

## BAB 3 PELAKSANAAN KERJA MAGANG

### 3.1 Kedudukan dan Koordinasi

Sesuai dengan Gambar 3.1, urutan kedudukan yang ada di posisi *Mobile Application Developer*. Posisi Ketua Umum Perkumpulan Maritim Muda Nusantara diduduki oleh Bapak Kaisar Akhir, selaku pendiri dari Perkumpulan Maritim Muda Nusantara. Posisi Wakil Ketua Umum Bidang Organisasi, Data, dan Advokasi Maritim dipegang oleh Bapak Putra Achdiyat Pratama, dan posisi Ketua Bidang Organisasi, Data, dan Transformasi Digital dipegang oleh Bapak Nasrullah Abdulrahman Rafli Junaid selaku mentor dan supervisor yang bertanggung jawab selama periode magang.



Gambar 3.1. Gambaran kedudukan dan koordinasi posisi *Mobile Application Developer*

Sesuai dengan Gambar 3.1, urutan kedudukan yang ada di posisi *Mobile Application Developer*. Posisi Ketua Umum Perkumpulan Maritim Muda Nusantara diduduki oleh Bapak Kaisar Akhir, selaku Founder dari Perkumpulan Maritim Muda Nusantara. Posisi Wakil Ketua Umum Bidang Organisasi, Data, dan

Advokasi Maritim dipegang oleh Bapak Putra Achdiyat Pratama, dan posisi Ketua Bidang Organisasi, Data, dan Transformasi Digital dipegang oleh Bapak Nasrullah Abdulrahman Rafli Junaid selaku mentor dan supervisor yang bertanggung jawab selama periode magang.

Dalam magang di Perkumpulan Maritim Muda Nusantara, penulis menjabat sebagai *Mobile Application Developer*. Selama proses magang, Tim *Mobile Application Developer* juga berkoordinasi dengan Bapak Kaisar Akhir sebagai Ketua Umum Maritim Muda yang memberikan arahan dalam pekerjaan dan Bapak Rafli sebagai mentor dan supervisor di tempat magang.

Dalam posisi *Mobile Application Developer* yang dipegang, posisi ini secara garis besar dibagi menjadi 2 tim, yaitu tim *Hybrid* dan tim WFH. Tim Hybrid fokus terhadap pengembangan aplikasi untuk klien Maritim Muda dan tim WFH fokus terhadap pengembangan dan *maintenance* aplikasi Maritim Muda Connect. Dalam tim WFH ini terdapat 4 anggota, yaitu: Darryl Clay Laya, Rivandi Mizwar, Muhammad Faza, Mohammad Nuafudin.

### **3.2 Tugas yang Dilakukan**

Tugas utama selama periode magang adalah *maintain* dan *update* terhadap aplikasi *mobile* Maritim Muda Connect. Maritim Muda Connect adalah aplikasi *mobile* yang merupakan versi *mobile* dari *website* Maritim Muda Hub. Tujuan utama dari aplikasi dan *website* ini adalah sebagai pusat informasi beragam untuk anggota yang telah terdaftar sebagai sahabat Maritim Muda. Anggota dari Maritim Muda Nusantara bisa dari luar pegawai utama (Anak Magang, Pejabat Organisasi). Aplikasi ini memberikan informasi beragam seperti informasi lowongan kerja, event terkait kemaritiman, beasiswa, dan juga produk/souvenir bagi anggota yang berminat. Identitas aplikasi sebagai pusat informasi memicu beberapa proyek peningkatan yang penulis kerjakan selama periode magang.

### **3.3 Uraian Pelaksanaan Magang**

Magang dilaksanakan di Perkumpulan Maritim Muda Nusantara secara *Work From Home* (WFH). Terdapat beberapa hari di mana diharuskan untuk mendatangi kantor Perkumpulan Maritim Muda Nusantara untuk bisa berdiskusi tatap muka tentang rencana kerja, atau untuk mengikuti rangkaian acara yang disiapkan oleh Perkumpulan Maritim Muda Nusantara seperti pelantikan untuk

pengurus organisasi selanjutnya, dll. Periode melaksanakan magang dimulai dari 30 Januari sampai 30 Juli di mana di tanggal 30 Januari dan 1 Februari merupakan hari singkat untuk melakukan *onboarding* dan perkenalan umum dari Perkumpulan Maritim Muda Nusantara melalui sarana *online*. Pekerjaan resmi dimulai di tanggal 3 Februari dan seterusnya sampai magang selesai pada 30 Juni.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

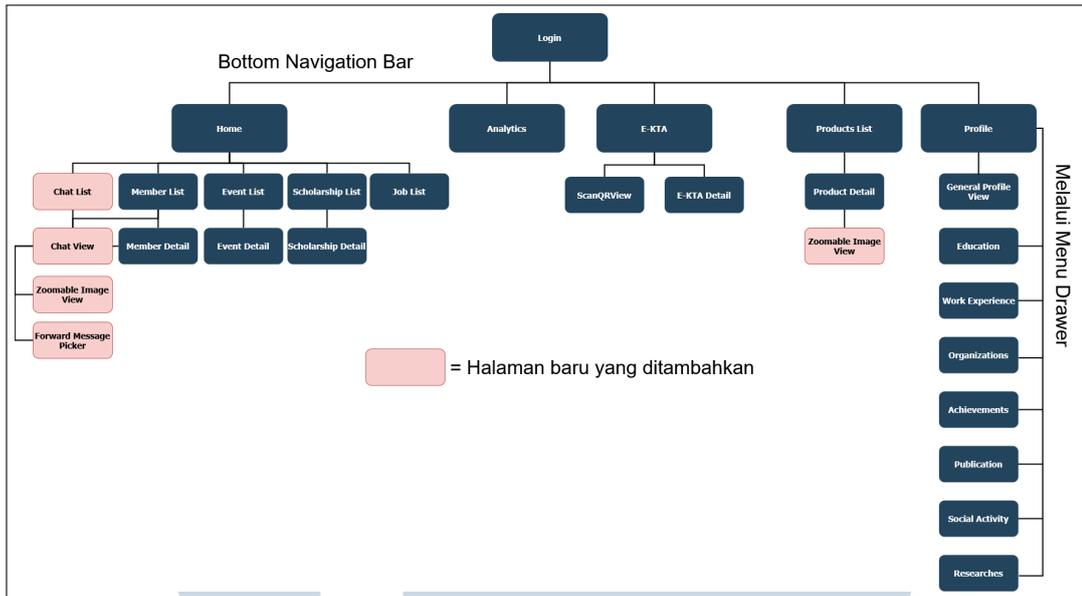
Minggu Ke -	Pekerjaan yang dilakukan
1	Pengenalan lingkungan dan garis besar magang
2-3	Mempelajari <i>Framework Flutter</i> , Installasi, dan cara desain dalam <i>Flutter</i>
4	Onboarding dengan <i>codebase</i> yang ada dan merencanakan untuk perbaruan aplikasi yang direncanakan
5	Melakukan update package dan riset untuk pengerjaan perbaruan yang telah direncanakan
6-10	Melakukan peningkatan UI/UX di berbagai halaman dan maintain aplikasi
11-Selesai	Melakukan pekerjaan untuk fitur <i>chat</i> antar anggota

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

### 3.4 Perancangan update aplikasi Maritim Muda Connect

Aplikasi Maritim Muda Connect yang menjadi tanggung jawab tim *Mobile App Developer* selama periode magang merupakan aplikasi yang dibangun menggunakan *framework Flutter*. *Flutter* adalah *framework* yang dikembangkan oleh Google untuk membangun aplikasi lintas *platform* dengan basis kode yang sama. *Flutter* dibangun menggunakan bahasa pemrograman *Dart*, yang memfokuskan pada pengembangan aplikasi dengan performa tinggi dan *User Interface* (UI) yang baik.

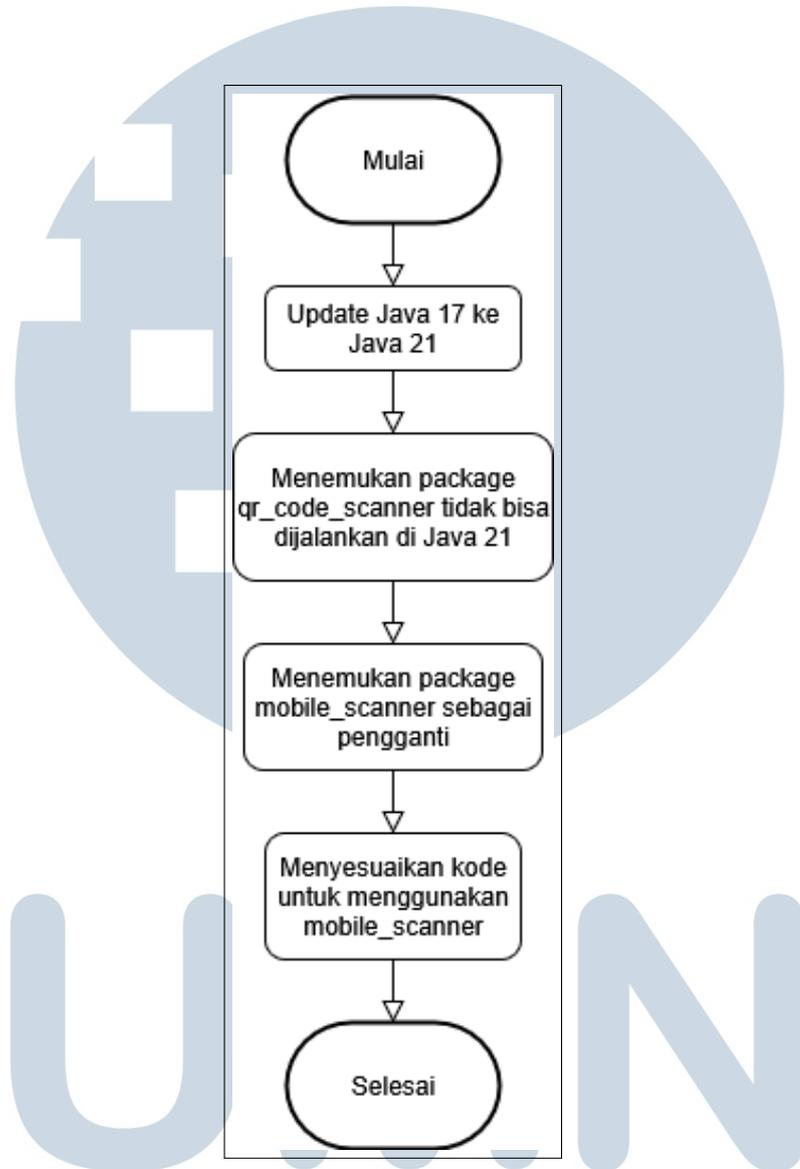
Aplikasi *Flutter* ini dibangun oleh tim pengembangan magang periode sebelumnya di tahun 2024 menggunakan *packageGetX*, *packageFlutter* yang memudahkan proses pengembangan aplikasi dengan pendekatan ringan dan mudah saat ingin melakukan *manajemen state*, *navigasi*, dan injeksi dependensi. Gambar 3.2 menggambarkan keseluruhan dari halaman yang ada di aplikasi Maritim Muda Connect.



Gambar 3.2. Site Map aplikasi Maritim Muda Connect

  
 UNIVERSITAS  
 MULTIMEDIA  
 NUSANTARA

### 3.4.1 Metodologi Tugas untuk bagian *upgrade platform Java* Maritim Muda Connect



Gambar 3.3. *Flowchart* alur proses pengerjaan *upgrade platform Java*

Pada tahap pengembangan menggunakan *emulator Android*, aplikasi Maritim Muda Connect awalnya tidak dapat dijalankan karena ketidakcocokan versi *Gradle* dan *Java* yang digunakan. Pada kode sumber awal, *Gradle* yang digunakan adalah versi 7.6.3 dengan *plugin com.android.application* versi 7.3.0, serta menggunakan *Java 17* (LTS). Gambar 3.3 mengilustrasikan alur pengerjaan proses *upgrade platform Java*.

Untuk mengatasi kendala tersebut, dilakukan peningkatan versi *Gradle* dan *plugin Android* agar mendukung penggunaan *Java 21*, yang saat ini merupakan

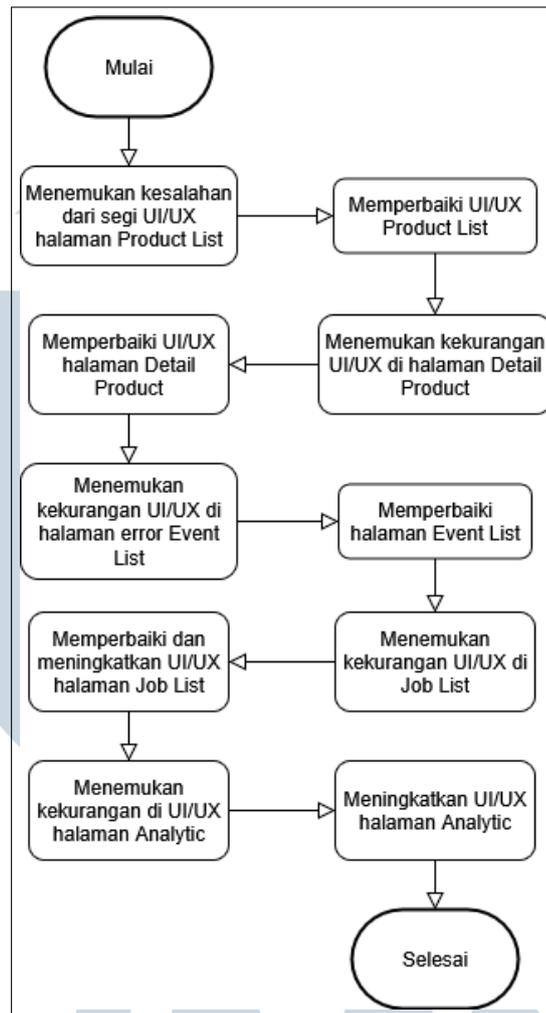
versi *Long Term Support* (LTS) terbaru. Keputusan ini dilakukan berdasarkan beberapa pertimbangan. *Java 21* menyediakan peningkatan keamanan dan perbaikan *bug* terbaru, lalu seluruh anggota tim pengembangan menggunakan *Java 21* sehingga diperlukan standarisasi lingkungan kerja, dan yang terakhir, *Java 21* dipilih untuk mempersiapkan kebutuhan skalabilitas dan penambahan fitur lanjutan di masa mendatang.

