop*.



Gambar 3.92 Tampilan Halaman Bagian Kedua (*Select Recipient*) dari *Newsletter Workspace* dengan Pemilihan *Recipient* melalui Metode *Add New Contacts* (opsi *Add Manually*) setelah Reorganisasi Struktur Folder

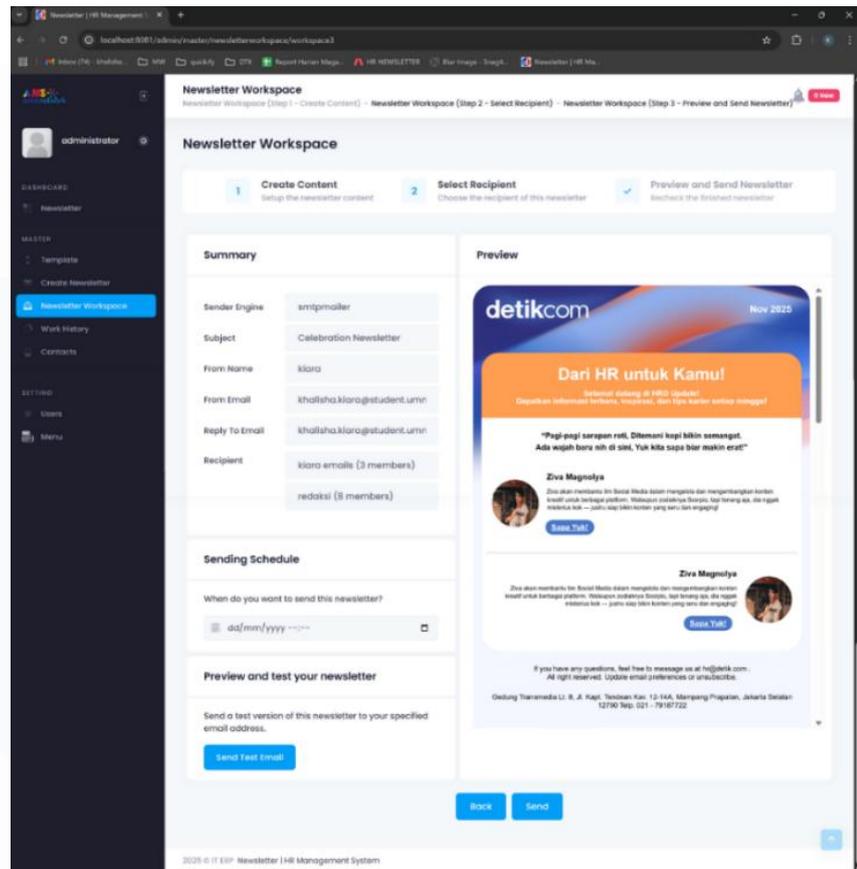


Gambar 3.93 Tampilan Halaman Bagian Kedua (*Select Recipient*) dari *Newsletter Workspace* dengan Pemilihan *Recipient* melalui Metode *Add New Contacts* (opsi *Drag and Drop*) setelah Reorganisasi Struktur Folder

3. Bagian ketiga, yaitu *Preview and Send Newsletter*, merupakan tampilan yang berfungsi sebagai tahap akhir untuk meninjau

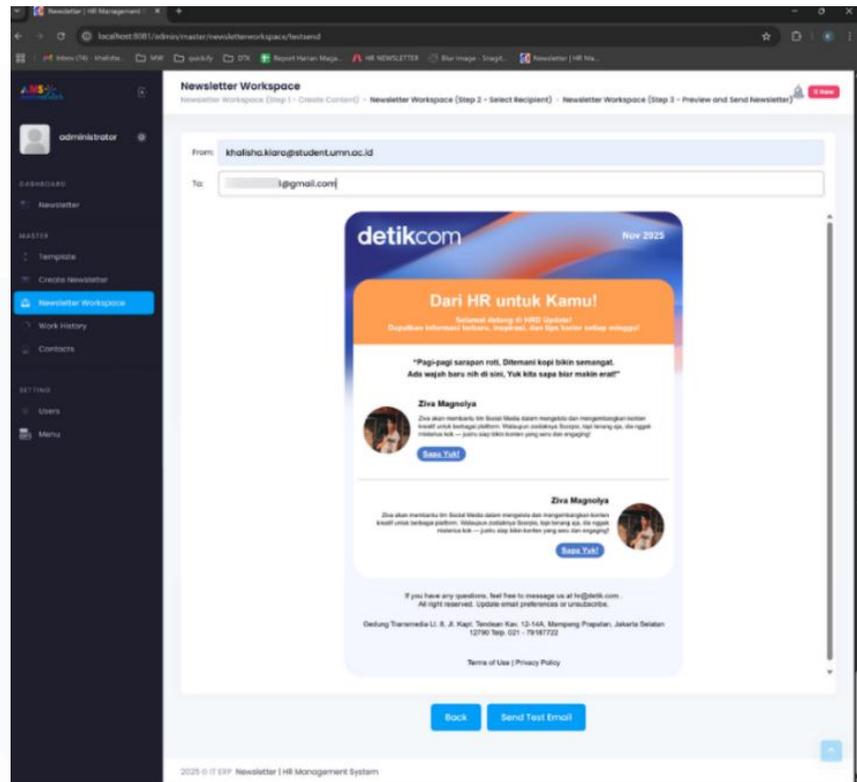
keseluruhan data yang telah diisi oleh pengguna sebelum mengirimkan *newsletter*. Pada bagian ini, pemindahan struktur ke folder baru dilakukan bersamaan dengan penerapan sejumlah modifikasi. Salah satu modifikasi yang dilakukan adalah penggunaan AJAX untuk menampilkan data yang berasal dari bagian pertama dan kedua halaman *Newsletter Workspace*. Dalam proses ini, struktur kondisi *if* digunakan untuk membedakan data yang berasal dari bagian pertama dan bagian kedua. Khusus untuk bagian kedua, ditambahkan juga kondisi *if-else* untuk membedakan data berdasarkan metode input yang digunakan, yaitu *Select from Contact* atau *Add New Contact*. Hal ini memungkinkan sistem menampilkan data yang sesuai dengan cara input yang dipilih oleh pengguna, sehingga proses pratinjau pada halaman *Preview and Send Newsletter* menjadi lebih akurat dan terstruktur. Implementasi kode dari modifikasi ini dapat dilihat pada Gambar 3.94.





