

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Pelaksanaan kerja magang bertempat di PT. Maybank Sekuritas Indonesia pada divisi Teknologi Informasi (TI) di bawah arahan Bapak Christofer Natadihardja selaku *OIC (Officer in Charge)* Departemen IT, serta supervisi magang. Selama kegiatan magang dibimbing oleh Faisal Abdillah yang berperan sebagai *Senior Backend Developer* sekaligus *PIC Project*. Aktivitas utama yang dilakukan meliputi penambahan fitur dan pemeliharaan pada situs *Collaboration Hub*. Dalam proses pelaksanaannya, koordinasi dilakukan secara intensif bersama pembimbing dan anggota tim melalui rapat harian yang diselenggarakan secara *offline*.

3.2 Tugas yang Dilakukan

Selama periode pelaksanaan magang, penulis mendapatkan arahan secara langsung dari Faisal Abdillah yang berperan sebagai *Senior Backend Developer* sekaligus *PIC Project*. Pada kegiatan magang ini, teknologi utama yang digunakan dalam pengembangan adalah *framework React* dengan *TypeScript*.

Tugas yang diberikan meliputi beberapa aktivitas utama, yaitu:

1. Melakukan proses pengembangan perangkat lunak (*coding*) menggunakan *framework React TypeScript* dalam rangka penambahan fitur baru serta pemeliharaan (*maintenance*) situs yang sudah ada.
2. Melaksanakan pengujian secara manual untuk memastikan fungsionalitas sistem berjalan dengan baik serta mendeteksi kemungkinan adanya kesalahan (*bug*).

3.3 Uraian Pelaksanaan Magang

Pada bagian ini dijelaskan tahapan pelaksanaan kegiatan magang yang berlangsung secara mingguan. Uraian yang disajikan meliputi deskripsi pekerjaan yang telah diselesaikan, hambatan atau permasalahan yang muncul selama proses, serta strategi penyelesaian yang diterapkan untuk mengatasi kendala tersebut.

Penyajian ini diharapkan dapat memberikan gambaran yang lebih sistematis mengenai perkembangan pengalaman serta peningkatan kompetensi yang diperoleh selama periode magang. Perlu ditekankan bahwa data yang ditampilkan merupakan data *testing* dan tidak memiliki keterkaitan dengan data nasabah, beberapa data penting akan di *blur* untuk menjaga integritas data.

3.3.1 Proses Pelaksanaan

Pelaksanaan kerja magang berlangsung selama 18 minggu atau 4 bulan, dengan kegiatan yang ditampilkan pada Tabel 3.1.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu

| Minggu Ke - | Pekerjaan yang di lakukan |
|--------------------------------|---|
| 1 | Implementasi <i>role</i> Ops dan Repo, termasuk penambahan halaman <i>API Client</i> , <i>API Client Account</i> , serta <i>Collateral Reverse Repo</i> beserta fitur <i>CRUD</i> . Melakukan integrasi <i>frontend</i> dengan <i>backend</i> untuk kedua <i>role</i> tersebut. |
| 2 | Pengembangan fitur <i>Collateral Details (List, Create, Edit)</i> serta integrasi dengan <i>backend</i> . Menyelesaikan kendala <i>error</i> pada proses <i>edit</i> dan menambahkan fitur <i>generate closing report</i> . |
| 3 | Penyelesaian integrasi untuk <i>role</i> Repo. Pengembangan menu baru <i>Broadcast Email</i> dan integrasinya, serta penambahan halaman <i>Trade Settlement: Trade Confirmation</i> dan <i>Married Trade</i> . |
| 4 | Restrukturisasi menu <i>Trade Settlement</i> menjadi <i>Contracting</i> dan <i>Contracted</i> . Perbaikan <i>UI</i> , penambahan <i>endpoint</i> baru, serta integrasi <i>API</i> untuk <i>Trade Confirmation</i> dan <i>Married Trade</i> , termasuk fitur <i>Re-Trigger Data</i> . |
| 5 | <i>Update endpoint API</i> pada <i>Trade Confirmation</i> dan <i>Married Trade</i> , penambahan <i>API</i> untuk status <i>upload</i> , integrasi <i>upload file</i> , <i>export ZIP</i> , pembagian submenu <i>Contracting</i> , serta perbaikan <i>logic</i> dan <i>date picker</i> . |
| Lanjut pada halaman berikutnya | |