Setelah proses *upgrade* tersebut selesai, terdeteksi masalah baru: *package Flutter qr\_code\_scanner* menjadi tidak kompatibel dengan versi *Gradle/Java* yang diperbarui. *Package* ini digunakan untuk memindai kode QR pada kartu tanda anggota pengguna di Perkumpulan Maritim Muda Nusantara. Ketidakcocokan muncul karena *qr\_code\_scanner* belum sepenuhnya mendukung konfigurasi *Gradle 8.x* atau *Java 21*, sehingga proses kompilasi aplikasi gagal.

Untuk kebutuhan penggantian *qr\_code\_scanner* di dalam aplikasi, digunakan *package Flutter mobile\_scanner*. *Package* ini dipilih karena merupakan perbaruan dari *qr\_code\_scanner* yang lama dari pengembang yang sama, dengan perbedaan utama bahwa *package* ini aktif dipelihara dan diperbarui secara berkala. Proses konversi *package* dari *qr\_code\_scanner* ke *mobile\_scanner* meliputi penggantian *widget* utama dari *QRView* ke *MobileScanner*. Perbedaan utama dari pemakaian *package* ini adalah membuat proses memindai *QR Code* menjadi lebih otomatis tanpa harus membuat metode terpisah untuk memproses hasil pindai *QR Code*. Lalu perbedaan terakhir ada di keperluan untuk membuat kotak/area pemindaian, di mana untuk saat ini menggunakan *mobile\_scanner* tidak perlu mendefinisikan variabel terpisah untuk menentukan area atau ukuran dari pemindaian.

### **3.4.2 Metodologi Tugas untuk perbaruan UI/UX Maritim Muda Connect**

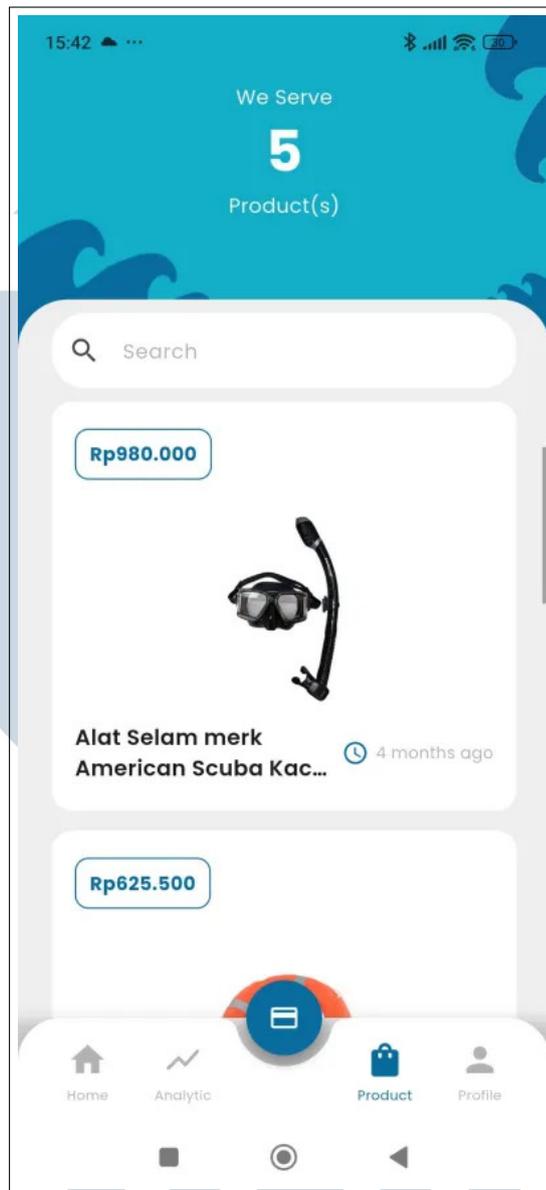
Dari sekian banyak halaman yang Maritim Muda Connect miliki (Gambar 3.2), beberapa halaman dari aplikasi ini memiliki kekurangan dari segi UI/UX yang ditampilkan. Gambar 3.4 menggambarkan alur dari proses pembaruan UI/UX yang dilakukan di beberapa halaman Maritim Muda Connect.



Gambar 3.4. *Flowchart* alur pengerjaan perbaruan UI/UX Maritim Muda Connect

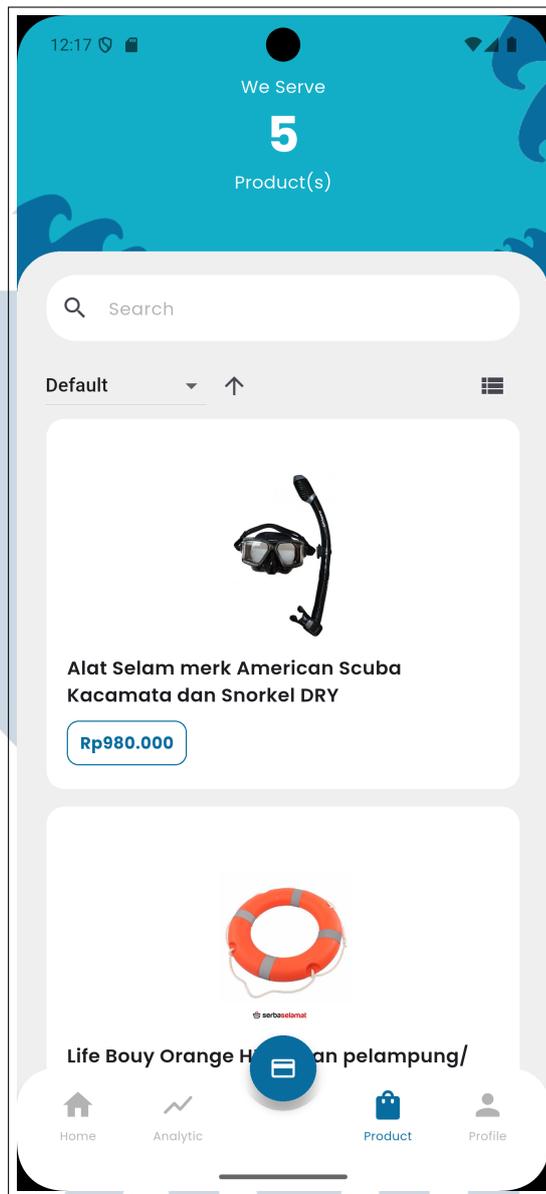
## A Halaman Product

Halaman Product dari Aplikasi Maritim Muda Connect adalah halaman yang berfungsi untuk memberikan daftar produk atau *souvenir* yang disediakan oleh Perkumpulan Maritim Muda Nusantara. Halaman produk ini sudah ada sebelumnya, dan pada keadaan awalnya, halaman ini memiliki *search bar* untuk mencari produk, dan susunan desain yang mendekati tipe *card*. Namun, sesuai dengan Gambar 3.5, desain dari halaman produk ini memiliki informasi yang terlalu berlebihan serta juga tata susun yang tidak konvensional dengan aplikasi lainnya.



Gambar 3.5. Tampilan awal halaman Product

Beberapa hal yang bisa dicatat dari Gambar 3.5 adalah sebagai berikut. Pertama, informasi *timestamp* produk menutupi nama produk, sehingga dinilai berlebihan karena tidak membantu pengguna dan malah mengobstruksikan nama lengkap dari produk yang diperlihatkan. Kedua, penempatan teks harga juga dinilai aneh dan tidak sepadan dengan tipe susunan yang umum seperti di aplikasi *e-commerce* lainnya. Perubahan dan nilai kurang ini telah didiskusikan dengan supervisor Bapak Rafli untuk memperbolehkan perubahan di bagian-bagian ini.

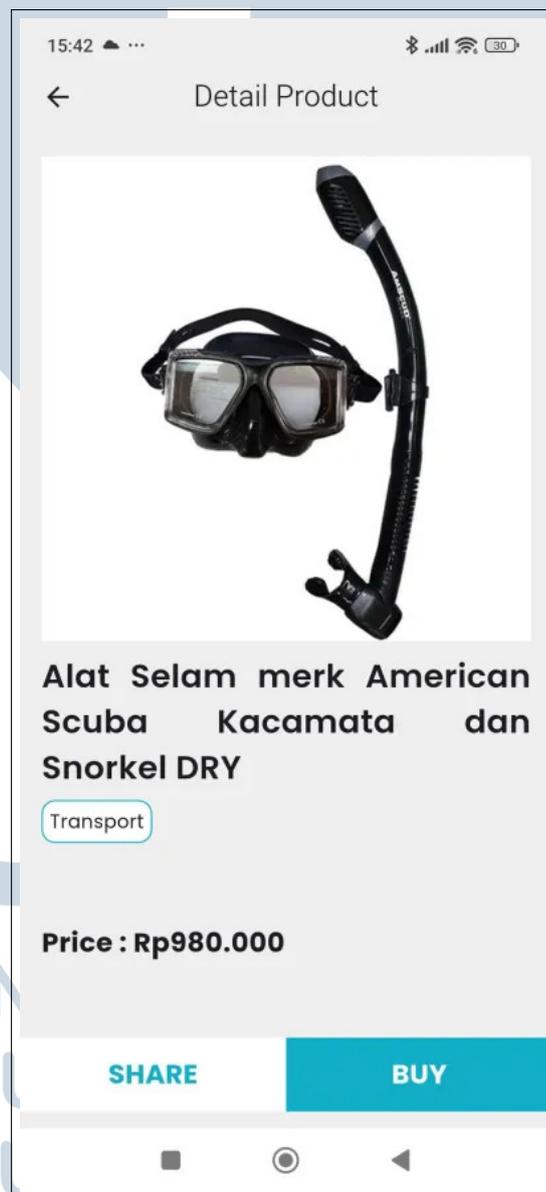


Gambar 3.6. Hasil Perubahan tampilan halaman *Product*

Setelah perubahan yang bisa dilihat di Gambar 3.6, bisa terlihat tata susun yang lebih familiar dengan standar desain untuk menjual produk. Informasi waktu dihilangkan untuk bisa menampilkan nama lengkap produk dengan lebih baik, serta penempatan teks harga juga sudah sesuai dengan standar desain untuk produk yang dijual dalam aplikasi seperti menurut [8] di mana urutan elemen mulai dari gambar, judul atau nama produk, lalu harga memperkuat daya tingkat keterbacaan dan kenyamanan pengguna. Lalu, di halaman ini ditambahkan opsi untuk mensortir serta penggantian tipe tampilan yang ada antara *Card View* dan *List View*. Penambahan ini mendukung skalabilitas jika produk yang dijual di

Maritim Muda Connect menjadi lebih banyak, dan mengikuti salah satu prinsip 8 *golden rules* dari *UI/UX Design* yaitu *Support Internal Locust of Control*, di mana pengguna diberikan kendali atas cara mereka melihat dan mengelola konten yang ada [9].

## B Halaman Detail Product



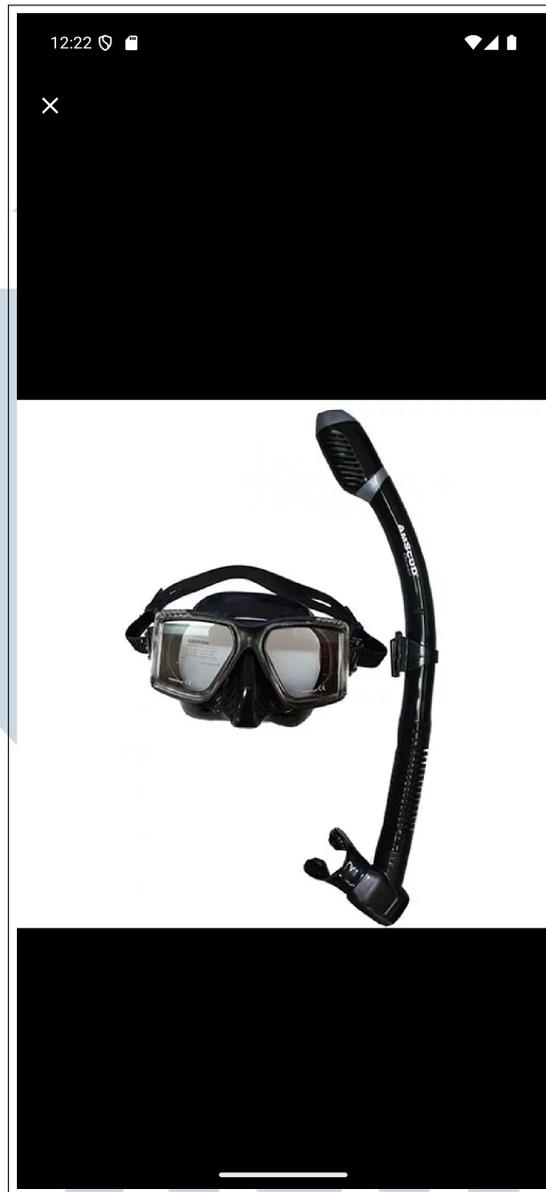
Gambar 3.7. Tampilan awal halaman *Detail Product*

Pada Gambar 3.7, terlihat banyak hal yang kurang tepat dari tampilan halaman. Pertama, hal terbesar yang menjadi masalah adalah ketidakadanya elevasi

atau pemisahan hierarki dari gambar produk dan konten teks. Semua elemen teks kecuali tombol *Share* dan *Buy* ditempatkan tanpa *background* atau *padding* untuk memisahkan elemen-elemen ini. Kesalahan ini menyebabkan penampilan datar dan sulit untuk dicerna oleh pengguna dengan baik. Kedua, gambar produk yang disediakan di halaman ini juga belum bisa diperbesar untuk bisa dilihat lebih baik.