Gambar 3.95 Tampilan Bagian Ketiga (*Preview and Send Newsletter*) dari *Newsletter Workspace*

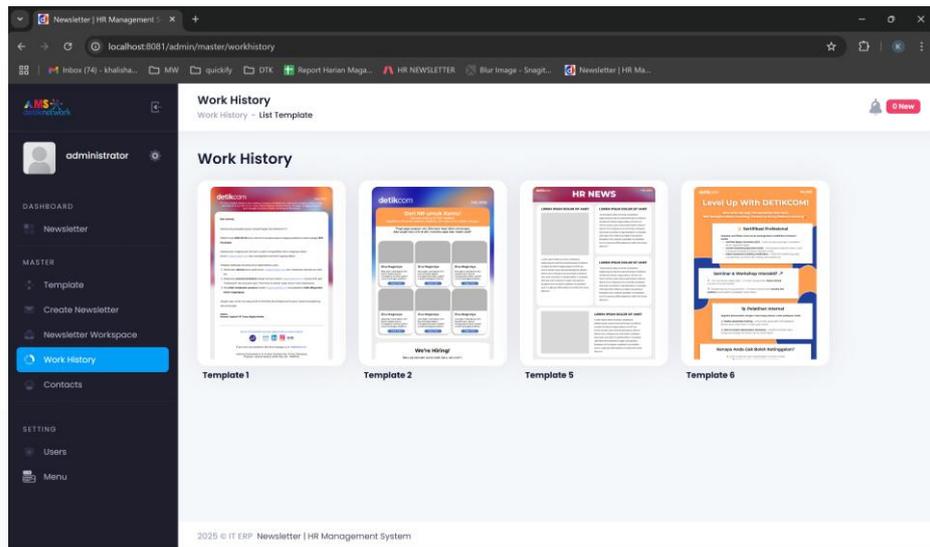
Dalam proses pengerjaan bagian ketiga, mentor mengarahkan untuk membuat halaman tambahan yang dapat digunakan untuk menguji pengiriman email *newsletter*. Menindaklanjuti arahan tersebut, dilakukan perancangan dan pengimplementasian halaman *Test Send Email* yang memungkinkan pengguna melihat *preview* keseluruhan *newsletter* yang telah dibuat, serta mengisi alamat email pengirim dan penerima untuk keperluan uji coba. Gambar 3.96 memperlihatkan tampilan halaman *Test Send Email* yang telah dikembangkan.



Gambar 3.96 Tampilan Bagian Ketiga (*Preview and Send Newsletter*) dari *Newsletter Workspace*

3.2.5.3.3 Halaman *Work History*

Sama seperti pada halaman *Create Newsletter*, pemindahan halaman *Work History* ke dalam struktur folder baru dilakukan tanpa perubahan pada tampilan maupun fungsionalitasnya. Implementasi halaman ini juga hanya dilakukan hingga tahap prototipe, karena pengembangan diarahkan terlebih dahulu pada fitur yang berkaitan langsung dengan proses pembuatan dan pengiriman email *newsletter*, yaitu halaman *Newsletter Workspace* dan *Contacts*. Halaman ini tetap menggunakan susunan *grid* untuk menampilkan pratinjau setiap *template*, dengan gambar yang dimasukkan secara manual karena integrasi dengan *database* belum diterapkan. Gambar 3.97 memperlihatkan tampilan halaman *Work History* setelah dipindahkan ke folder baru.



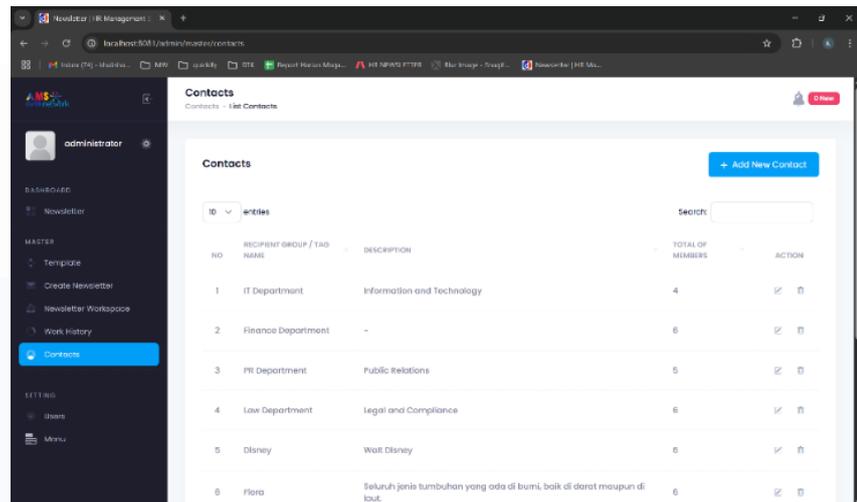
Gambar 3.97 Tampilan Halaman *Work History* setelah Reorganisasi Struktur Folder

3.2.5.3.4 Halaman *Contacts*

Pemindahan halaman *Contacts* dilakukan dengan disertai beberapa modifikasi pada masing-masing bagiannya. Halaman ini terbagi menjadi lima bagian, yaitu:

1. Bagian pertama yaitu *Contact List*, yang merupakan tampilan awal yang menampilkan seluruh data kontak. Pada bagian ini, dilakukan pemindahan ke struktur folder baru disertai penyesuaian pada tabel, berupa integrasi dengan *database*. Integrasi tabel *contacts* digunakan untuk menampilkan informasi seperti *Recipient Group/Tag Name* dan *Description*, sedangkan integrasi tabel *members* dimanfaatkan untuk menghitung jumlah anggota (*members*) yang memiliki relasi dengan setiap kontak melalui *contactId*.

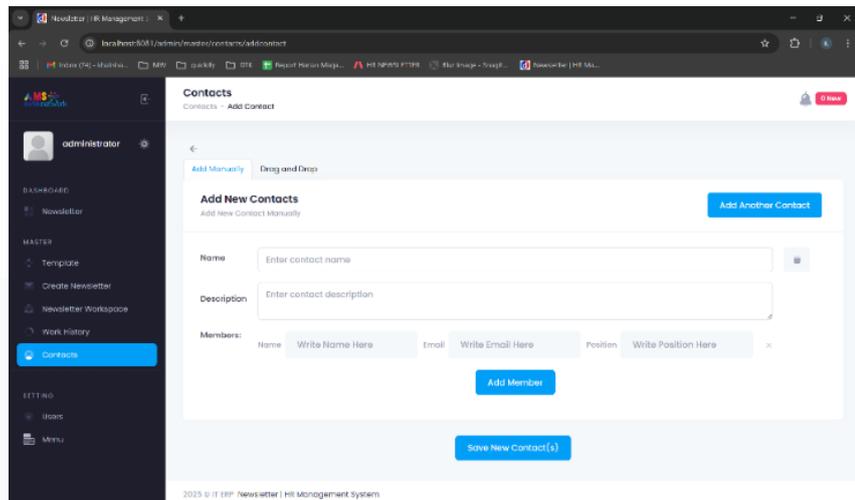
UNIVERSITAS
MULTIMEDIA
NUSANTARA



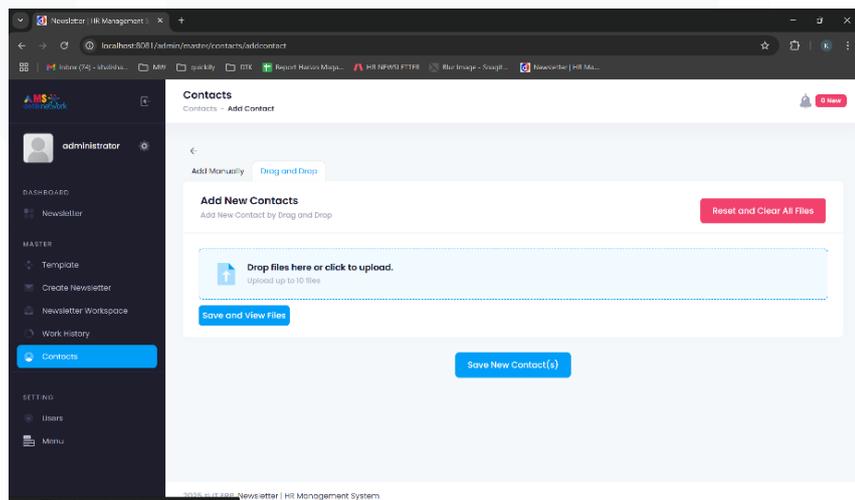
Gambar 3.99 Tampilan Bagian Pertama (*Contact List*) dari Halaman *Contacts* setelah Reorganisasi Struktur Folder