Tabel 3.1 Pekerjaan yang dilakukan tiap minggu (lanjutan)

| Minggu Ke - | Pekerjaan yang di lakukan |
|--------------------------------|---|
| 6 | Penambahan <i>filter</i> pada <i>Trade List</i> , integrasi <i>API</i> pada <i>Insti Retail</i> , <i>SG</i> , dan <i>S-Invest</i> . Perbaikan <i>searchbar</i> dan interval <i>API Married Trade</i> , serta penambahan <i>loader</i> untuk <i>Re-Trigger Data</i> . |
| 7 | Fokus pada <i>testing API</i> dan <i>bug fixing</i> . Penambahan menu <i>Client List</i> , <i>Price Adjustment</i> , serta komponen <i>download file (Excel, PDF, TXT)</i> . |
| 8 | <i>Update UI</i> dan integrasi <i>API</i> untuk <i>Client List</i> , <i>Trade List</i> , dan <i>Insti Retail</i> . Penambahan submenu <i>Data Order</i> , <i>update</i> komponen <i>Dropdown</i> , <i>Searchbar</i> , serta integrasi <i>Data Order</i> dan fitur <i>download Excel</i> . |
| 9 | Pengembangan fitur <i>edit</i> , <i>delete</i> , <i>filter</i> pada <i>Data Order</i> beserta integrasi <i>API</i> . Penambahan tombol <i>Update Client List</i> , perbaikan <i>bug filter</i> , penambahan menu <i>Reporting</i> , serta <i>update</i> komponen <i>multiple upload</i> . |
| 10 | Pembagian menu <i>Reporting</i> menjadi <i>Email Setting</i> dan <i>Send Email</i> . Pengembangan <i>UI</i> , fitur <i>CRUD Email Setting</i> , integrasi <i>API</i> , <i>refactor</i> kode, serta penambahan halaman <i>Show detail Email Setting</i> . |
| 11 | Integrasi <i>API</i> dan <i>refactor Send Email</i> . Penambahan tombol <i>Upload Care</i> dan <i>DMA</i> pada <i>Data Order</i> . <i>Update endpoint API Client List</i> dan <i>Account</i> , penambahan submenu <i>Client Management List</i> , serta halaman <i>Create/Edit</i> untuk <i>Client Management</i> . |
| 12 | Integrasi <i>Create/Edit Client Management List</i> , <i>update Price Adjustment Show</i> , penambahan halaman <i>Broker Ranking</i> , integrasi <i>API</i> , <i>download PDF Price Adjustment</i> , perbaikan <i>bug Email Setting</i> , dan fitur <i>paste Data Order</i> . |
| 13 | Penambahan ekspor <i>Excel Broker Ranking</i> dan <i>update Price Adjustment (Top Price Adjustment)</i> . Perbaikan <i>UI Price Adjustment</i> , <i>refactor</i> format angka dan <i>filter</i> , penambahan <i>global loader</i> , serta penyeragaman <i>style</i> tabel. |
| Lanjut pada halaman berikutnya | |

Tabel 3.1 Pekerjaan yang dilakukan tiap minggu (lanjutan)

| Minggu Ke - | Pekerjaan yang di lakukan |
|-------------|--|
| 14 | Penambahan <i>upload report</i> di <i>Broker Ranking</i> , kondisi <i>export Married Trade</i> , penambahan <i>Crosstom Allocation</i> , <i>fixing logic contracted list</i> , <i>update API Data Order</i> , <i>multiple download TC</i> , serta integrasi dan fitur <i>add row</i> pada <i>Crosstom Allocation</i> . |
| 15 | Pengembangan komponen <i>Chip</i> di <i>Email Setting</i> , konfigurasi <i>TinyMCE</i> dan <i>dataProvider</i> , pembuatan halaman <i>Currency Management</i> beserta <i>CRUD</i> , <i>fixing Married Trade logic</i> , serta integrasi tombol <i>export Crosstom</i> . |
| 16 | Penambahan fitur <i>merge cell</i> pada <i>Data Order</i> , perbaikan <i>bug Currency Management</i> , <i>research auto select file upload</i> , <i>progress loading Married Trade</i> , serta <i>drag and drop upload</i> pada <i>Send Email</i> . |
| 17 | Penyelesaian fitur <i>merge cell</i> dan <i>split Data Order</i> , penyelesaian <i>auto select file upload</i> , penambahan halaman <i>Right Transaction</i> beserta integrasi, serta melakukan <i>testing menyeluruh aplikasi Trade Settlement</i> . |

Pengembangan fitur *trade settlement* dilakukan dengan dukungan perangkat lunak dan perangkat keras yang sesuai. Adapun perangkat lunak yang digunakan dalam proses implementasi adalah sebagai berikut:

1. React Framework versi 18.2.0.
2. Typescript versi 5.1.6.
3. Tailwind CSS versi 3.4.17.
4. Visual Studio Code versi 1.99.2.
5. GitHub Desktop versi 3.4.18 (arm64).
6. Sistem operasi macOS Sonoma versi 14.5.