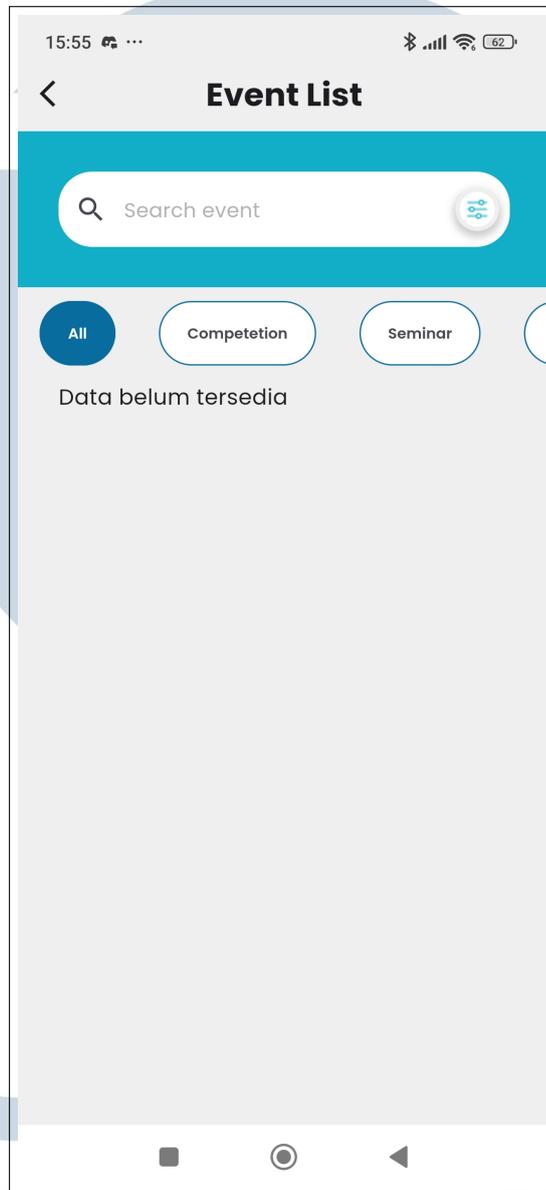
Gambar 3.8. Tampilan halaman *Detail Product* setelah perubahan



Gambar 3.9. Tampilan foto produk dari halaman *Detail Product*

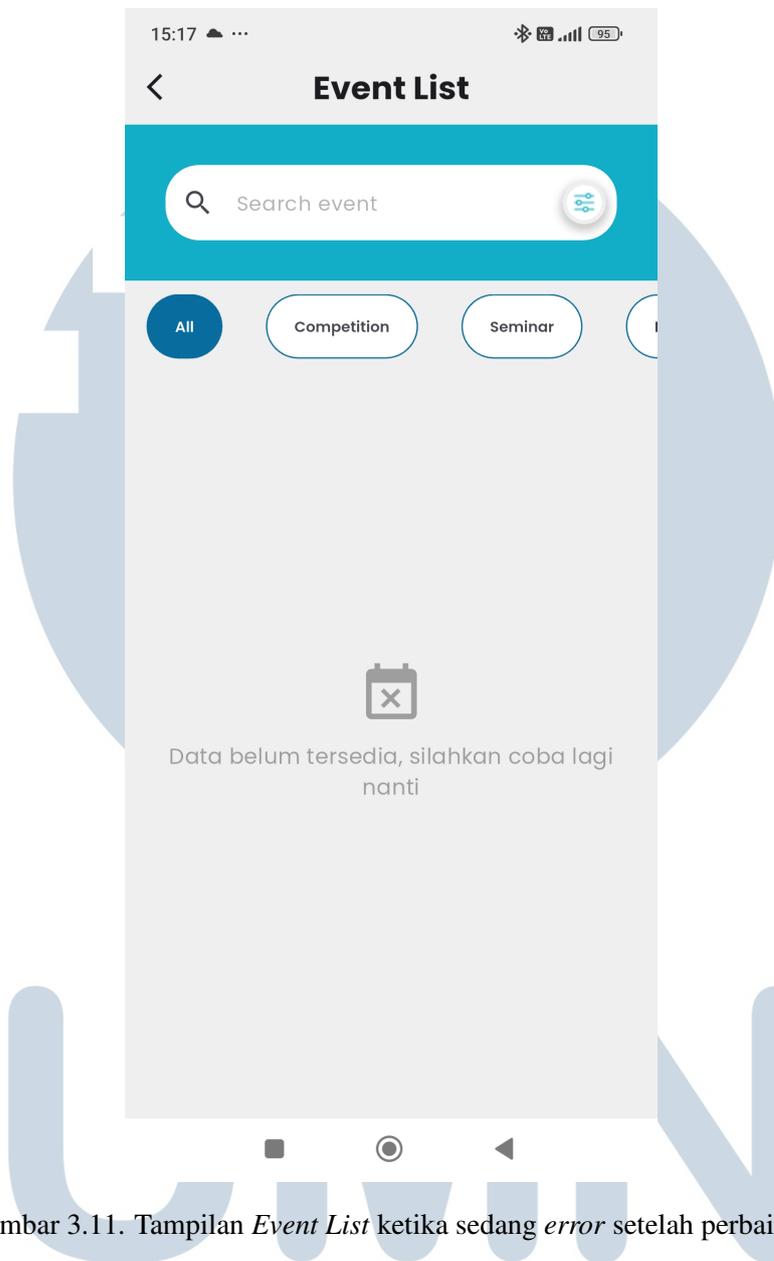
Gambar 3.8 menampilkan tampilan halaman *Detail Product* yang sudah dirombak. Tampilan ini dinilai memiliki kesan yang lebih *modern*, mengurangi teks berlebihan, dan juga *styling* yang lebih sejalan dengan tema aplikasi. Fitur untuk bisa memencet gambar produk untuk bisa diperbesar juga sudah ditambahkan, seperti di Gambar 3.9.

### C Halaman *Event List*



Gambar 3.10. Tampilan *Event List* ketika sedang mengalami *error*

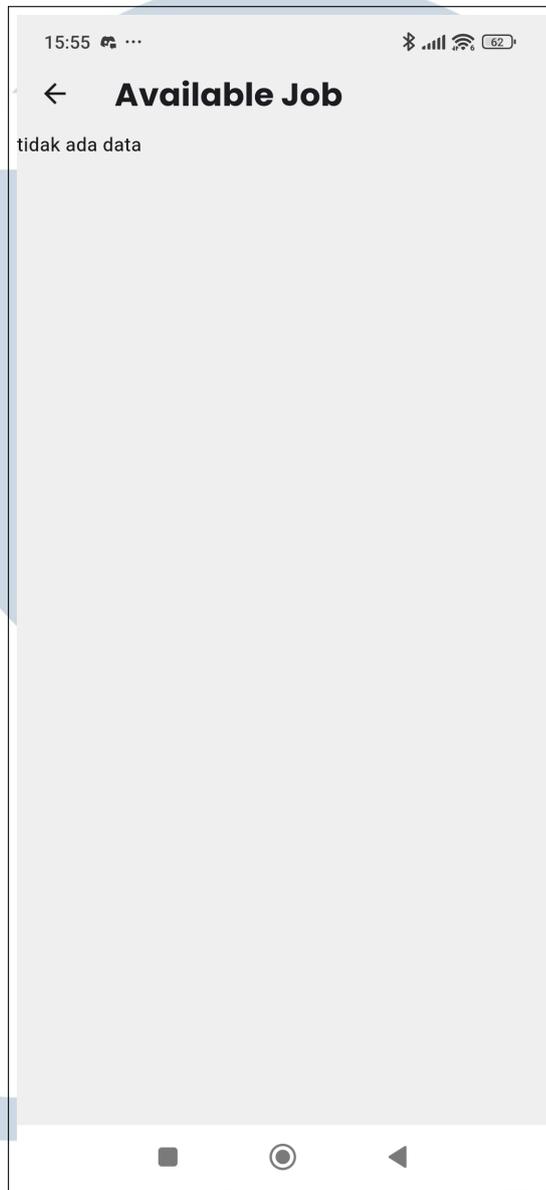
Gambar 3.10 menampilkan tampilan *error* yang diperlihatkan oleh halaman *Event List*. Dalam halaman ini, diperbaiki di bagian salah ketik dari kata "*Competition*" menjadi "*Competition*", dan modifikasi di kalimat penjelasan *error* yang diberikan oleh halaman. Fungsi *refresh* juga telah diberikan di halaman. Gambar 3.11 menunjukkan tampilan yang sudah diperbaiki.



Gambar 3.11. Tampilan *Event List* ketika sedang *error* setelah perbaikan

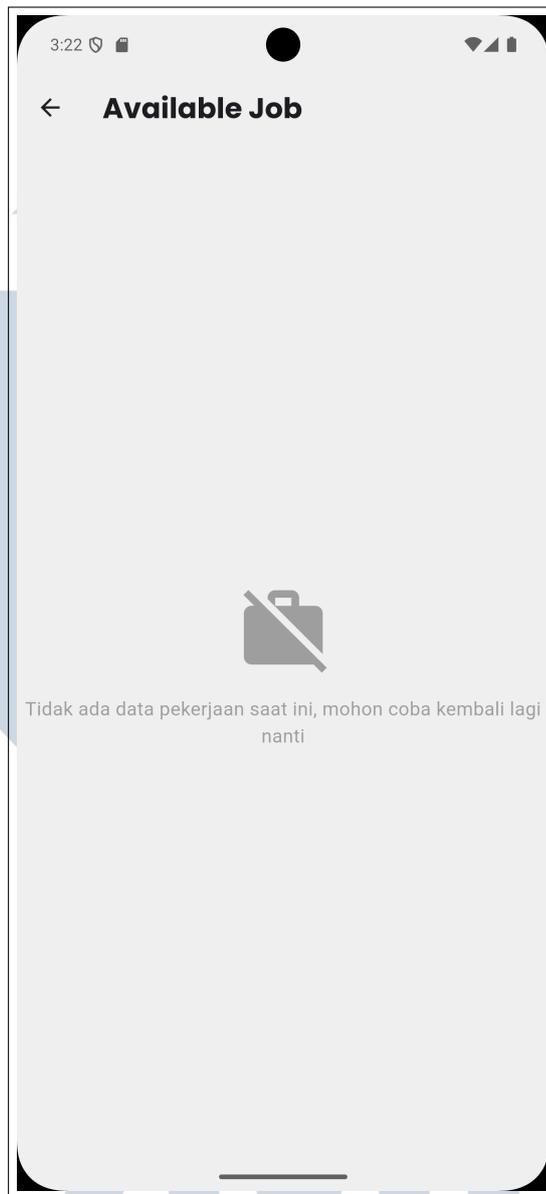
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

## D Halaman Job List



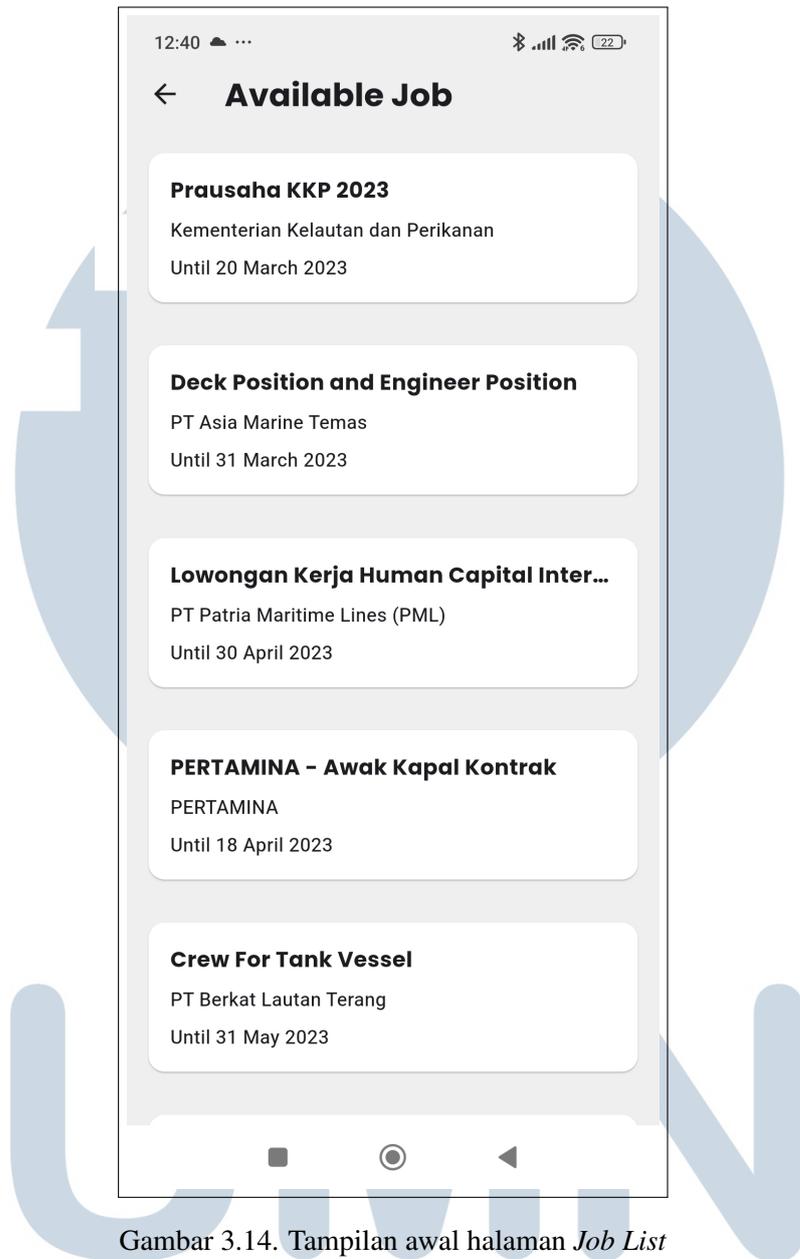
Gambar 3.12. Tampilan halaman *Job List* saat sedang *error*

Tampilan *error* yang diperlihatkan halaman *Job List* terlihat di Gambar 3.12. Halaman ini diberikan perubahan dalam pesan *error* yang diberikan. Gambar 3.13 menunjukkan tampilan yang sudah diperbaiki. Tampilan *error* yang baru ini lebih sesuai dengan standar desain halaman *error* dari aplikasi lain pada umumnya, serta memberikan informasi yang sedikit lebih lengkap dibandingkan dengan tampilan sebelumnya.