2. Bagian kedua, yaitu *Add Contact*, merupakan tampilan yang digunakan untuk menambahkan kontak beserta anggota di dalamnya. Pada bagian ini, dilakukan pemindahan ke struktur folder baru disertai dengan sejumlah modifikasi. Salah satu modifikasi tersebut adalah pengembangan tombol ‘*Save New Contact*’ pada metode *add manually*, dengan menambahkan validasi menggunakan fungsi *validate* dari *library* jQuery Validation, yang dilengkapi dengan *rules* dinamis. Validasi ini bertujuan untuk memastikan bahwa setiap input yang ditambahkan dan diisi oleh pengguna dapat dicek secara individual. Selain itu, ditambahkan pula konfirmasi menggunakan *swal.fire* dari *library* SweetAlert2 untuk memberikan umpan balik visual kepada pengguna sebelum data benar-benar disimpan.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

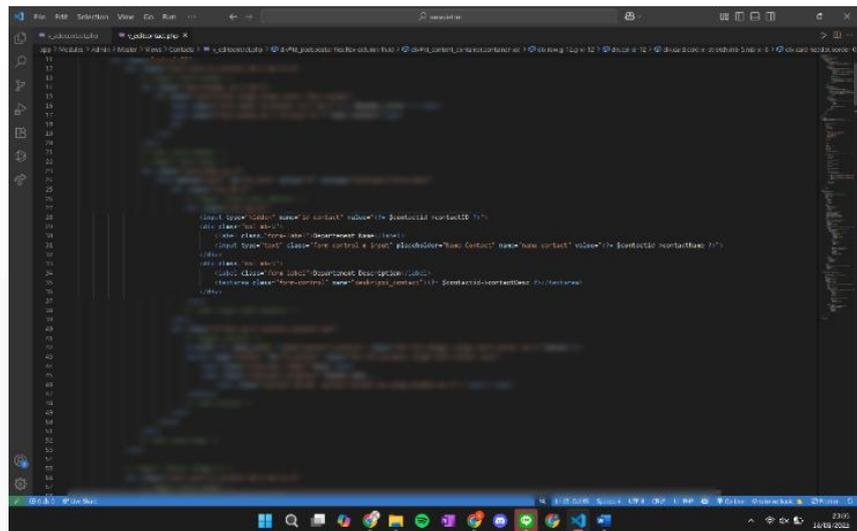


Gambar 3.101 Tampilan Bagian Kedua (*Add Contact*) dari Halaman *Contacts* melalui Input Manual setelah Reorganisasi Struktur Folder



Gambar 3.102 Tampilan Bagian Kedua (*Add Contact*) dari Halaman *Contacts* melalui *Drag and Drop* setelah Reorganisasi Struktur Folder

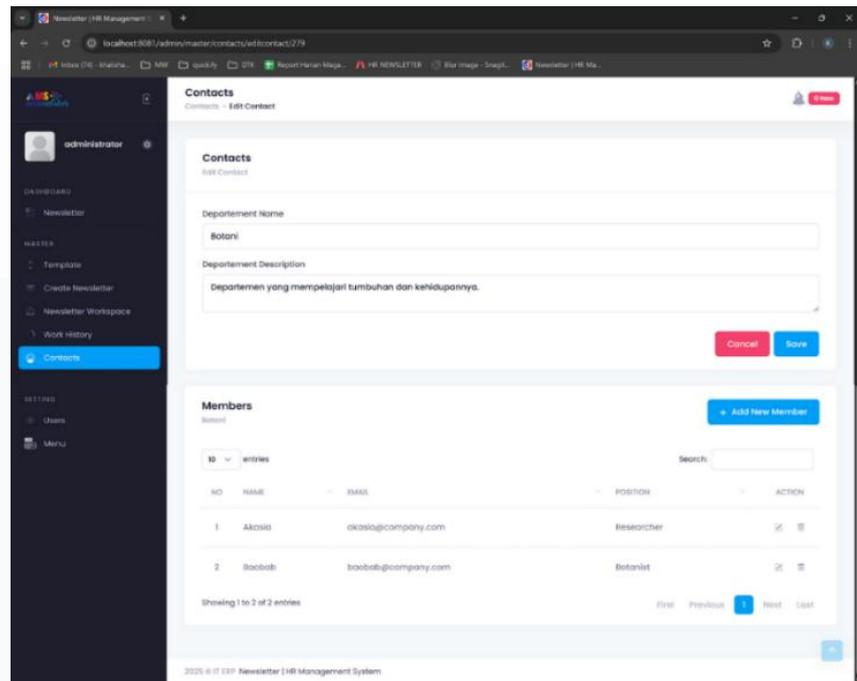
3. Bagian ketiga, yaitu *Edit Contact*, merupakan tampilan yang digunakan untuk mengubah informasi kontak, seperti nama dan deskripsi, serta menampilkan daftar anggota yang terkait dengan kontak tersebut. Pada bagian ini, dilakukan pemindahan ke struktur folder baru yang disertai dengan sejumlah modifikasi. Salah satu modifikasi utamanya adalah integrasi ke *database* agar proses pengeditan nama dan deskripsi dapat dilakukan secara langsung, sebagaimana ditunjukkan pada Gambar 3.103.



```
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Gambar 3.103 Kode Bagian Ketiga (Edit *Contact*) dari Halaman *Contacts* setelah Reorganisasi Struktur Folder

Selain itu, tabel anggota juga disesuaikan agar hanya menampilkan data anggota yang terhubung dengan kontak yang sedang diedit. Fungsi DataTable kembali digunakan untuk menampilkan data secara interaktif, seperti pada bagian pertama (*Contact List*). Validasi menggunakan jQuery Validation serta konfirmasi dengan SweetAlert2 juga diterapkan pada tombol simpan untuk memastikan input yang dimasukkan pengguna telah sesuai sebelum data disimpan ke dalam sistem. Tampilan akhir dari bagian ketiga (Edit *Contact*) dari halaman *Contacts* ditunjukkan pada Gambar 3.104.



Gambar 3.104 Tampilan Bagian Ketiga (Edit *Contact*) dari Halaman *Contacts* setelah Reorganisasi Struktur Folder

4. Bagian keempat, yaitu *Add Member*, merupakan tampilan yang digunakan untuk menambahkan anggota ke dalam suatu kontak. Pada bagian ini, dilakukan pemindahan ke struktur folder baru yang disertai dengan beberapa modifikasi. Seperti ditunjukkan pada Gambar 3.105, dilakukan modifikasi pada fitur penambahan anggota melalui metode *drag and drop* dengan menambahkan validasi input serta konfirmasi penyimpanan menggunakan *library* jQuery Validation dan SweetAlert2, yang diterapkan pada tombol "*Save and View Files*". Tombol ini juga dapat menampilkan data yang diunggah melalui Dropzone ke dalam sebuah tabel yang telah diberikan *class* tambahan untuk keperluan *styling*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

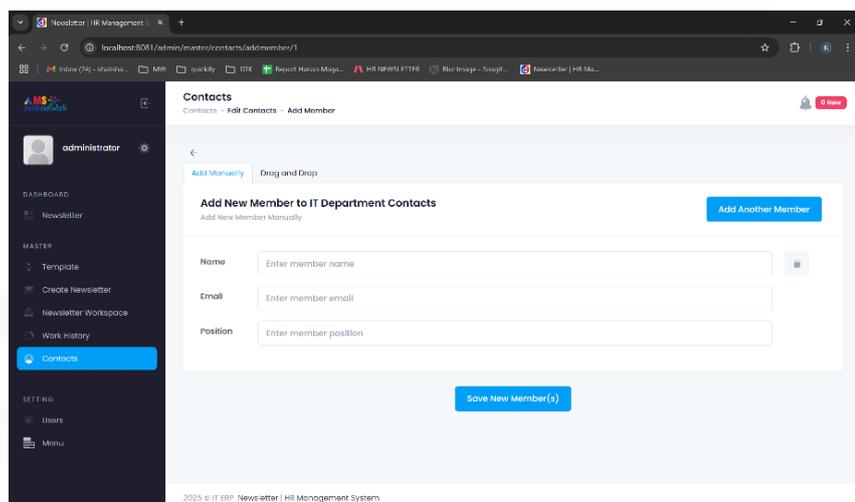
```