Sedangkan perangkat keras yang digunakan selama pelaksanaan kegiatan magang adalah sebuah laptop Macbook Pro M2 16 inci dengan spesifikasi berikut:

1. Chipset Apple M2 Pro.
2. Memori (RAM) sebesar 16 GB.
3. Penyimpanan internal (storage) berkapasitas 512 GB.

3.3.2 Pengembangan Fitur Trade Settlement

Pada bagian ini, akan dijelaskan mengenai proses pengembangan fitur *trade settlement*. Fitur ini bertujuan untuk memangkas alur kerja yang tidak efisien, mengotomatisasi proses yang berulang, serta menyusun kembali data agar lebih terstruktur. Implementasi fitur ini juga diharapkan dapat meningkatkan akurasi dalam pencatatan transaksi sekaligus meminimalisasi potensi terjadinya kesalahan manusia. Ditekankan kembali bahwa data yang ditampilkan merupakan data *testing* dan tidak memiliki keterkaitan dengan data nasabah, beberapa data penting akan di *blur* untuk menjaga integritas data.

A Software Development Life Cycle (SDLC)

Dalam proyek ini, metode pengembangan sistem yang diterapkan adalah *Software Development Life Cycle (SDLC)* menggunakan pendekatan *Agile*. *Agile* dipilih karena karakteristiknya yang iteratif, fleksibel, dan mampu beradaptasi dengan perubahan kebutuhan sistem. Penerapan *Agile* dilakukan dengan menggunakan mekanisme *sprint*, yaitu siklus kerja yang terstruktur dalam periode waktu tertentu untuk menyelesaikan sejumlah pekerjaan yang telah ditentukan. Tiap *sprint* mengikuti alur perencanaan, pengembangan, evaluasi, serta perbaikan berkelanjutan. Model ini memberikan kemudahan bagi tim untuk melakukan penyesuaian secara responsif terhadap dinamika kebutuhan bisnis, sekaligus menjamin kualitas setiap fitur yang dikembangkan tetap konsisten. Visualisasi alur pengembangan yang terbagi dalam beberapa fase *sprint* berdasarkan periode mingguan dan bulanan ditampilkan dalam bentuk *Gantt chart* pada Gambar 3.1.



Gambar 3.1. Gantt chart project trade settlement

Pada bulan Mei, proyek dimulai dengan dua minggu pertama yang berfokus pada kegiatan *Research* dan *Transfer Knowledge*. Pada fase ini dilakukan analisis

kebutuhan, studi literatur, serta *knowledge sharing* terkait sistem dan teknologi yang akan digunakan. Memasuki minggu ketiga, aktivitas berlanjut pada *Database Configuration*, yang meliputi perancangan *ERD*, pembuatan *database schema*, dan penyiapan lingkungan *database*. Tidak lupa juga dilakukan *development* awal untuk mempersiapkan fondasi *backend* dan *frontend* sebagai langkah awal pengembangan sistem utama. Pada minggu keempat, proses pembuatan menu *contracting*, *contracted*, dan *married trade* dilakukan.

Di bulan Juni, *development* mulai memasuki tahap yang lebih teknis dan berorientasi pada implementasi fitur. Pada minggu pertama dan kedua, aktivitas utamanya yaitu melanjutkan *development* serta integrasi menu *contracting*, *contracted*, dan *married trade*. Proses ini melibatkan penyusunan komponen tampilan, penyesuaian alur transaksi, serta penghubungan dengan *backend* menggunakan *api* yang sudah disiapkan. Selanjutnya, di minggu ketiga diarahkan pada *development* dan integrasi menu *Price Adjustment* dan *Trade List*, sehingga kedua menu tersebut dapat memproses data transaksi secara konsisten. Pada minggu keempat, ruang lingkup pekerjaan diperluas mencakup pembuatan dan integrasi menu *Client List*, *Data Order*, dan *S-Invest*.