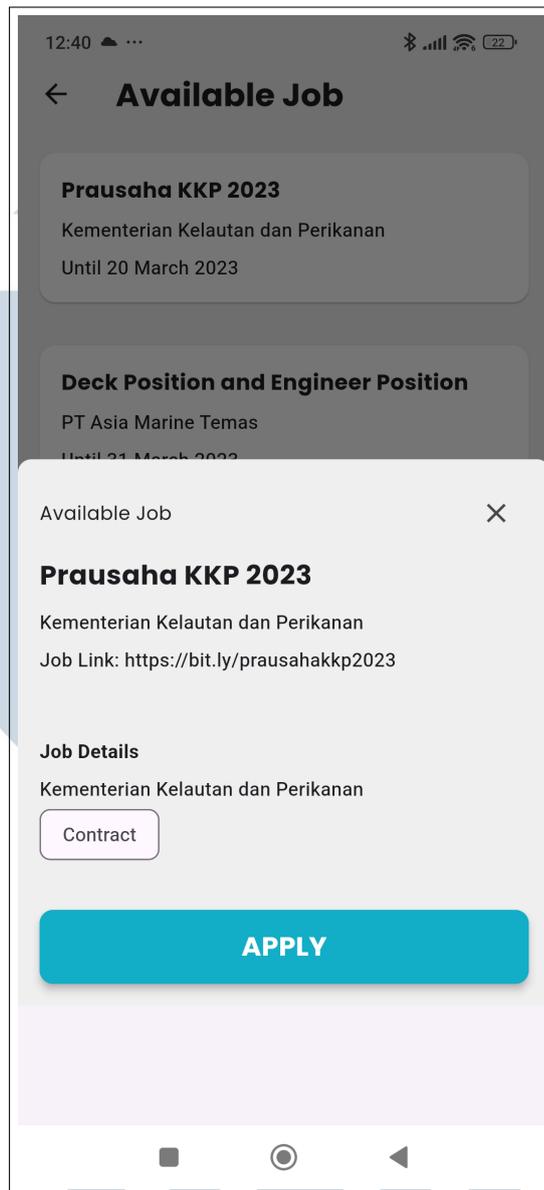
Gambar 3.13. Tampilan *Job List* ketika sedang *error* setelah perbaikan

Halaman *Job List* dari aplikasi Maritim Muda Connect adalah halaman yang memberikan informasi lowongan kerja terkait sektor kemaritiman. Halaman ini memberikan informasi sekilas tentang lowongan pekerjaan yang sedang dibuka, dan menyediakan tautan menuju informasi yang lebih lanjut tentang lowongan pekerjaan yang dipilih. Tampilan dari halaman ini bisa dilihat di Gambar 3.14.



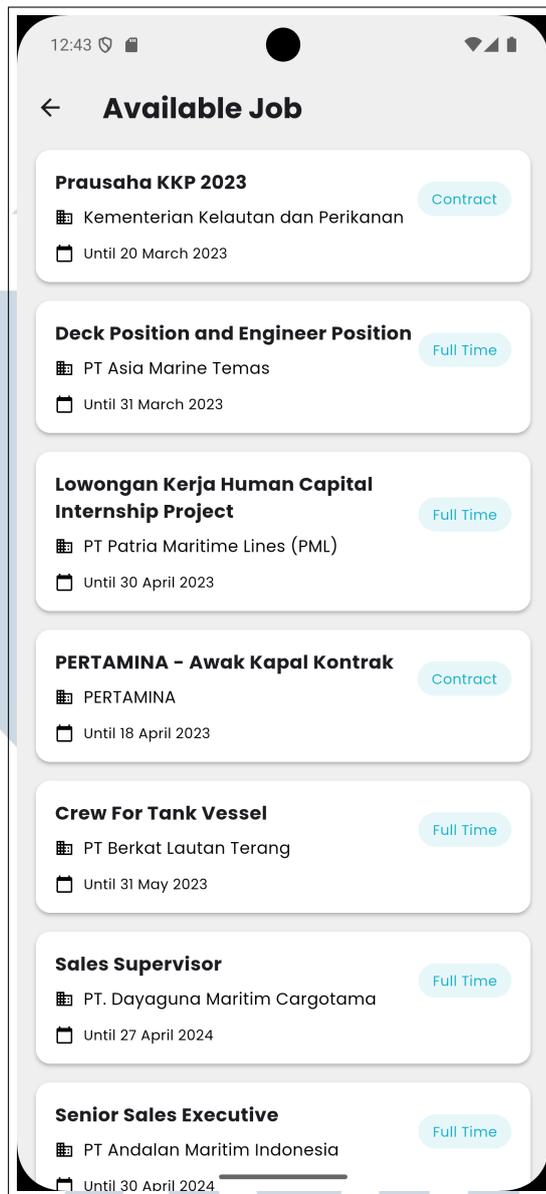
Gambar 3.14. Tampilan awal halaman *Job List*

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



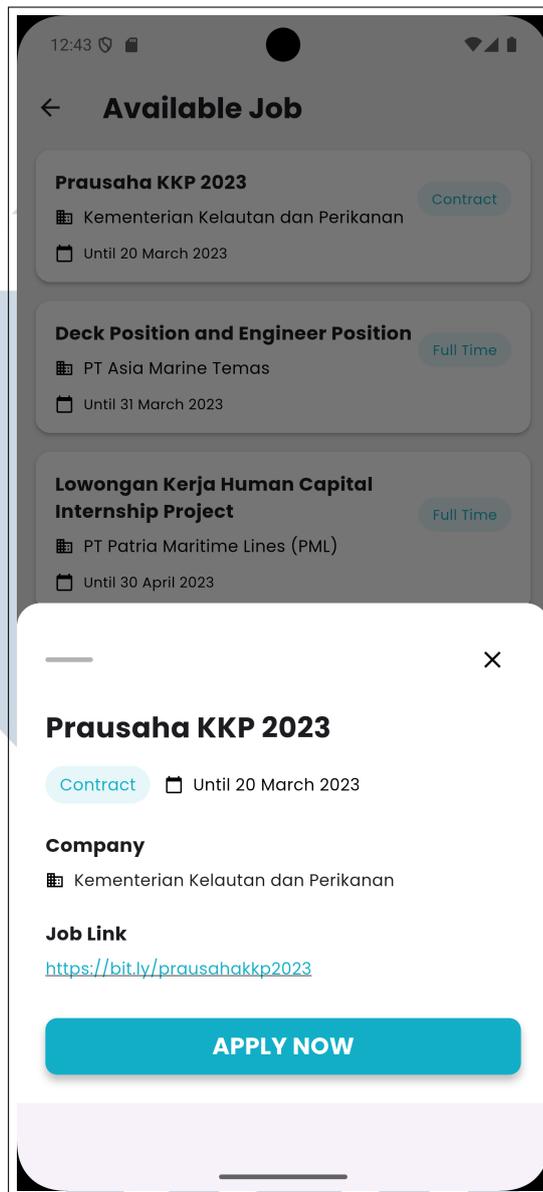
Gambar 3.15. Tampilan awal ketika memilih pekerjaan

Gambar 3.14 menampilkan tampilan halaman *Job List* sebelum perubahan. Tampilan ini memiliki beberapa kekurangan, seperti kurangnya kontras dalam tampilan untuk meningkatkan kemudahan membaca, posisi atau nama pekerjaan yang melebihi batas dan menjadi tersembunyi, serta jarak antara elemen job yang terlihat agak jauh. Gambar 3.17 menampilkan *bottom sheet menu* yang muncul ketika salah satu elemen pekerjaan dipilih. Tampilan *bottom sheet menu* ini terlihat buruk, di mana tidak ada pemisahan atau elevasi antar teks, serta juga tata susun informasi yang tidak teratur.



Gambar 3.16. Tampilan halaman *Job List* setelah perubahan

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.17. Perubahan tampilan ketika memilih pekerjaan

Setelah perubahan yang bisa terlihat di Gambar 3.16, ditambahkan elemen seperti ikon dan kategori pekerjaan yang memberikan informasi dengan lebih efisien dan mudah dicerna oleh pengguna. Nama atau posisi pekerjaan yang ditampilkan juga sudah tidak tersembunyi. Tampilan *bottom sheet menu* juga sudah diperbaiki di Gambar 3.17, memperbaiki tata susun informasi teks, memberikan ikon untuk memperjelas informasi, dan memberikan indikator *bottom sheet* agar pengguna tidak terbingung dengan tampilan yang ada.

## E Halaman *Analytic*

Halaman *Analytic* adalah tampilan di aplikasi Maritim Muda Connect yang menampilkan statistik data dari pengguna Maritim Muda Connect. Tampilan awal ini di Gambar 3.18 memperlihatkan angka statistik yang statis dan tidak bisa dimodifikasi untuk mempermudah pengguna mencerna data. Tampilan halaman ini diberikan modifikasi seperti di Gambar 3.19, menambahkan opsi untuk memodifikasi tampilan sesuai keinginan pengguna, dan merapikan tata susun serta ukuran grafik yang ditampilkan.

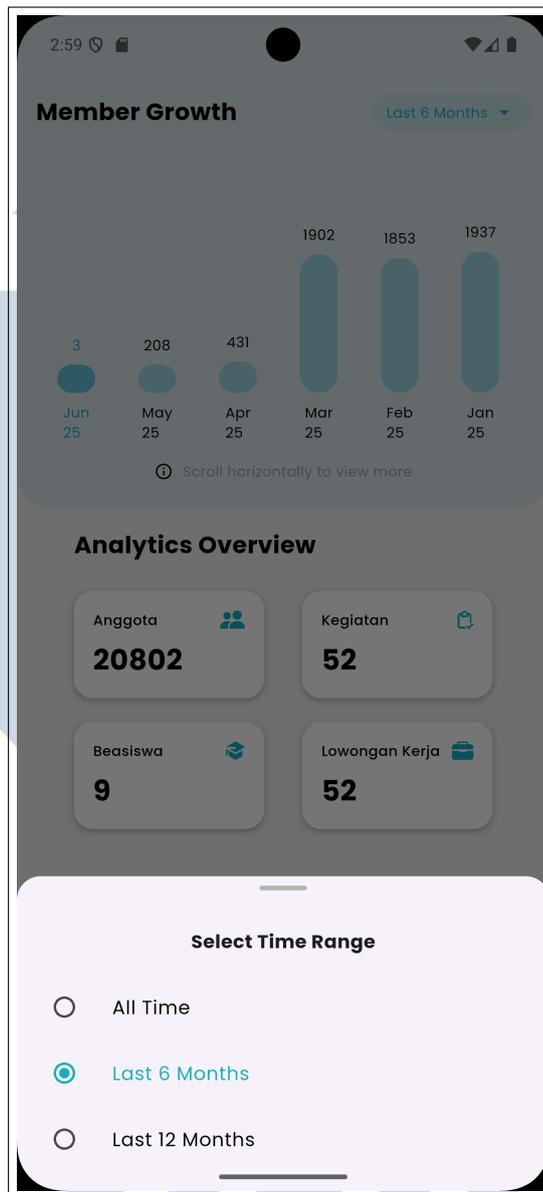


Gambar 3.18. Tampilan awal halaman *Analytic*



Gambar 3.19. Tampilan halaman *Analytic* setelah perbaikan

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

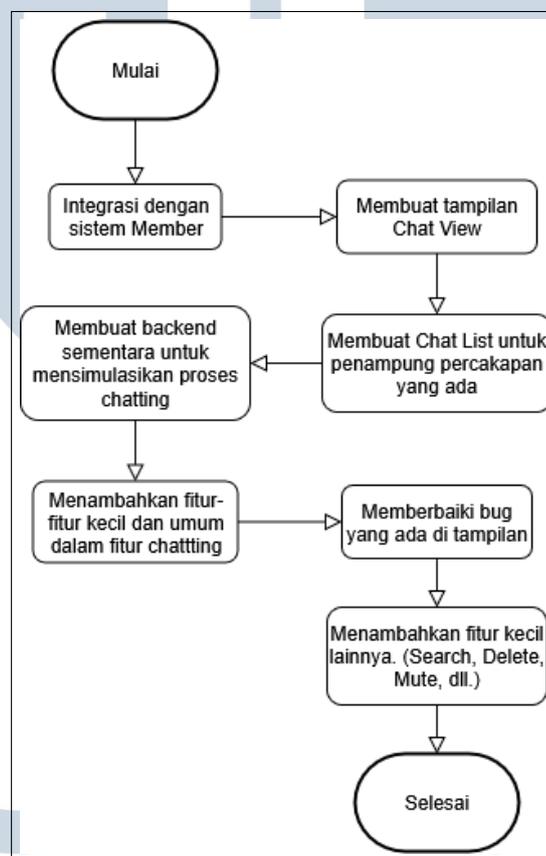


Gambar 3.20. Tampilan opsi mengubah jarak waktu *Analytics*

### 3.4.3 Penambahan Fitur *Chat* antar anggota

Fitur Chatting adalah fitur yang memperbolehkan pengguna aplikasi untuk bisa mengirimkan pesan ke pengguna lainnya. Fitur ini diminta dibuat oleh Pak Kaisar selaku Ketua Umum Perkumpulan Maritim Muda Nusantara, di mana fitur ini memperkuat kegunaan aplikasi Maritim Muda Connect sebagai pusat pertukaran informasi antar anggota Perkumpulan Maritim Muda Nusantara. Untuk pembuatan fitur ini, tidak ditentukan secara penuh kebutuhan fitur yang diinginkan, namun dalam diskusi bersama Pak Rafli menentukan beberapa basis untuk pembuatan fitur