398 // @title: Add New Member
399 // @description: Add new member to the contacts list
400 // @tags: contacts, members
401 // @security: public
402 // @method: POST
403 // @url: /api/contacts/members
404 // @response: 201: {
405 //   "message": "Member added successfully",
406 //   "data": {
407 //     "id": 1,
408 //     "name": "John Doe",
409 //     "email": "john.doe@example.com",
410 //     "position": "Software Engineer",
411 //     "department": "IT",
412 //     "created_at": "2023-10-27T10:00:00Z",
413 //     "updated_at": "2023-10-27T10:00:00Z"
414 //   }
415 // }
416 // @response: 400: {
417 //   "message": "Invalid input",
418 //   "errors": {
419 //     "name": "The name field is required.",
420 //     "email": "The email field is required and must be a valid email address.",
421 //     "position": "The position field is required.",
422 //     "department": "The department field is required."
423 //   }
424 // }
425 // @response: 500: {
426 //   "message": "Internal server error",
427 //   "error": "An unexpected error occurred while processing your request."
428 // }
429 // @response: 401: {
430 //   "message": "Unauthorized",
431 //   "error": "You are not authorized to perform this action."
432 // }
433 // @response: 403: {
434 //   "message": "Forbidden",
435 //   "error": "You do not have permission to perform this action."
436 // }
437 // @response: 404: {
438 //   "message": "Not Found",
439 //   "error": "The resource you are looking for does not exist."
440 // }
441 // @response: 502: {
442 //   "message": "Bad Gateway",
443 //   "error": "The server received an invalid response from the upstream server."
444 // }
445 // @response: 503: {
446 //   "message": "Service Unavailable",
447 //   "error": "The server is currently unavailable due to maintenance or high load."
448 // }
449 // @response: 504: {
450 //   "message": "Gateway Timeout",
451 //   "error": "The server did not receive a timely response from the upstream server."
452 // }
453 // @response: 505: {
454 //   "message": "HTTP Version Not Supported",
455 //   "error": "The server does not support the requested HTTP version."
456 // }
457 // @response: 507: {
458 //   "message": "Insufficient Storage",
459 //   "error": "The server has insufficient storage space to complete the request."
460 // }
461 // @response: 509: {
462 //   "message": "Bandwidth Limit Exceeded",
463 //   "error": "The server has exceeded its bandwidth limit."
464 // }
465 // @response: 511: {
466 //   "message": "Network Authentication Required",
467 //   "error": "The server requires network authentication."
468 // }
469 // @response: 520: {
470 //   "message": "Web Server is Down",
471 //   "error": "The web server is currently down."
472 // }
473 // @response: 522: {
474 //   "message": "Webbandwidth Exceeded",
475 //   "error": "The web server has exceeded its bandwidth limit."
476 // }
477 // @response: 523: {
478 //   "message": "Web Server is Overloaded",
479 //   "error": "The web server is currently overloaded."
480 // }
481 // @response: 524: {
482 //   "message": "Web Server is Shutting Down",
483 //   "error": "The web server is currently shutting down."
484 // }
485 // @response: 525: {
486 //   "message": "Web Server is Starting Up",
487 //   "error": "The web server is currently starting up."
488 // }
489 // @response: 526: {
490 //   "message": "Web Server is Restarting",
491 //   "error": "The web server is currently restarting."
492 // }
493 // @response: 527: {
494 //   "message": "Web Server is Stopping",
495 //   "error": "The web server is currently stopping."
496 // }
497 // @response: 528: {
498 //   "message": "Web Server is Upgrading",
499 //   "error": "The web server is currently upgrading."
500 // }
501 // @response: 529: {
502 //   "message": "Web Server is Down for Maintenance",
503 //   "error": "The web server is currently down for maintenance."
504 // }
505 // @response: 530: {
506 //   "message": "Web Server is Back Online",
507 //   "error": "The web server is back online."
508 // }
509 // @response: 531: {
510 //   "message": "Web Server is Under Maintenance",
511 //   "error": "The web server is currently under maintenance."
512 // }
513 // @response: 532: {
514 //   "message": "Web Server is Scheduled for Downtime",
515 //   "error": "The web server is scheduled for downtime."
516 // }
517 // @response: 533: {
518 //   "message": "Web Server is Unexpectedly Downtime",
519 //   "error": "The web server is experiencing unexpected downtime."
520 // }
521 // @response: 534: {
522 //   "message": "Web Server is Partially Downtime",
523 //   "error": "The web server is experiencing partial downtime."
524 // }
525 // @response: 535: {
526 //   "message": "Web Server is Fully Downtime",
527 //   "error": "The web server is experiencing full downtime."
528 // }
529 // @response: 536: {
530 //   "message": "Web Server is Partially Up",
531 //   "error": "The web server is partially up."
532 // }
533 // @response: 537: {
534 //   "message": "Web Server is Fully Up",
535 //   "error": "The web server is fully up."
536 // }
537 // @response: 538: {
538 //   "message": "Web Server is Restarting",
539 //   "error": "The web server is restarting."
540 // }
541 // @response: 539: {
542 //   "message": "Web Server is Stopping",
543 //   "error": "The web server is stopping."
544 // }
545 // @response: 540: {
546 //   "message": "Web Server is Starting Up",
547 //   "error": "The web server is starting up."
548 // }
549 // @response: 541: {
550 //   "message": "Web Server is Shutting Down",
551 //   "error": "The web server is shutting down."
552 // }
553 // @response: 542: {
554 //   "message": "Web Server is Upgrading",
555 //   "error": "The web server is upgrading."
556 // }
557 // @response: 543: {
558 //   "message": "Web Server is Down for Maintenance",
559 //   "error": "The web server is down for maintenance."
560 // }
561 // @response: 544: {
562 //   "message": "Web Server is Back Online",
563 //   "error": "The web server is back online."
564 // }
565 // @response: 545: {
566 //   "message": "Web Server is Under Maintenance",
567 //   "error": "The web server is under maintenance."
568 // }
569 // @response: 546: {
570 //   "message": "Web Server is Scheduled for Downtime",
571 //   "error": "The web server is scheduled for downtime."
572 // }
573 // @response: 547: {
574 //   "message": "Web Server is Unexpectedly Downtime",
575 //   "error": "The web server is experiencing unexpected downtime."
576 // }
577 // @response: 548: {
578 //   "message": "Web Server is Partially Downtime",
579 //   "error": "The web server is experiencing partial downtime."
580 // }
581 // @response: 549: {
582 //   "message": "Web Server is Fully Downtime",
583 //   "error": "The web server is experiencing full downtime."
584 // }
585 // @response: 550: {
586 //   "message": "Web Server is Partially Up",
587 //   "error": "The web server is partially up."
588 // }
589 // @response: 551: {
590 //   "message": "Web Server is Fully Up",
591 //   "error": "The web server is fully up."
592 // }
593 // @response: 552: {
594 //   "message": "Web Server is Restarting",