Pada bulan Juli, proses *development* dilanjutkan dengan cakupan fitur yang semakin kompleks. Di minggu pertama difokuskan untuk menyelesaikan dan menstabilkan integrasi menu *Client List*, *Data Order*, dan *S-Invest*, sehingga seluruh proses transaksi dan pengelolaan data dapat berjalan secara konsisten. Pada minggu kedua dan ketiga, kegiatan beralih pada pengembangan menu *Client List Management*, *Email Setting*, dan *Send Email*, seluruh fitur diuji dalam lingkup *sprint* untuk memastikan kompatibilitas antar menu. Selanjutnya, minggu keempat diarahkan pada pembuatan dan integrasi menu *Broker Ranking*, *Crosstom Allocation*, dan *Currency Management*, yang membutuhkan validasi data agregasi, alokasi transaksi, serta pengaturan mata uang. Pada bulan ini, tim berfokus pada peningkatan cakupan fitur secara menyeluruh dan memastikan setiap menu berfungsi sesuai kebutuhan bisnis.

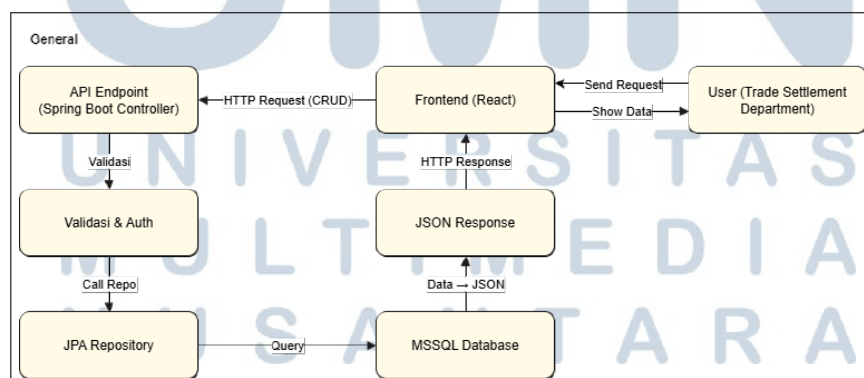
Kegiatan pada bulan Agustus diawali dengan penyelesaian penuh untuk menu yang dikerjakan pada bulan sebelumnya. Minggu pertama dan kedua digunakan untuk finalisasi *development* dan integrasi menu *Broker Ranking*, *Crosstom Allocation*, dan *Currency Management*, termasuk penyempurnaan tampilan, optimalisasi performa *query*, serta konsolidasi data antar menu. Memasuki minggu ketiga dan keempat, fokus pekerjaan bergeser ke tahap *Bug Fixing*, *Maintenance* setiap menu, dan pelaksanaan *Testing*. Pada fase ini

dilakukan pengujian menyeluruh secara manual untuk memastikan seluruh fitur stabil. Evaluasi performa dan perbaikan berulang dilakukan sebagai persiapan menuju tahap *deployment*.

Pada minggu pertama bulan September, proyek memasuki fase akhir yaitu *Production Setup* dan *System Deployment*. Kegiatan mencakup *proses build* dan *deployment* aplikasi, serta verifikasi akhir terhadap seluruh fungsi sistem. Selain itu, tim juga menyiapkan dan memberikan *user training* untuk memastikan *user* memahami cara operasional sistem secara optimal. Tahap ini menjadi penutup seluruh rangkaian pengembangan sebelum sistem resmi dioperasikan pada lingkungan *production*.

B Interaksi Frontend dengan Backend

Pada sistem *trade settlement* ini, *Frontend* (FE) menjadi bagian yang langsung berinteraksi dengan *user*, khususnya tim *Trade Settlement Department*. Melalui FE, *user* bisa melihat data, melakukan pencarian, serta menambah, mengubah, atau menghapus informasi yang dibutuhkan. Sebaliknya, *Backend* (BE) menghubungkan FE ke *database* dan memastikan bahwa *API* aplikasi dapat memproses permintaan data atau perubahan *user* dengan benar. BE dikembangkan menggunakan *Spring Boot Framework* berbasis Java karena strukturnya yang stabil, mendukung integrasi dengan berbagai modul, dan mudah diatur untuk kebutuhan sistem bisnis. BE menggunakan *Java Persistence API (JPA)* sebagai lapisan untuk mengelola data dan menghubungkan logika bisnis dengan *database Microsoft SQL Server*. Seperti yang ditunjukkan pada Gambar 3.2, proses komunikasi antara FE dan BE.