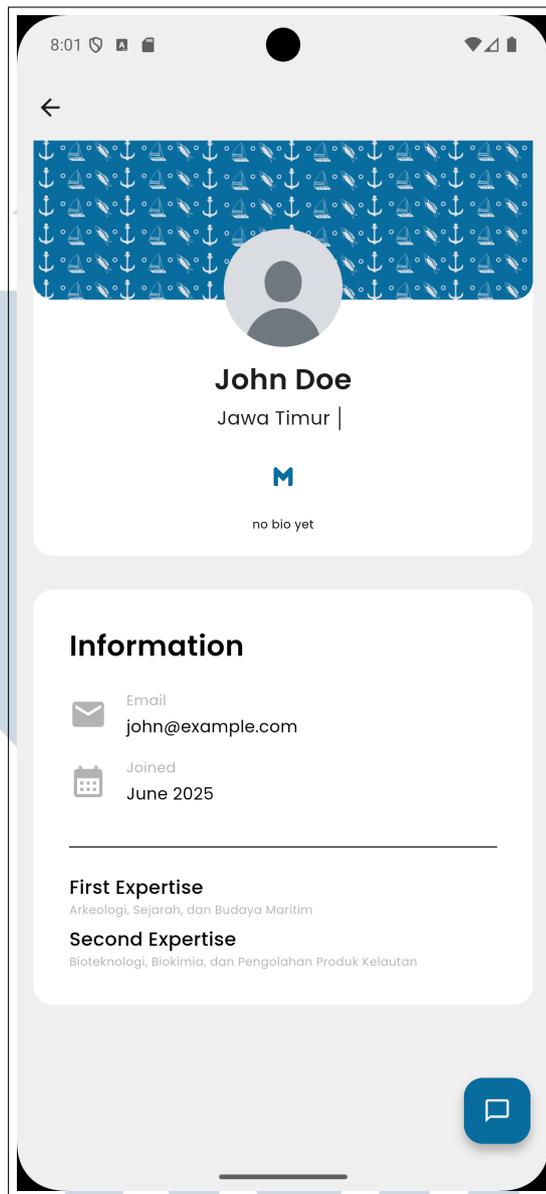
ini yaitu: Memiliki fungsionalitas dasar dari fitur *chatting* aplikasi lainnya, yaitu kontak antar 2 pengguna, adanya pengiriman media seperti foto, video, dan *file* lainnya, serta pencarian pesan, notifikasi, dan penghapusan pesan. Perlu dicatat bahwa pekerjaan fitur ini belum bisa dipublikasikan dan tidak selesai sepenuhnya, karena pekerjaan dari sisi *backend* atau *server* yang tidak selesai. Pak Raffi menginstruksikan untuk mengerjakan sebanyak dan sebisa mungkin untuk fitur ini, sehingga sebagian besar dari fitur yang diperlihatkan akan menggunakan data palsu (*Dummy Data*). Gambar 3.21 mengilustrasikan alur pengerjaan fitur *Chat*.



Gambar 3.21. Flowchart alur pengerjaan fitur *Chat*

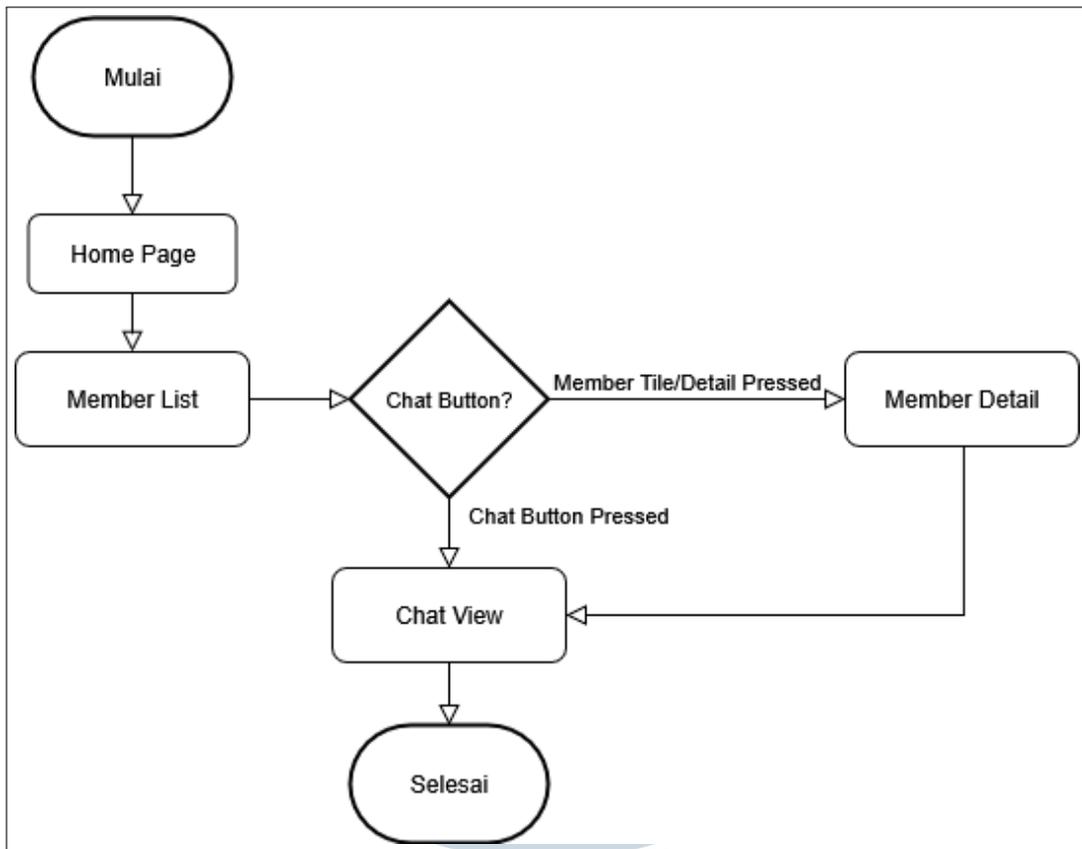
#### A Integrasi dengan Member List

Untuk bisa menggunakan fitur *Chat* ini, digunakan fitur sebelumnya yang sudah ada yaitu *Member List*. Fitur *Member* ini berfungsi untuk bisa mencari sesama anggota Perkumpulan Maritim Muda Nusantara yang sudah terdaftar. Pengguna dapat mencari nama anggota spesifik yang mereka inginkan untuk bisa menghubungi mereka melalui e-mail atau media sosial yang tercantum di profil mereka seperti di Gambar 3.22.



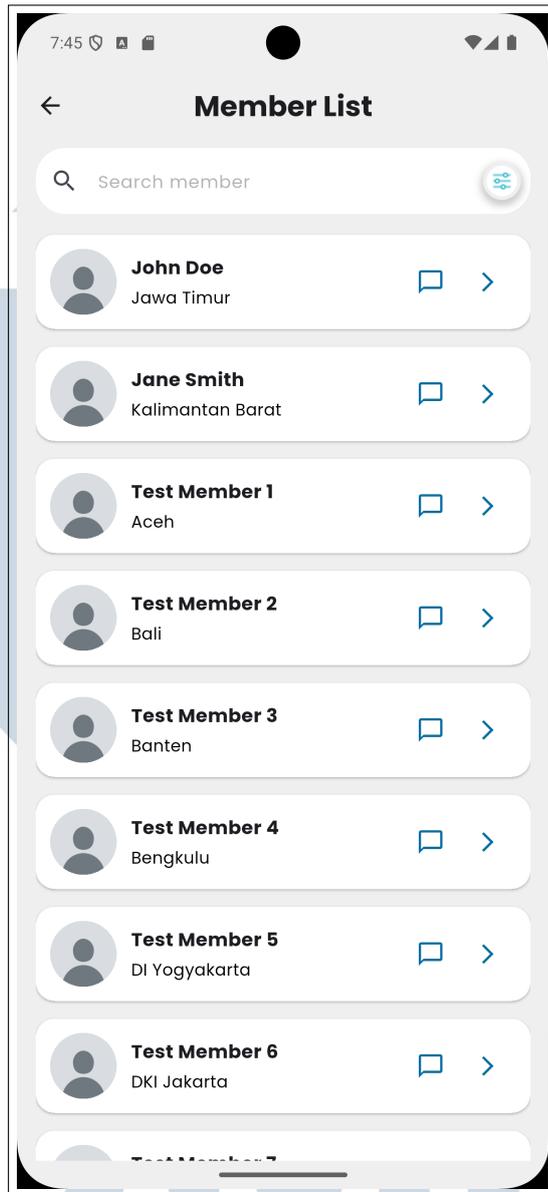
Gambar 3.22. Contoh Tampilan halaman *Member Detail*

Untuk dapat mengakses fitur *Chat* ini, diperlukan tujuan atau penerima. Penerima ini ditentukan melalui pemilihan member dari *Member List* atau *Member Detail*, yang lalu akan menggunakan identifikasi ID untuk membuat halaman *Chat View* dengan penerima. Gambar 3.23 menggambarkan alur dari integrasi sistem *Member*, dengan adanya tombol ikon *Chat* yang bisa langsung membawa user ke *Chat View* atau bisa memencet elemen *Tile* dari suatu *Member* untuk dibawa ke *Member Detail* terlebih dahulu, baru memencet tombol *Chat* lagi untuk langsung dibawa ke *Chat View*. Gambar 3.24 memberikan contoh tampilan *Member List* dan tampilan dari elemen *Tile Member* yang ada.