595 //   "error": "The web server is restarting."
596 // }
597 // @response: 553: {
598 //   "message": "Web Server is Stopping",
599 //   "error": "The web server is stopping."
600 // }
601 // @response: 554: {
602 //   "message": "Web Server is Starting Up",
603 //   "error": "The web server is starting up."
604 // }
605 // @response: 555: {
606 //   "message": "Web Server is Shutting Down",
607 //   "error": "The web server is shutting down."
608 // }
609 // @response: 556: {
610 //   "message": "Web Server is Upgrading",
611 //   "error": "The web server is upgrading."
612 // }
613 // @response: 557: {
614 //   "message": "Web Server is Down for Maintenance",
615 //   "error": "The web server is down for maintenance."
616 // }
617 // @response: 558: {
618 //   "message": "Web Server is Back Online",
619 //   "error": "The web server is back online."
620 // }
621 // @response: 559: {
622 //   "message": "Web Server is Under Maintenance",
623 //   "error": "The web server is under maintenance."
624 // }
625 // @response: 560: {
626 //   "message": "Web Server is Scheduled for Downtime",
627 //   "error": "The web server is scheduled for downtime."
628 // }
629 // @response: 561: {
630 //   "message": "Web Server is Unexpectedly Downtime",
631 //   "error": "The web server is experiencing unexpected downtime."
632 // }
633 // @response: 562: {
634 //   "message": "Web Server is Partially Downtime",
635 //   "error": "The web server is experiencing partial downtime."
636 // }
637 // @response: 563: {
638 //   "message": "Web Server is Fully Downtime",
639 //   "error": "The web server is experiencing full downtime."
640 // }
641 // @response: 564: {
642 //   "message": "Web Server is Partially Up",
643 //   "error": "The web server is partially up."
644 // }
645 // @response: 565: {
646 //   "message": "Web Server is Fully Up",
647 //   "error": "The web server is fully up."
648 // }
649 // @response: 566: {
650 //   "message": "Web Server is Restarting",
651 //   "error": "The web server is restarting."
652 // }
653 // @response: 567: {
654 //   "message": "Web Server is Stopping",
655 //   "error": "The web server is stopping."
656 // }
657 // @response: 568: {
658 //   "message": "Web Server is Starting Up",
659 //   "error": "The web server is starting up."
660 // }
661 // @response: 569: {
662 //   "message": "Web Server is Shutting Down",
663 //   "error": "The web server is shutting down."
664 // }
665 // @response: 570: {
666 //   "message": "Web Server is Upgrading",
667 //   "error": "The web server is upgrading."
668 // }
669 // @response: 571: {
670 //   "message": "Web Server is Down for Maintenance",
671 //   "error": "The web server is down for maintenance."
672 // }
673 // @response: 572: {
674 //   "message": "Web Server is Back Online",
675 //   "error": "The web server is back online."
676 // }
677 // @response: 573: {
678 //   "message": "Web Server is Under Maintenance",
679 //   "error": "The web server is under maintenance."
680 // }
681 // @response: 574: {
682 //   "message": "Web Server is Scheduled for Downtime",
683 //   "error": "The web server is scheduled for downtime."
684 // }
685 // @response: 575: {
686 //   "message": "Web Server is Unexpectedly Downtime",
687 //   "error": "The web server is experiencing unexpected downtime."
688 // }
689 // @response: 576: {
690 //   "message": "Web Server is Partially Downtime",
691 //   "error": "The web server is experiencing partial downtime."
692 // }
693 // @response: 577: {
694 //   "message": "Web Server is Fully Downtime",
695 //   "error": "The web server is experiencing full downtime."
696 // }
697 // @response: 578: {
698 //   "message": "Web Server is Partially Up",
699 //   "error": "The web server is partially up."
700 // }
701 // @response: 579: {
702 //   "message": "Web Server is Fully Up",
703 //   "error": "The web server is fully up."
704 // }
705 // @response: 580: {
706 //   "message": "Web Server is Restarting",
707 //   "error": "The web server is restarting."
708 // }
709 // @response: 581: {
710 //   "message": "Web Server is Stopping",
711 //   "error": "The web server is stopping."
712 // }
713 // @response: 582: {
714 //   "message": "Web Server is Starting Up",
715 //   "error": "The web server is starting up."
716 // }
717 // @response: 583: {
718 //   "message": "Web Server is Shutting Down",
719 //   "error": "The web server is shutting down."
720 // }
721 // @response: 584: {
722 //   "message": "Web Server is Upgrading",
723 //   "error": "The web server is upgrading."
724 // }
725 // @response: 585: {
726 //   "message": "Web Server is Down for Maintenance",
727 //   "error": "The web server is down for maintenance."
728 // }
729 // @response: 586: {
730 //   "message": "Web Server is Back Online",
731 //   "error": "The web server is back online."
732 // }
733 // @response: 587: {
734 //   "message": "Web Server is Under Maintenance",
735 //   "error": "The web server is under maintenance."
736 // }
737 // @response: 588: {
738 //   "message": "Web Server is Scheduled for Downtime",
739 //   "error": "The web server is scheduled for downtime."
740 // }
741 // @response: 589: {
742 //   "message": "Web Server is Unexpectedly Downtime",
743 //   "error": "The web server is experiencing unexpected downtime."
744 // }
745 // @response: 590: {
746 //   "message": "Web Server is Partially Downtime",
747 //   "error": "The web server is experiencing partial downtime."
748 // }
749 // @response: 591: {
750 //   "message": "Web Server is Fully Downtime",
751 //   "error": "The web server is experiencing full downtime."
752 // }
753 // @response: 592: {
754 //   "message": "Web Server is Partially Up",
755 //   "error": "The web server is partially up."
756 // }
757 // @response: 593: {
758 //   "message": "Web Server is Fully Up",
759 //   "error": "The web server is fully up."
760 // }
761 // @response: 594: {
762 //   "message": "Web Server is Restarting",
763 //   "error": "The web server is restarting."
764 // }
765 // @response: 595: {
766 //   "message": "Web Server is Stopping",
767 //   "error": "The web server is stopping."
768 // }
769 // @response: 596: {
770 //   "message": "Web Server is Starting Up",
771 //   "error": "The web server is starting up."
772 // }
773 // @response: 597: {
774 //   "message": "Web Server is Shutting Down",
775 //   "error": "The web server is shutting down."
776 // }
777 // @response: 598: {
778 //   "message": "Web Server is Upgrading",
779 //   "error": "The web server is upgrading."
780 // }
781 // @response: 599: {
782 //   "message": "Web Server is Down for Maintenance",
783 //   "error": "The web server is down for maintenance."
784 // }
785 // @response: 600: {
786 //   "message": "Web Server is Back Online",
787 //   "error": "The web server is back online."
788 // }
789 // @response: 601: {
790 //   "message": "Web Server is Under Maintenance",
791 //   "error": "The web server is under maintenance."
792 // }
793 // @response: 602: {
794 //   "message": "Web Server is Scheduled for Downtime",
795 //   "error": "The web server is scheduled for downtime."
796 // }
797 // @response: 603: {
798 //   "message": "Web Server is Unexpectedly Downtime",