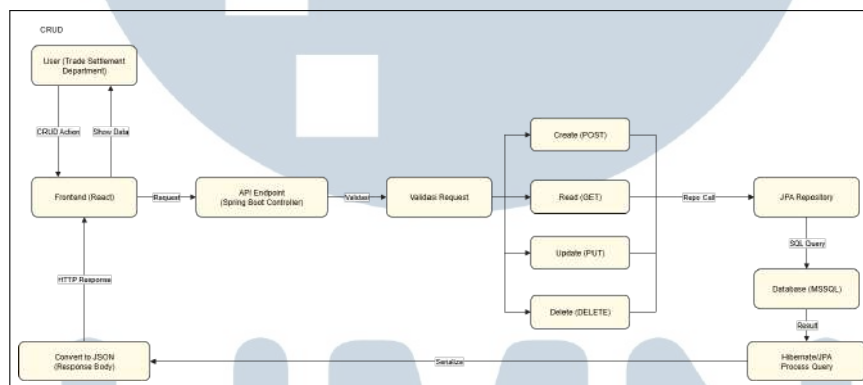


Gambar 3.2. Flow connection between FE and BE

Proses komunikasi antara FE dan BE dimulai saat *user* melakukan sesuatu

di aplikasi, seperti menampilkan data atau menyimpan entri baru. Selanjutnya, FE mengirimkan permintaan ke *API Endpoint* yang sesuai. Setelah itu, BE akan memeriksa data yang diterima untuk memastikan bahwa data tersebut sesuai dengan format dan aturan yang tepat. Jika valid, permintaan tersebut akan dikirim ke *repository JPA* untuk diproses menjadi *query* dan dieksekusi oleh *database*. Setelah itu, hasilnya dikembalikan dalam bentuk data yang telah dikonversi menjadi format *JSON*.

Selain untuk menampilkan data, sistem *trade settlement* ini juga mendukung operasi *CRUD* yang memungkinkan *user* berinteraksi secara menyeluruh dengan data yang ada. Artinya, FE tidak hanya dapat membaca data, tetapi juga dapat mengubah, menambah, atau menghapus data sesuai kebutuhan operasional. Pola kerja yang serupa digunakan untuk setiap tindakan *CRUD*, seperti yang ditunjukkan pada Gambar 3.3.



Gambar 3.3. Flow connection between FE and BE in CRUD action

Setelah FE mengirimkan *request* ke BE sesuai jenis operasinya, BE menjalankan validasi dan otorisasi sebelum meneruskan proses ke lapisan *repository JPA*. Setelah *query* dijalankan pada *database*, hasilnya dikembalikan ke BE dalam bentuk data mentah yang dikonversi menjadi *JSON response*, dan kemudian dikirim ke FE dan ditampilkan secara langsung di *interface user*.

Dengan alur seperti ini, interaksi antara FE dan BE berjalan secara efisien dan terintegrasi. Setiap proses data berlangsung secara otomatis, cepat, dan tetap terjaga konsistensinya, sehingga mendukung kelancaran operasional sistem secara menyeluruh.