Gambar 3.23. *Flowchart* integrasi sistem *Member* dengan *Chat*

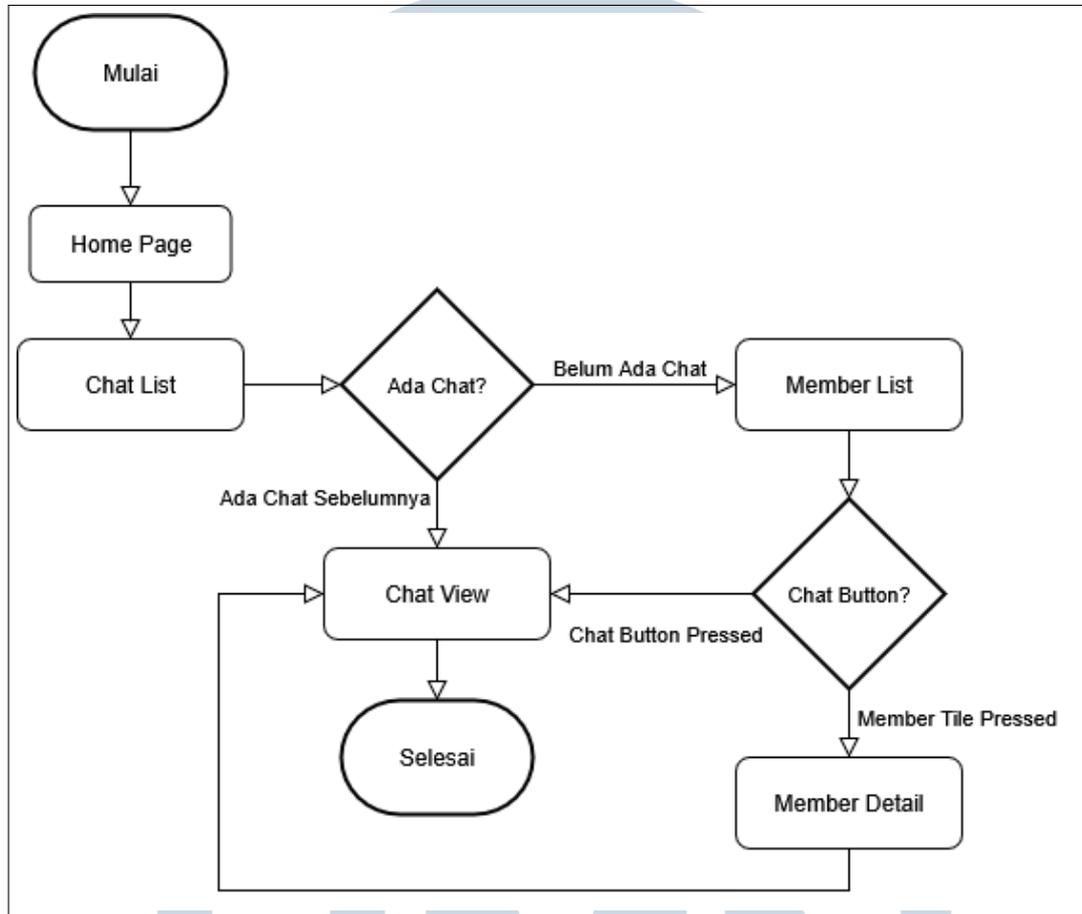




Gambar 3.24. Contoh Tampilan halaman *Member List*

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

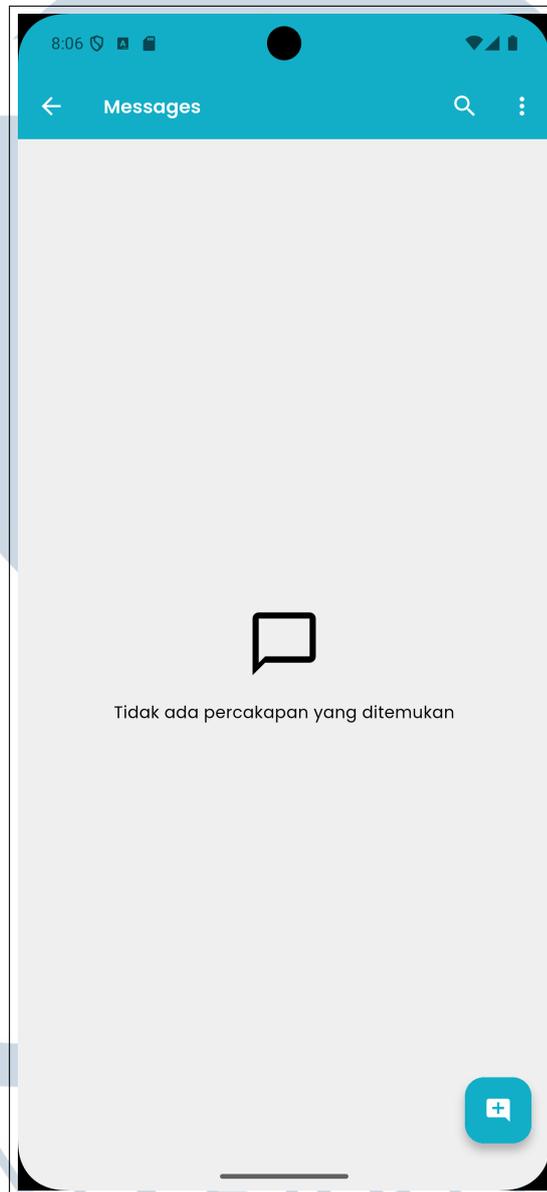
## B Chat List



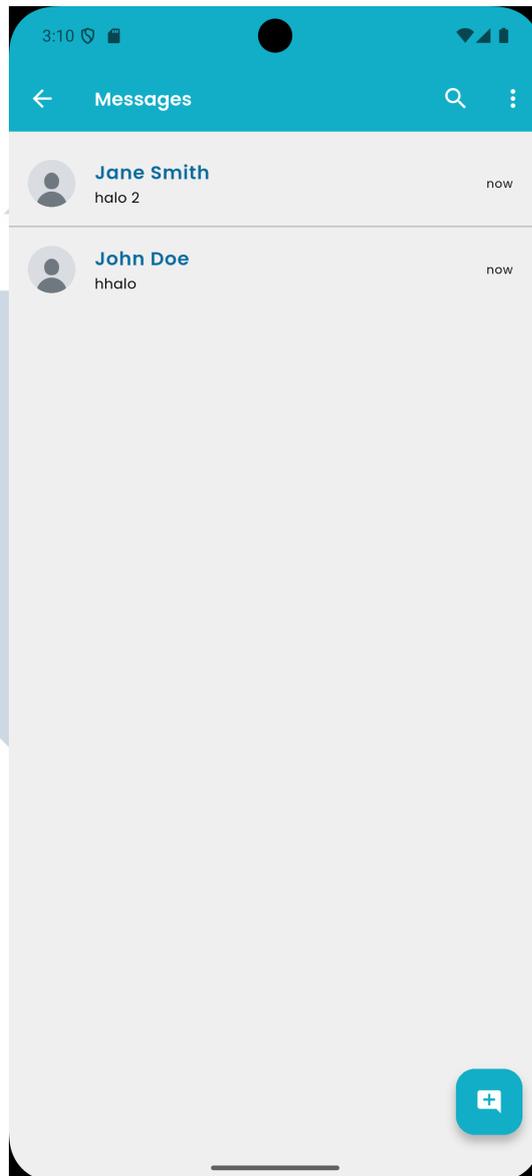
Gambar 3.25. Flowchart fungsi Chat List

*Chat List* adalah menu baru yang dibuat untuk menampilkan daftar percakapan yang pengguna telah kirim atau terima. Menu ini berfungsi seperti menu "*Chats*" yang ada di aplikasi *chatting* lainnya. Penjelasan alur dari fungsi ini dapat dilihat di Gambar 3.25, dengan contoh tampilan di Gambar 3.26 ketika dalam keadaan kosong dan di Gambar 3.27. Menu ini memperbolehkan pengguna untuk mengakses kembali percakapan yang telah terjadi, membuat percakapan baru, dan juga opsi fitur umum lainnya seperti *Profile*, *Delete Chat*, *Delete All Chat*, *Search*, *Mute*, dan *Mark all as read*. Terdapat tombol di sebelah kanan bawah untuk membawa pengguna langsung ke *Member List* untuk mencari pengguna untuk bercakap dengan. Pengguna juga bisa langsung melihat profil anggota lainnya dengan menekan lama pada suatu percakapan yang ada. *Delete Chat* di menu ini berfungsi untuk menyembunyikan percakapan yang sudah ada, dan akan muncul kembali jika ada pesan baru. Opsi *Search* di menu ini hanya bisa mencari sesuai

nama dari pihak yang terlibat dalam percakapan. Opsi *Mute* berfungsi untuk mematikan notifikasi dari percakapan tertentu. Dan yang terakhir, *Mark all as read* akan membuat semua pesan yang belum dibaca dianggap telah dibaca.



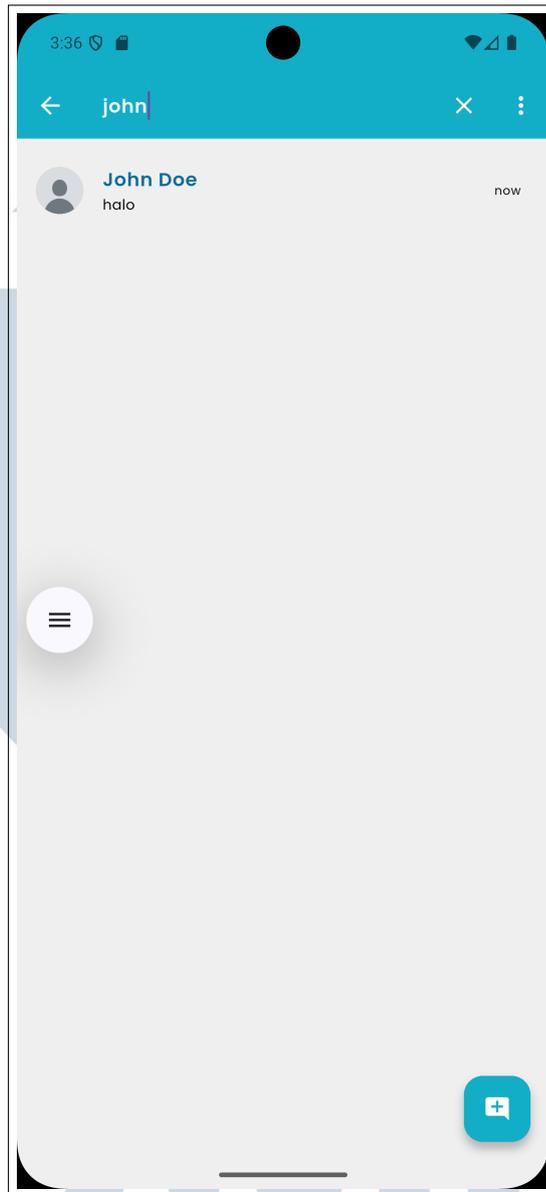
Gambar 3.26. Tampilan halaman *Chat List* ketika masih kosong



Gambar 3.27. Tampilan halaman *Chat List* ketika sudah ada percakapan

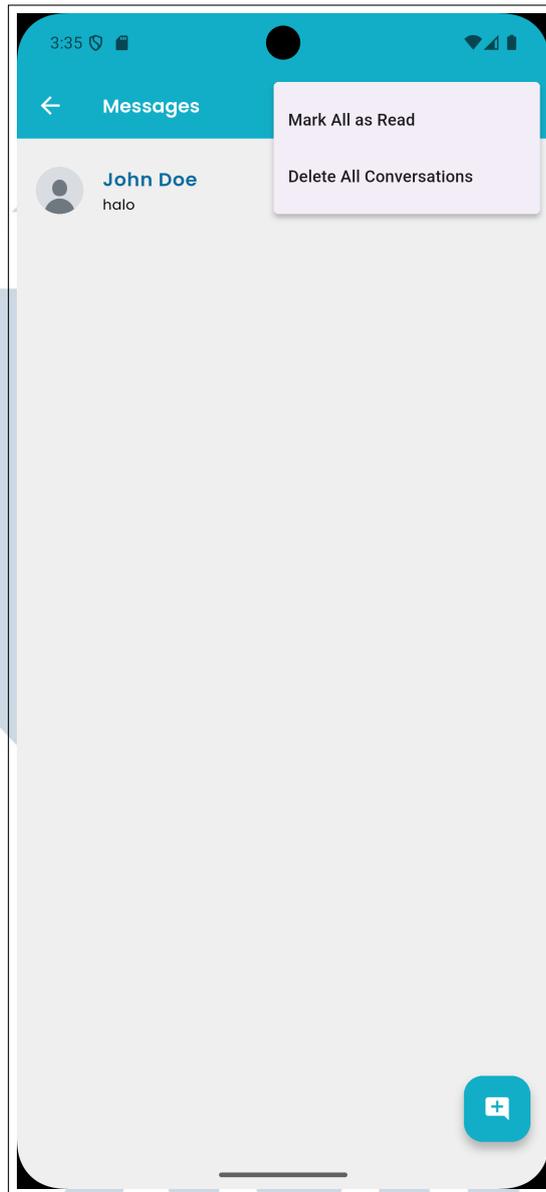
Fitur *Search* di dalam *Chat List* digunakan untuk mencari percakapan yang sudah pernah terjadi antar pengguna. Contoh tampilan penggunaan fitur ini terlihat di Gambar 3.28.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



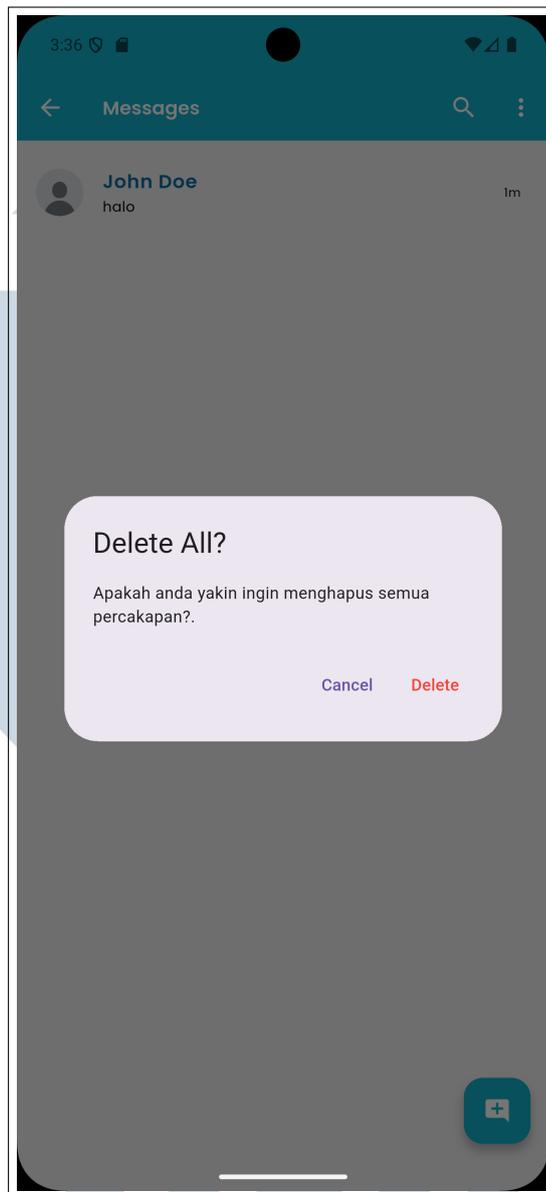
Gambar 3.28. Tampilan fitur *Search* di dalam *Chat List*

Terdapat *overflow* menu yang menampilkan aksi tambahan yang pengguna bisa lakukan. Di dalam *overflow* menu ini ada 2 aksi, yaitu melakukan *Mark all as read*, yaitu membuat semua indikator belum terbaca dalam *Chat List* hilang, dan juga *Delete all conversations*, yang akan menyembunyikan semua percakapan yang ada. Aksi ini melakukan penyembunyian, karena ini adalah perilaku yang sudah umum dalam aplikasi atau sistem *chatting* lainnya. Gambar 3.29 menampilkan tampilan *overflow* menu dalam *Chat List*.