799 //   "error": "The web server is experiencing unexpected downtime."
800 // }
801 // @response: 604: {
802 //   "message": "Web Server is Partially Downtime",
803 //   "error": "The web server is experiencing partial downtime."
804 // }
805 // @response: 605: {
806 //   "message": "Web Server is Fully Downtime",
807 //   "error": "The web server is experiencing full downtime."
808 // }
809 // @response: 606: {
810 //   "message": "Web Server is Partially Up",
811 //   "error": "The web server is partially up."
812 // }
813 // @response: 607: {
814 //   "message": "Web Server is Fully Up",
815 //   "error": "The web server is fully up."
816 // }
817 // @response: 608: {
818 //   "message": "Web Server is Restarting",
819 //   "error": "The web server is restarting."
820 // }
821 // @response: 609: {
822 //   "message": "Web Server is Stopping",
823 //   "error": "The web server is stopping."
824 // }
825 // @response: 610: {
826 //   "message": "Web Server is Starting Up",
827 //   "error": "The web server is starting up."
828 // }
829 // @response: 611: {
830 //   "message": "Web Server is Shutting Down",
831 //   "error": "The web server is shutting down."
832 // }
833 // @response: 612: {
834 //   "message": "Web Server is Upgrading",
835 //   "error": "The web server is upgrading."
836 // }
837 // @response: 613: {
838 //   "message": "Web Server is Down for Maintenance",
839 //   "error": "The web server is down for maintenance."
840 // }
841 // @response: 614: {
842 //   "message": "Web Server is Back Online",
843 //   "error": "The web server is back online."
844 // }
845 // @response: 615: {
846 //   "message": "Web Server is Under Maintenance",
847 //   "error": "The web server is under maintenance."
848 // }
849 // @response: 616: {
850 //   "message": "Web Server is Scheduled for Downtime",
851 //   "error": "The web server is scheduled for downtime."
852 // }
853 // @response: 617: {
854 //   "message": "Web Server is Unexpectedly Downtime",
855 //   "error": "The web server is experiencing unexpected downtime."
856 // }
857 // @response: 618: {
858 //   "message": "Web Server is Partially Downtime",
859 //   "error": "The web server is experiencing partial downtime."
860 // }
861 // @response: 619: {
862 //   "message": "Web Server is Fully Downtime",
863 //   "error": "The web server is experiencing full downtime."
864 // }
865 // @response: 620: {
866 //   "message": "Web Server is Partially Up",
867 //   "error": "The web server is partially up."
868 // }
869 // @response: 621: {
870 //   "message": "Web Server is Fully Up",
871 //   "error": "The web server is fully up."
872 // }
873 // @response: 622: {
874 //   "message": "Web Server is Restarting",
875 //   "error": "The web server is restarting."
876 // }
877 // @response: 623: {
878 //   "message": "Web Server is Stopping",
879 //   "error": "The web server is stopping."
880 // }
881 // @response: 624: {
882 //   "message": "Web Server is Starting Up",
883 //   "error": "The web server is starting up."
884 // }
885 // @response: 625: {
886 //   "message": "Web Server is Shutting Down",
887 //   "error": "The web server is shutting down."
888 // }
889 // @response: 626: {
890 //   "message": "Web Server is Upgrading",
891 //   "error": "The web server is upgrading."
892 // }
893 // @response: 627: {
894 //   "message": "Web Server is Down for Maintenance",
895 //   "error": "The web server is down for maintenance."
896 // }
897 // @response: 628: {
898 //   "message": "Web Server is Back Online",
899 //   "error": "The web server is back online."
900 // }
901 // @response: 629: {
902 //   "message": "Web Server is Under Maintenance",
903 //   "error": "The web server is under maintenance."
904 // }
905 // @response: 630: {
906 //   "message": "Web Server is Scheduled for Downtime",
907 //   "error": "The web server is scheduled for downtime."
908 // }
909 // @response: 631: {
910 //   "message": "Web Server is Unexpectedly Downtime",
911 //   "error": "The web server is experiencing unexpected downtime."
912 // }
913 // @response: 632: {
914 //   "message": "Web Server is Partially Downtime",
915 //   "error": "The web server is experiencing partial downtime."
916 // }
917 // @response: 633: {
918 //   "message": "Web Server is Fully Downtime",
919 //   "error": "The web server is experiencing full downtime."
920 // }
921 // @response: 634: {
922 //   "message": "Web Server is Partially Up",
923 //   "error": "The web server is partially up."
924 // }
925 // @response: 635: {
926 //   "message": "Web Server is Fully Up",
927 //   "error": "The web server is fully up."
928 // }
929 // @response: 636: {
930 //   "message": "Web Server is Restarting",
931 //   "error": "The web server is restarting."
932 // }
933 // @response: 637: {
934 //   "message": "Web Server is Stopping",
935 //   "error": "The web server is stopping."
936 // }
937 // @response: 638: {
938 //   "message": "Web Server is Starting Up",
939 //   "error": "The web server is starting up."
940 // }
941 // @response: 639: {
942 //   "message": "Web Server is Shutting Down",
943 //   "error": "The web server is shutting down."
944 // }
945 // @response: 640: {
946 //   "message": "Web Server is Upgrading",
947 //   "error": "The web server is upgrading."
948 // }
949 // @response: 641: {
950 //   "message": "Web Server is Down for Maintenance",
951 //   "error": "The web server is down for maintenance."
952 // }
953 // @response: 642: {
954 //   "message": "Web Server is Back Online",
955 //   "error": "The web server is back online."
956 // }
957 // @response: 643: {
958 //   "message": "Web Server is Under Maintenance",
959 //   "error": "The web server is under maintenance."
960 // }
961 // @response: 644: {
962 //   "message": "Web Server is Scheduled for Downtime",
963 //   "error": "The web server is scheduled for downtime."
964 // }
965 // @response: 645: {
966 //   "message": "Web Server is Unexpectedly Downtime",
967 //   "error": "The web server is experiencing unexpected downtime."
968 // }
969 // @response: 646: {
970 //   "message": "Web Server is Partially Downtime",
971 //   "error": "The web server is experiencing partial downtime."
972 // }
973 // @response: 647: {
974 //   "message": "Web Server is Fully Downtime",
975 //   "error": "The web server is experiencing full downtime."
976 // }
977 // @response: 648: {
978 //   "message": "Web Server is Partially Up",
979 //   "error": "The web server is partially up."
980 // }
981 // @response: 649: {
982 //   "message": "Web Server is Fully Up",
983 //   "error": "The web server is fully up."
984 // }
985 // @response: 650: {
986 //   "message": "Web Server is Restarting",
987 //   "error": "The web server is restarting."
988 // }
989 // @response: 651: {
990 //   "message": "Web Server is Stopping",
991 //   "error": "The web server is stopping."
992 // }
993 // @response: 652: {
994 //   "message": "Web Server is Starting Up",
995 //   "error": "The web server is starting up."
996 // }
997 // @response: 653: {
998 //   "message": "Web Server is Shutting Down",
999 //   "error": "The web server is shutting down."
1000 // }