C Keamanan Data

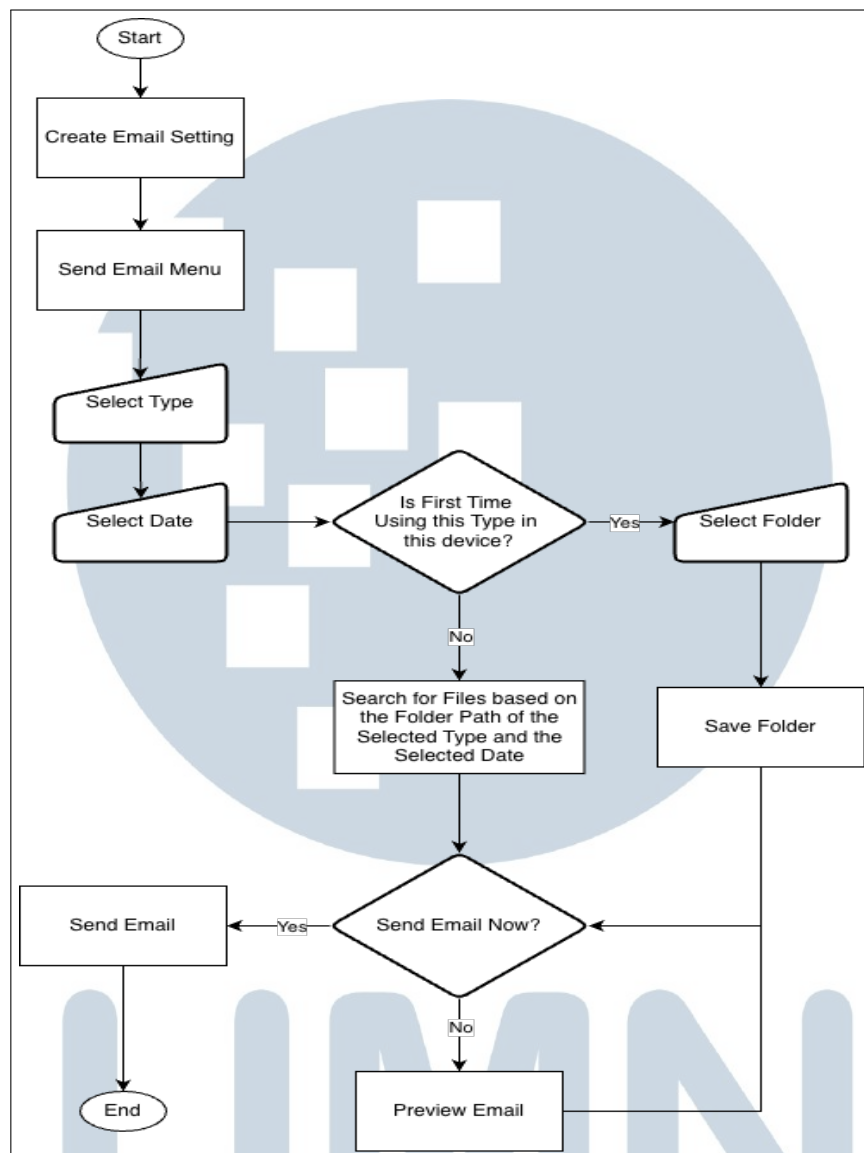
Data pada sistem ini diamankan di dalam server yang dilengkapi dengan berbagai *security tools* seperti *Imperva* untuk perlindungan aplikasi web, serta konfigurasi *firewall* yang membatasi akses tidak sah dari luar jaringan. Selain itu, akses terhadap data dan web internal hanya dapat dilakukan melalui koneksi internet perusahaan, baik menggunakan kabel LAN maupun *Wi-Fi* resmi yang terhubung ke jaringan internal. Penggunaan jaringan ini memastikan bahwa seluruh aktivitas komunikasi data berlangsung dalam lingkungan yang aman dan terkontrol.

Setiap perangkat yang digunakan untuk mengakses sistem juga telah dilengkapi dengan lapisan proteksi tambahan seperti *ZScaler* untuk keamanan jaringan berbasis *cloud*, *Forcepoint DLP* untuk mencegah kebocoran data, serta *McAfee* sebagai perlindungan terhadap ancaman *malware* dan virus. Sebagai langkah pengamanan tambahan, akses ke layanan yang tidak berhubungan dengan operasional perusahaan seperti *Google Service*, *YouTube*, dan *WhatsApp* diblokir. Dengan penerapan kebijakan dan teknologi ini, sistem mampu menjaga kerahasiaan, integritas, serta ketersediaan data dengan tingkat keamanan yang tinggi.

D Otomatisasi

Otomatisasi yang diterapkan pada sistem ini membawa perubahan besar dibandingkan cara kerja manual sebelumnya. Pada menu *Married Trade* di aplikasi sebelumnya, *user* harus menarik data dari *Trade Database* secara manual setiap kali dibutuhkan. Saat ini, proses tersebut sudah digantikan oleh *scheduler* yang otomatis mengambil data dari *Trade Database*, sehingga pekerjaan jadi lebih cepat dan tidak lagi bergantung sepenuhnya pada intervensi *user*. Namun, jika terjadi permasalahan pada *scheduler*, telah disediakan pula tombol untuk menarik data dari *Trade Database*.

Pada menu *Send Email*, sistem memiliki kemampuan untuk melakukan *auto select file*, yaitu proses pemilihan *file* secara otomatis berdasarkan tipe dan tanggal yang dipilih *user*. Fitur ini memungkinkan sistem untuk mencari *file* pada *folder* yang telah terdaftar sebelumnya tanpa perlu melakukan intervensi manual setiap kali *user* ingin mengirim email. Dengan demikian, sistem dapat mempercepat proses pengambilan berkas dan mengurangi kesalahan pengiriman. Otomasi ini adalah komponen penting dalam meningkatkan efisiensi kerja *user*.