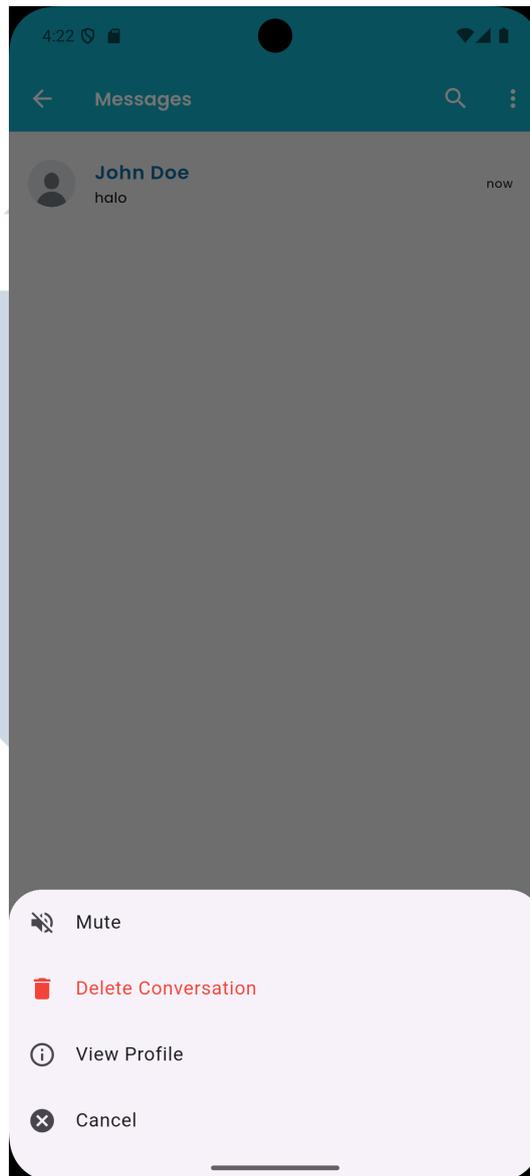
Gambar 3.29. Tampilan *overflow* menu dalam *Chat List*

Ketika memilih opsi untuk *Delete all conversations*, pengguna akan diberikan dialog peringatan. Dialog peringatan ini (Gambar 3.30) memperingati pengguna ketika melakukan aksi yang tidak bisa diulang. Adanya menu ini mengikuti prinsip dari *8 Golden Rules* tentang *Offer Informative Feedback* di mana pengguna harus diberi wawasan atas aksi yang akan dilakukannya [10].



Gambar 3.30. Tampilan dialog peringatan dalam *Chat List*

Dalam setiap objek percakapan yang ada di *Chat List*, terdapat *long press menu* untuk pengguna (Gambar 3.31). Dalam menu ini terdapat beberapa aksi yang pengguna bisa lakukan, yaitu untuk *Mute* atau mematikan notifikasi dari pengguna individual, *Delete conversation* untuk menyembunyikan percakapan individual. Perilaku penyembunyian ini sudah umum, dan masih bisa diganti pada seterusnya. Opsi terakhir dari *long press menu* ini adalah *View Profile*, di mana opsi ini akan membawa pengguna ke *Member Detail*.



Gambar 3.31. Tampilan *long press menu* di setiap objek percakapan

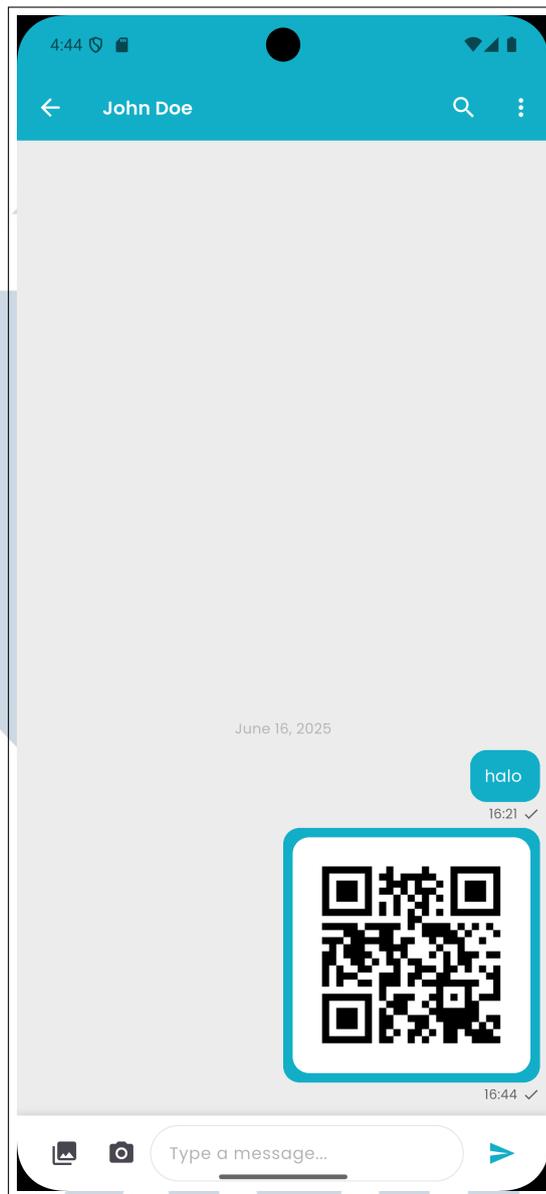
## C Chat View

*Chat View* adalah halaman yang menampilkan percakapan pengguna dengan pengguna lainnya. Halaman ini dapat diakses dengan memilih pengguna lain melalui *Member List*, *Member Detail*, dan dari *Chat List*. Halaman *Chat View* ini memiliki beberapa fungsionalitas umum dari sistem *Chat* yang ada di aplikasi lain, seperti Mengirim Pesan, Menerima Pesan, reaksi *Emoji*, *Reply*, *Copy*, *Forward*, *Share*, *Delete*, *Delete all*, dan *Search*. Operasi *backend* yang terjadi di fitur ini masih belum terintegrasi sepenuhnya, sehingga beberapa operasi fitur yang ada

akan berbeda di kelanjutannya. Terdapat juga akses untuk bisa melihat profil dari pengguna dalam percakapan. Gambar 3.32 menampilkan halaman *Chat View* ketika masih kosong atau belum ada pesan yang dikirim, dan Gambar 3.33 ketika sudah mengirimkan pesan dengan contoh mengirim file media juga.



Gambar 3.32. Tampilan *Chat View* ketika baru memasuki percakapan

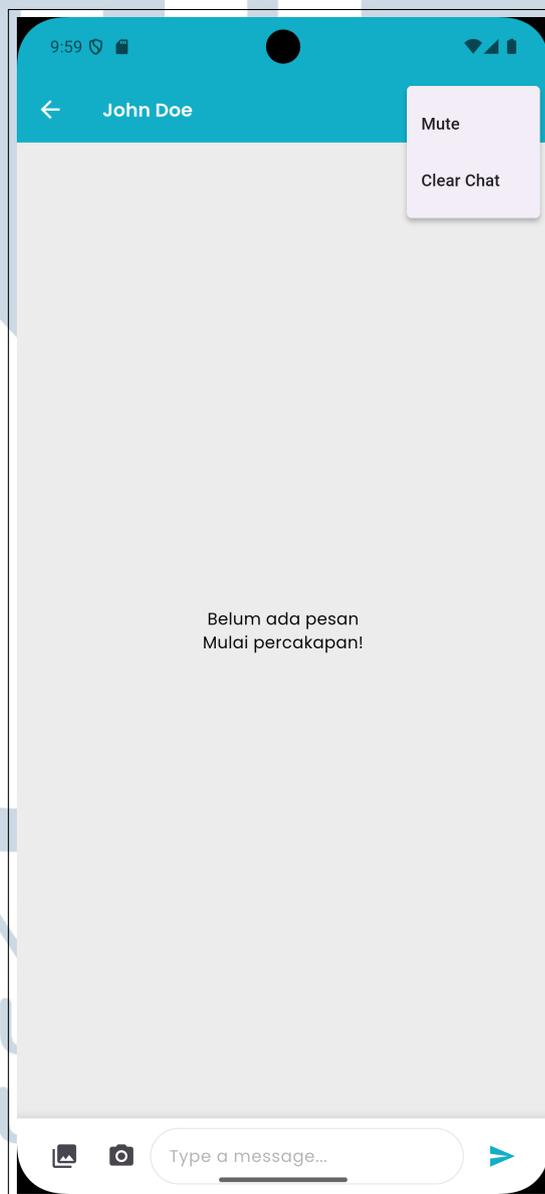


Gambar 3.33. Tampilan *Chat View* dengan isi

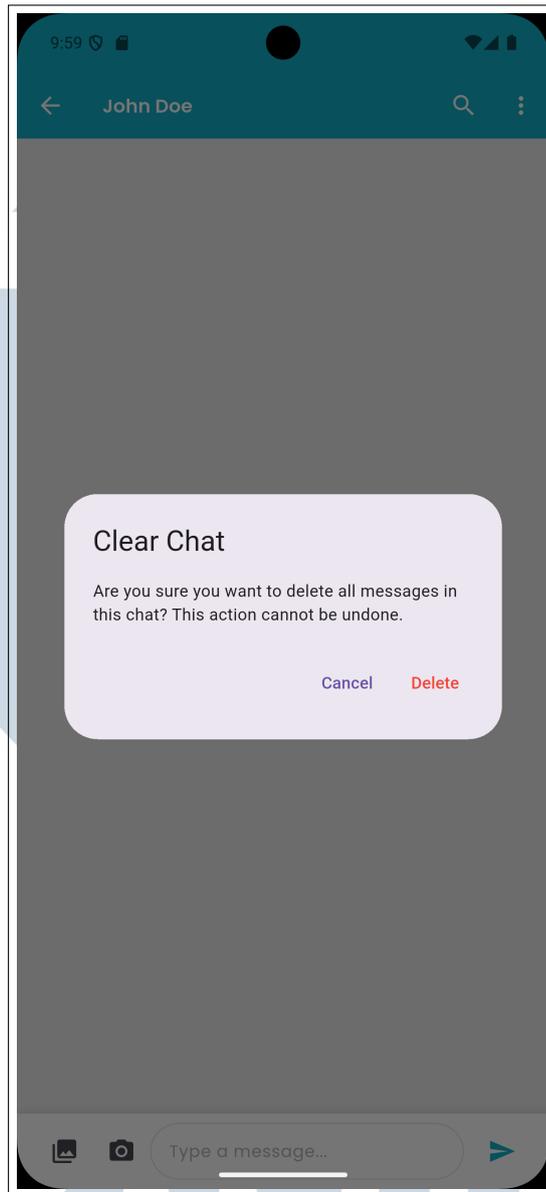
Pengguna dapat langsung mengakses kamera dengan tombol kamera di sebelah area ketik pesan, atau dengan memilih media lewat tombol sebelah kamera. Pesan media akan langsung dikirim ketika sudah dipilih. Media yang dipilih juga bisa ditekan untuk diperbesar atau dilihat dengan lebih baik, seperti di 3.9. Setiap pesan yang terkirim memiliki catatan waktu kapan dikirim, serta juga ikon *checkbox* untuk menandakan apabila pesan sudah terbaca. *Separator* berbasis tanggal pengiriman pesan juga ada.

Di *App Bar* yang ada di halaman ini, terdapat beberapa tombol yang pengguna bisa tekan untuk melakukan aksi tertentu. Pengguna dapat tekan nama

pengguna dalam percakapan untuk langsung ke profil mereka. Terdapat tombol *Search* untuk memfilter pesan sesuai dengan yang dicari. Lalu yang terakhir ada menu *overflow* yang berisi opsi untuk *Mute* dan untuk *Clear Chat*. *Mute* untuk mematikan notifikasi di percakapan, dan *Clear Chat* untuk menghapus semua pesan yang telah dikirim di percakapan. Aksi *Clear Chat* ini memiliki dialog peringatan sebelum melakukan aksi karena merupakan aksi yang tidak bisa diulang. Gambar 3.34 memperlihatkan tampilan *overflow* menu, dan Gambar 3.35 memperlihatkan tampilan peringatan ketika ingin melakukan *Clear Chat*.



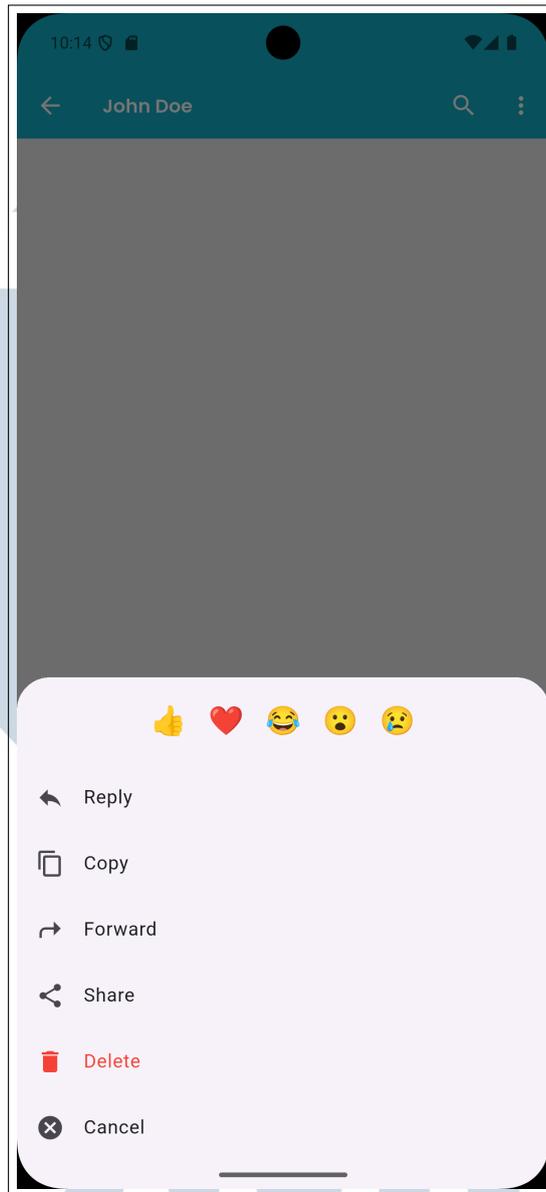
Gambar 3.34. Tampilan *overflow* menu di halaman *Chat View*