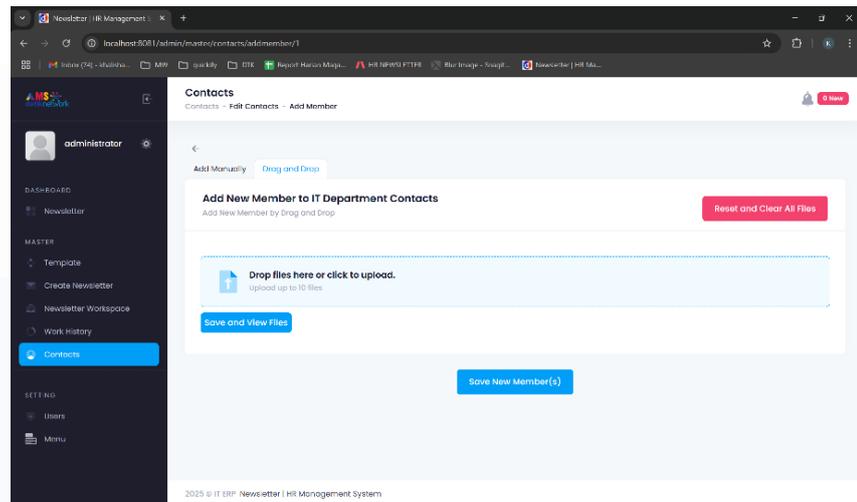
```

Gambar 3.105 Bagian Keempat (*Add Member*) dari Halaman *Contacts* setelah Reorganisasi Struktur Folder

Selain itu, ditambahkan juga pengaturan tambahan berupa penonaktifan tombol "Remove" pada Dropzone serta penguncian tombol "Save" dan fitur *drag and drop* setelah data berhasil dibaca dan disimpan oleh *controller*. Tujuan dari modifikasi ini adalah untuk mencegah pengguna melakukan perubahan terhadap *file* yang telah tersimpan. Tampilan akhir dari bagian *Add Member* dapat dilihat pada Gambar 3.106 dan Gambar 3.107.

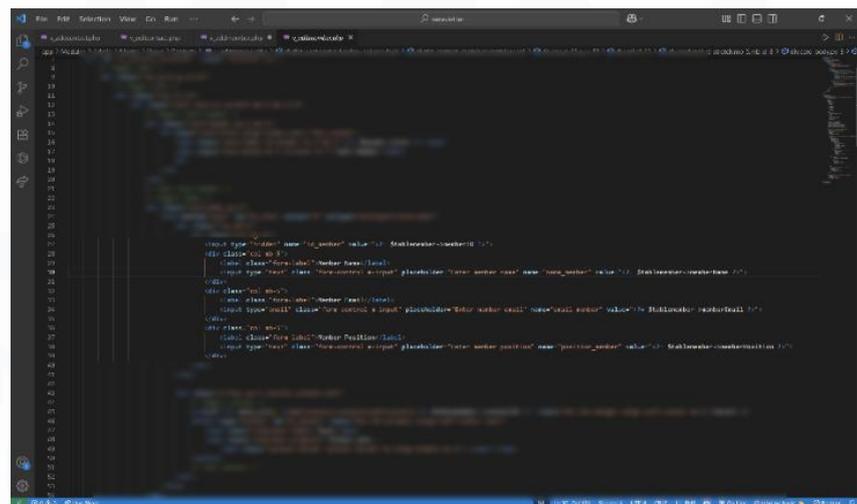


Gambar 3.106 Tampilan Bagian Keempat (*Add Member*) dari Halaman *Contacts* melalui Input Manual setelah Reorganisasi Struktur Folder



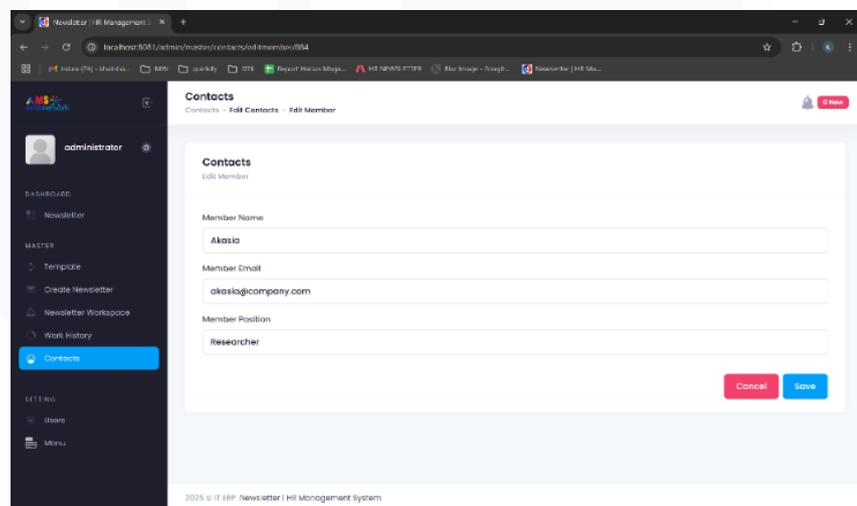
Gambar 3.107 Tampilan Bagian Keempat (*Add Member*) dari Halaman *Contacts* melalui *Drag and Drop* setelah Reorganisasi Struktur Folder

5. Bagian kelima, yaitu *Edit Member*, merupakan tampilan yang digunakan untuk mengubah informasi anggota, seperti nama, email, dan posisi. Pada bagian ini, dilakukan pemindahan ke struktur folder baru yang disertai dengan beberapa modifikasi, termasuk integrasi dengan *database* untuk menampilkan informasi anggota yang dapat diedit sesuai kebutuhan. Gambar 3.108 menunjukkan implementasi kode integrasi *database* yang dilakukan pada bagian kelima, yaitu *Edit Member*.



Gambar 3.108 Kode Bagian Kelima (*Edit Member*) dari Halaman *Contacts* setelah Reorganisasi Struktur Folder

Selain itu, dilakukan pula modifikasi pada tombol “Save” dengan menambahkan validasi dan konfirmasi menggunakan *library* jQuery Validation dan SweetAlert2, sebagaimana telah diterapkan pada bagian lain dalam halaman *Contacts*. Tujuan dari modifikasi ini adalah untuk memastikan bahwa data yang diubah telah sesuai sebelum disimpan ke dalam *database*. Hasil tampilan bagian kelima (Edit Member) dari halaman *Contacts* dapat dilihat pada Gambar 3.109.



Gambar 3.109 Tampilan Bagian Kelima (Edit Member) dari Halaman *Contacts* setelah Reorganisasi Struktur Folder

3.2.5.3.5 Evaluasi

Setelah proses reorganisasi struktur dan pemindahan halaman ke dalam folder baru diselesaikan, perkembangan implementasi aplikasi disampaikan kepada supervisor dan mentor. Berdasarkan hasil peninjauan, mentor memberikan arahan tambahan untuk menambahkan bagian khusus yang memungkinkan pengguna menyisipkan lampiran saat mengirimkan *newsletter* melalui email. Meskipun bagian khusus ini belum diterapkan, baik supervisor maupun mentor menyatakan bahwa hasil pengembangan sejauh ini telah sesuai dengan arahan yang diberikan. Selanjutnya, supervisor mengoordinasikan pertemuan dengan tim HR, yang kemudian dilaksanakan bersama dengan mentor.

Dalam pertemuan tersebut, tim HR memberikan respons yang positif terhadap aplikasi *website* yang telah dikembangkan. Secara umum, tata letak (*layout*) dan fitur editor *drag and drop* dinilai sudah sesuai. Namun, tim HR mengusulkan beberapa peningkatan, antara lain penambahan pilihan jenis *font* yang lebih beragam, termasuk *font* Montserrat yang merupakan standar penggunaan di lingkungan detikcom. Selain itu, mereka juga mengharapkan adanya fitur personalisasi pada teks yang digunakan dalam *newsletter*.

Fitur pengiriman email juga telah diuji dan berfungsi dengan baik. Namun, tim HR memberikan masukan untuk dilakukan riset lebih lanjut mengenai batasan pengiriman email dalam jumlah besar, khususnya terkait potensi email masuk ke folder spam dan kemungkinan akun Gmail pengirim terblokir (*disabled*). Hal ini menjadi fokus pengembangan lanjutan yang akan dilaksanakan selama sisa masa magang hingga bulan Agustus.

3.3 Kendala yang Ditemukan

Selama pelaksanaan magang dan pengembangan aplikasi HR *Newsletter* berbasis web, terdapat berbagai kendala yang dihadapi, baik dari aspek teknis maupun non-teknis. Beberapa di antaranya adalah sebagai berikut:

1. Kesulitan pada tahap awal pengembangan aplikasi karena sudah cukup lama tidak menggunakan bahasa pemrograman PHP dan JavaScript secara aktif, sehingga perlu mempelajarinya kembali untuk mendukung kelancaran proses pengembangan.
2. Belum familier dan belum memiliki pengalaman menggunakan *framework* CodeIgniter 4 serta *tools* pendukung seperti Composer dan Git Bash yang digunakan dalam pengembangan aplikasi, yang mengakibatkan diperlukannya waktu tambahan untuk mempelajari proses instalasi dan alur kerjanya.