Gambar 3.4. Flow Otomatisasi Email

Tetapi pada awal penggunaan setiap tipe pada *device*, sistem masih memerlukan proses *upload folder*. Tahap ini berfungsi sebagai *whitelisting folder*, di mana *folder* yang telah diunggah disimpan sebagai referensi dan dapat diakses oleh sistem melalui *interface web*. Prosedur ini memastikan bahwa sistem memiliki izin akses yang tepat ke lokasi penyimpanan *file* yang akan digunakan untuk pengiriman email. Setelah *folder* disimpan, *user* tidak perlu mengulangi *upload folder* pada penggunaan dengan tipe yang sama di percobaan berikutnya.

Setelah *folder* di-*whitelist*, sistem dapat secara otomatis mencari *file* berdasarkan tipe dan tanggal yang telah dipilih tanpa perlu *user* melakukan apa

pun lagi. Karena *user* hanya perlu memilih parameter yang relevan, proses pengiriman email menjadi lebih cepat dan praktis. Selain itu, sistem menawarkan fitur *preview* email sebelum dikirim, sehingga *user* dapat memastikan bahwa isi dan lampiran email telah sesuai. Mekanisme ini mempercepat dan mengurangi kesalahan pengiriman.

Pada menu *Crosstom Allocation*, sebelumnya *user* harus melakukan pengecekan baris demi baris pada data yang belum *match order* secara manual. Saat ini, proses tersebut langsung dikerjakan oleh *Backend*, dan hasilnya otomatis ditampilkan di *Frontend*. Hal ini membuat *user* cukup melihat hasil akhirnya saja tanpa perlu menghitung ulang. Perhitungan dilakukan secara sistematis dalam beberapa tahap. Ini memungkinkan untuk menghitung hasil *crossing* antara sisi *buy* dan *sell* dengan benar. Proses ini dilakukan oleh *backend* dalam urutan berikut:

1. Data diurutkan mulai dari kolom *buySell* secara menurun *descending* dan kolom *netVolume* secara *ascending*.
2. Kemudian, sistem membandingkan nilai *netVolume* terbesar dari sisi *buy* dengan nilai *netVolume* terbesar dari sisi *sell*.
3. Nilai yang lebih kecil dari perbandingan ini dimasukkan ke dalam kolom *crossVolume*.
4. Setelah itu, nilai *netVolume* dikurangi dengan nilai *crossVolume*, dan hasilnya disimpan ke kolom *netCrossVolume*.
5. Jika *netCrossVolume* masih belum bernilai nol atau habis, maka proses akan diulang kembali dari langkah pertama hingga seluruh data selesai diproses.

Data yang telah diproses pada langkah keempat akan ditempatkan di bagian atas tabel sesuai dengan urutan *buySell*. Dengan cara ini, sistem memastikan bahwa volume setiap pasangan transaksi *buy* dan *sell* secara proporsional dihitung, dan tidak ada volume yang masih belum dicocokkan.

Selain itu, ada tombol *export/download* pada beberapa menu, yang membuat pembuatan laporan menjadi lebih mudah. Saat ini, laporan dapat diunduh sesuai format dengan sekali klik, tidak seperti sebelumnya, di mana data harus disalin dari *Excel* lalu ditempel ke aplikasi yang ada sebelum diekspor. Secara keseluruhan, otomatisasi alur kerja menjadi jauh lebih mudah, mengurangi kemungkinan kesalahan, dan meningkatkan kecepatan serta akurasi proses bisnis.

E Flowchart

Flowchart pada bagian ini digunakan untuk memperlihatkan alur kerja dari fitur *trade settlement*. Dengan adanya diagram ini, proses ini memudahkan dalam memahami tahapan yang saling berkaitan secara runtut. Penyajian dalam bentuk visual juga membantu dalam menemukan kemungkinan kendala atau titik lemah pada proses yang berjalan. Pada akhirnya, *flowchart* menjadi panduan sederhana namun efektif untuk mendukung analisis dan pengembangan sistem.

E.1 Flowchart Trade List

Menu *Trade List* merupakan salah satu fitur utama dalam sistem *Trade Settlement* yang berfungsi untuk menampilkan seluruh data transaksi perdagangan yang telah tercatat di sistem. Melalui menu ini, *user* dapat melihat transaksi harian yang dilakukan oleh nasabah. Selain itu, menu ini juga menyediakan fitur pengeditan untuk memperbarui informasi perdagangan apabila ditemukan kesalahan atau perubahan data. Pada Gambar 3.5 ditunjukkan *flowchart* untuk menu *Trade List*.