Gambar 3.35. Tampilan dialog peringatan untuk *Clear Chat*

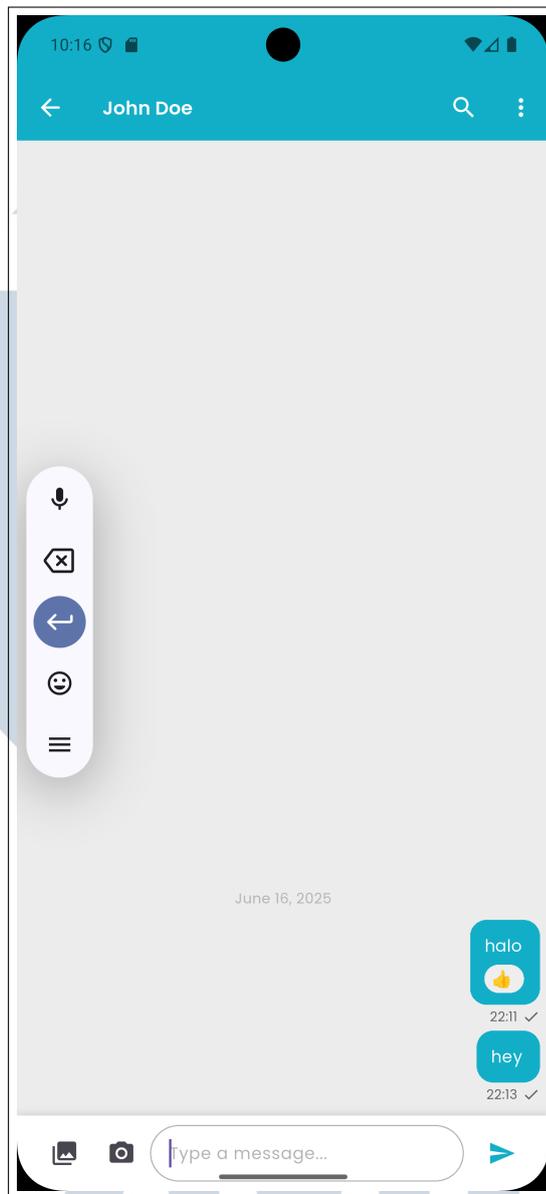
Di setiap elemen *Chat Bubble* yang ada di *Chat View*, terdapat *long press menu* yang bisa diakses seperti di Gambar 3.36.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



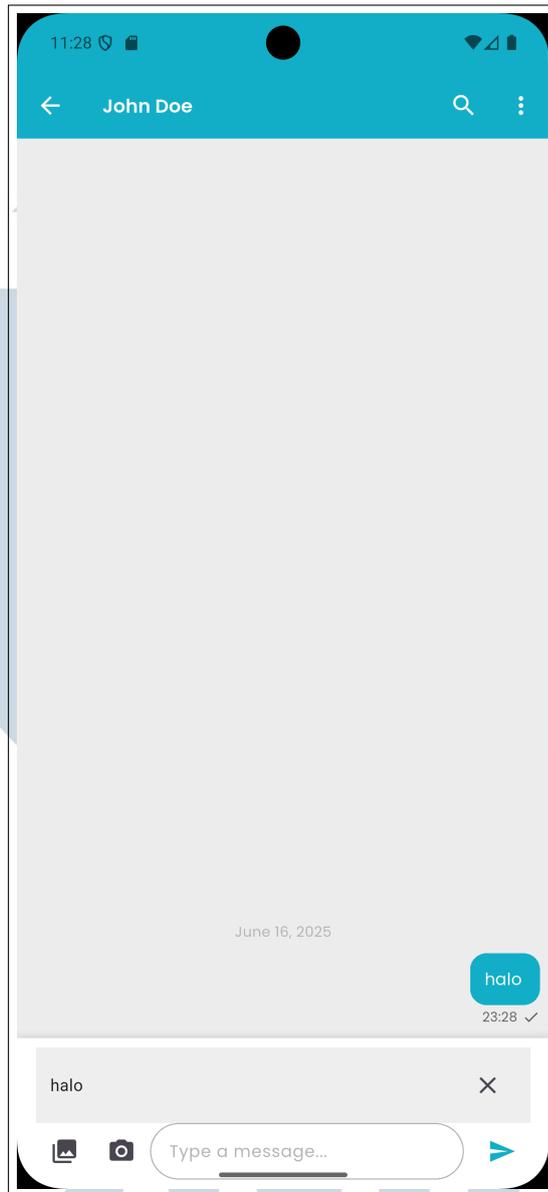
Gambar 3.36. Tampilan *long press menu* di Chat View

*Long press menu* ini memiliki beberapa aksi, dimulai dari reaksi *emoji* seperti di Gambar 3.36, dan di Gambar 3.37. Pilihan *Emoji* yang ditampilkan dipilih sesuai dari contoh dan diskusi dengan Bapak Rafli.



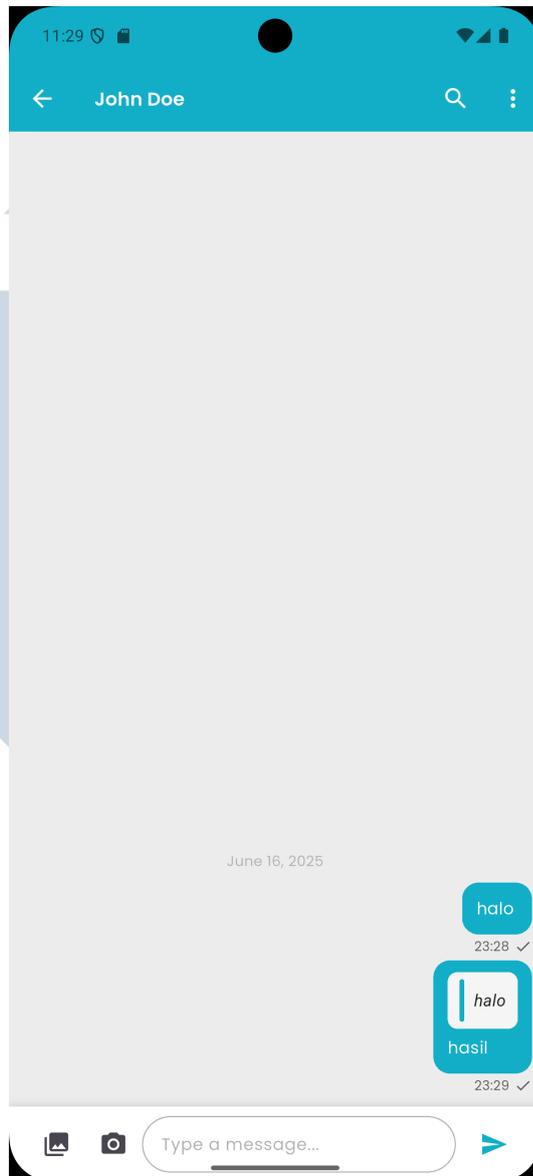
Gambar 3.37. Tampilan ketika pesan diberi reaksi *Emoji*

Opsi pertama dari *long press menu* ini adalah *Reply*, di mana pengguna bisa memilih pesan yang mereka secara spesifik ingin menanggapi. Fitur ini sudah cukup umum di aplikasi dan fitur *chatting* dari aplikasi lainnya. Contoh tampilan fitur ini bisa dilihat di Gambar 3.38, dan Gambar 3.39.



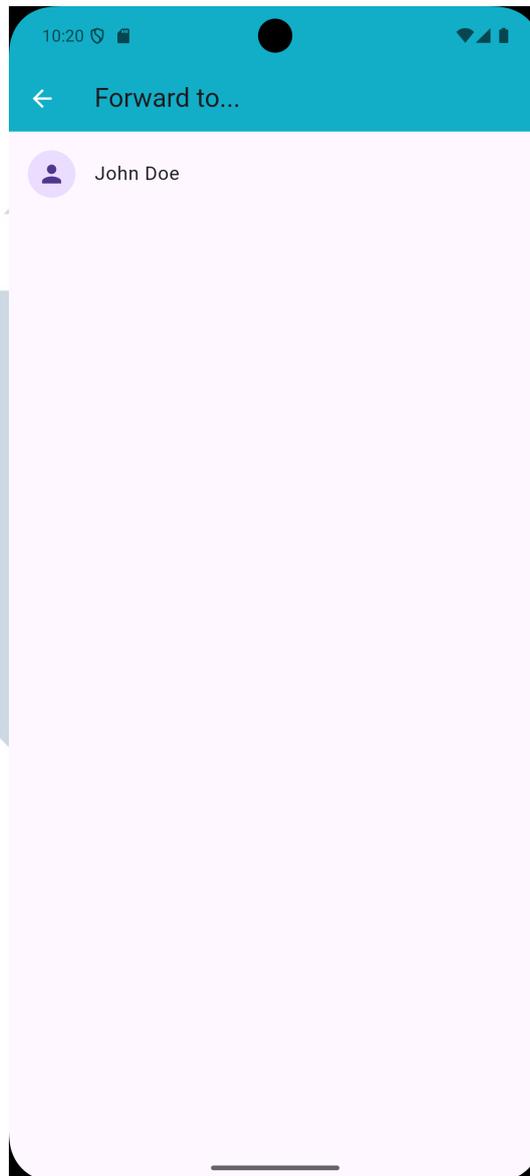
Gambar 3.38. Tampilan area ketik ketika sedang ingin *Reply* ke pesan tertentu

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



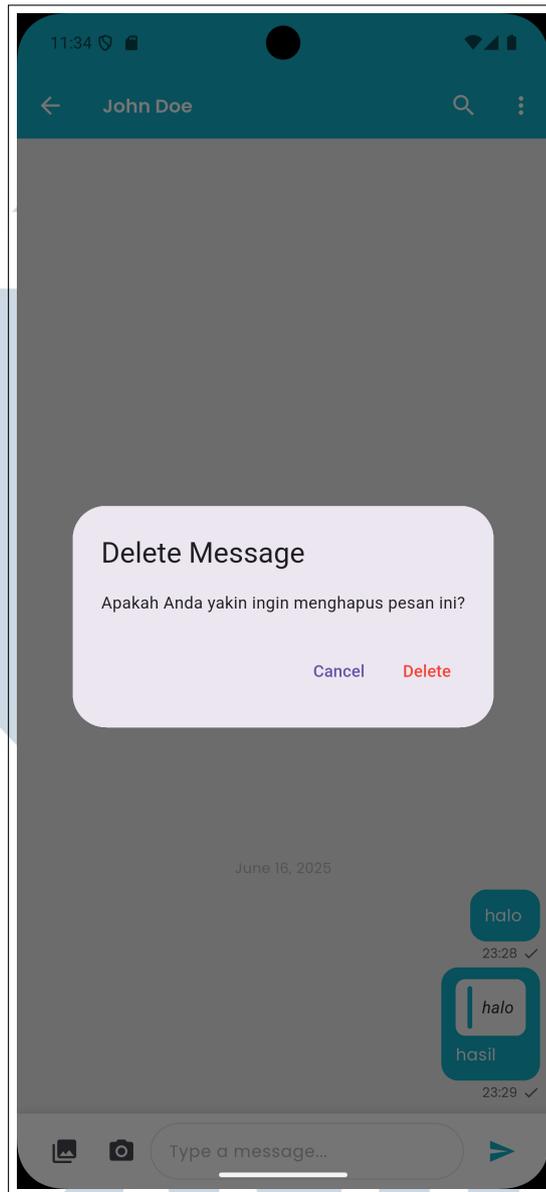
Gambar 3.39. Tampilan halaman ketika pesan ditanggapi melalui *Reply*

Opsi kedua dari *long press menu* adalah *Copy* yang akan menduplikat pesan yang terpilih, lalu diikuti dengan *Forward* yang akan membawa pengguna untuk memilih pengguna untuk melakukan *Forward* pesan. *Forward* hanya bisa dilakukan ke pengguna yang sudah ada percakapan sebelumnya. Contoh pemilihan pengguna untuk *Forward* ini bisa dilihat di Gambar 3.40.



Gambar 3.40. Contoh tampilan ketika melakukan *Forward*

Opsi selanjutnya adalah Share yang berperilaku sama dengan aplikasi lainnya ketika ingin menyebarkan suatu pesan baik teks maupun file media. Opsi terakhir dari menu long press ini adalah Delete, yang akan menghapus pesan yang ada. Opsi ini hadir dengan dialog peringatan karena perilakunya sebagai aksi yang tidak bisa diulang. Contoh tampilan terdapat di Gambar 3.41



Gambar 3.41. Tampilan dialog peringatan ketika melakukan *Delete Chat*

### 3.5 Kendala dan Solusi yang Ditemukan

Selama pengembangan yang dilakukan di aplikasi Maritim Muda Connect, terdapat beberapa kendala yang ditemukan. Kendala pertama yang ditemukan tentunya adalah saat mendapatkan akses dari repositori Maritim Muda Connect, di mana kode yang ada tidak bisa dijalankan karena ketidakcocokan versi *Java*. Seluruh tim *Mobile App Developer* memiliki *Java 21*, sehingga ditemukan solusi untuk melakukan pembaruan versi *Java* untuk meningkatkan skalabilitas, keamanan, dan kemudahan pengembangan untuk tim *Mobile Apps Developer*.

Kendala utama kedua selama masa pengembangan adalah sulitnya berkomunikasi dengan efektif karena status pekerjaan sebagai WFH. Beberapa kali terdapat pekerjaan yang tidak bisa dikomunikasikan secara efektif. Solusi sementara yang ditemukan adalah adanya beberapa hari kerja yang dilakukan di kantor.

Kendala ketiga dan terakhir adalah ketidakselesaian fitur *Chat* secara kepenuhan karena pekerjaan *backend* yang tidak selesai. Kendala ini terdiri dari beberapa faktor seperti keterlambatan dalam menerima akses ke *server*, kurangnya wawasan dalam merancang sistem *Chat*, dan yang lainnya. Faktor-faktor ini juga menyebabkan perancangan fitur dengan memasukkan data palsu, dan juga fungsi sementara yang tidak bisa dipakai di aplikasi asli atau dalam *production*.