3. Mengalami kesulitan dalam menggunakan aplikasi Figma karena sudah cukup lama tidak menggunakannya, maka perlu adanya penyesuaian diri kembali dengan fitur, *shortcut*, *plugin*, dan alur kerjanya.
4. Menghadapi tantangan dalam menggunakan PostgreSQL karena belum memiliki pengalaman langsung dalam pengoperasian sistem manajemen basis data tersebut, mulai dari instalasi hingga tahap perancangan dan pengelolaannya, sehingga perlu dilakukannya pembelajaran mandiri untuk dapat mengimplementasikannya secara efektif dalam pengembangan aplikasi.
5. Menghadapi tantangan dalam menindaklanjuti permintaan tim HR untuk menyediakan editor konten *newsletter* berbasis *drag and drop*, karena belum memiliki pengalaman dengan teknologi yang mendukung fitur tersebut, dan untuk itu, perlu dilakukannya riset terlebih dahulu untuk menemukan solusi yang sesuai dengan kebutuhan aplikasi.
6. Terjadi penundaan pada proses pembuatan dan pengintegrasian *database* tabel *template* di halaman *Create Newsletter* dan *Work History*, karena mentor pembimbing mengarahkan pengembangan agar terlebih dahulu difokuskan pada fitur utama, yaitu *Newsletter Workspace* dan *Contacts*, yang berkaitan langsung dengan pembuatan serta pengiriman email *newsletter*.

3.4 Solusi atas Kendala yang Ditemukan

Sebagai tindak lanjut atas kendala yang muncul selama kegiatan magang dan pengembangan aplikasi HR *Newsletter* berbasis web, diterapkan beberapa solusi berikut:

1. Mempelajari ulang konsep dasar PHP dan JavaScript melalui video pembelajaran YouTube seperti *channel* Web Programming UNPAS, karena menyajikan penjelasan yang runtut dan disertai contoh implementasi secara bertahap, dokumentasi resmi PHP dan JavaScript karena memberikan informasi langsung dari pengembang teknologi terkait, serta artikel seperti MDN Web Docs karena penyampaiannya sistematis dan mudah dipahami.

2. Untuk memahami *framework* CodeIgniter 4 serta *tools* pendukung seperti Composer dan Git Bash, dilakukan pembelajaran mandiri melalui video tutorial di YouTube seperti *channel* Web Programming UNPAS yang menjelaskan struktur, cara kerja, dan penerapannya. Selain itu, dokumentasi resmi dari *framework* dan *tools* pendukung juga dipelajari untuk mendapatkan panduan lengkap, disertai dengan pelaksanaan beberapa percobaan langsung terkait proyek yang dikembangkan.
3. Dalam penyesuaian kembali dengan aplikasi Figma, dilakukan pembelajaran melalui tutorial video di YouTube, salah satunya dari *channel* DesignWithArash, yang menyajikan penjelasan fitur, *shortcut*, *plugin*, dan alur kerja Figma dengan cara yang mudah dipahami. Selanjutnya, dilakukan penerapan secara langsung materi yang telah dipelajari, mencakup pembuatan *frame*, penggunaan komponen dan *plugin*, pemanfaatan fitur serta *shortcut*, hingga pengaturan *layout* halaman aplikasi.
4. Mempelajari PostgreSQL mulai dari tahap instalasi, pembuatan *database*, hingga pengelolaan data melalui video tutorial YouTube, seperti *channel* Database Star, yang menyajikan langkah-langkah secara jelas dan mudah dipahami. Selain itu, dokumentasi resmi PostgreSQL yang tersedia secara *online* juga dipelajari untuk memperdalam pemahaman.
5. Melakukan riset terhadap teknologi yang relevan untuk memenuhi kebutuhan editor konten berbasis *drag and drop*, dan menemukan bahwa GrapesJS merupakan solusi yang paling sesuai. GrapesJS dipilih karena bersifat *open-source*, memiliki fitur *visual builder drag and drop* yang lengkap, serta mendukung ekspor hasil desain ke HTML yang kompatibel dengan pengiriman email, sesuai dengan kebutuhan aplikasi HR *Newsletter*. Cara penggunaannya kemudian dipelajari melalui dokumentasi resmi GrapesJS sebagai panduan utama, tutorial di YouTube seperti *channel* Code Dexterous yang menyajikan contoh implementasi secara mudah dipahami, serta berbagai contoh proyek di platform seperti CodeSandbox untuk memahami integrasi GrapesJS ke dalam aplikasi secara praktis.

6. Sebagai tindak lanjut atas arahan dari mentor, jadwal pengembangan disesuaikan dengan memprioritaskan penyelesaian *halaman Newsletter Workspace* dan *Contacts* terlebih dahulu, sehingga pengembangan dapat berjalan lebih terarah dan fokus pada fitur-fitur utama yang mendukung proses pembuatan serta pengiriman email *newsletter*.