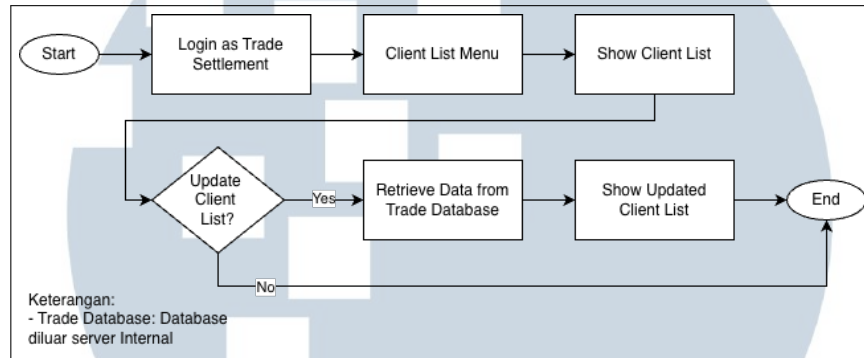
Gambar 3.5. Flowchart Trade List

Berdasarkan Gambar 3.5, sistem akan menampilkan daftar transaksi perdagangan yang tersedia. Jika *user* ingin melakukan perubahan pada salah satu data, maka *user* dapat langsung mengubah data pada list tersebut kemudian menekan tombol *save*. Setelah proses pengeditan selesai, data yang telah diperbarui akan ditampilkan kembali dalam daftar transaksi. Jika tidak ada perubahan yang dilakukan, proses akan langsung berakhir setelah daftar transaksi ditampilkan.

E.2 Flowchart Client List

Menu *Client List* merupakan bagian penting dalam sistem *Trade Settlement* yang berfungsi untuk menampilkan daftar nasabah yang terdaftar di dalam sistem.

Fitur ini dirancang untuk menjaga konsistensi dan keakuratan data nasabah dengan cara melakukan sinkronisasi otomatis terhadap *Trade Database*. Dengan demikian, *user* tidak perlu melakukan pembaruan data secara manual, sehingga risiko kesalahan input dapat diminimalkan. Pada Gambar 3.6 menunjukkan *flowchart* untuk menu *Client List*.

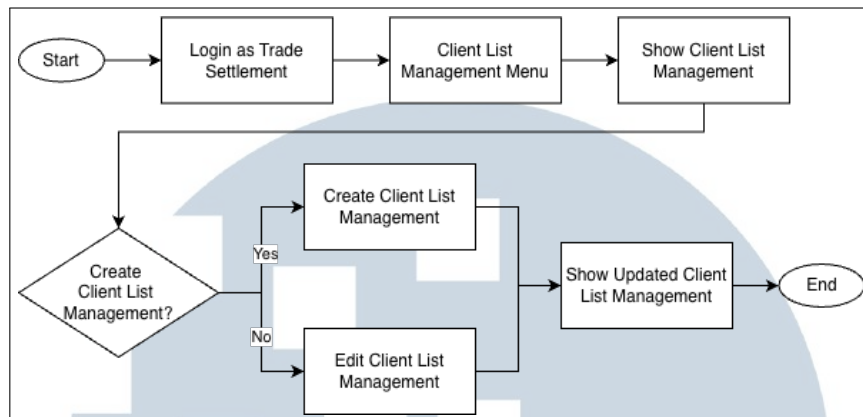


Gambar 3.6. Flowchart Client List

Berdasarkan Gambar 3.6, sistem akan menampilkan daftar nasabah yang telah tersimpan di database. Jika *user* menekan tombol *Update Client List*, sistem akan mengambil data terbaru dari *Trade Database* dan memperbarui daftar nasabah yang ditampilkan. Setelah proses pembaruan selesai, daftar nasabah yang telah diperbarui ditampilkan kembali di layar. Jika tidak ada pembaruan dilakukan, maka sistem hanya menampilkan daftar nasabah yang sudah ada dan proses berakhir.

E.3 Flowchart Client List Management

Pada Gambar 3.7 memperlihatkan alur proses pada menu *Client List Management*, yang berfungsi untuk mengelola data nasabah di dalam sistem *Trade Settlement*. Melalui menu ini, *user* dapat melakukan pembuatan data nasabah baru maupun memperbarui data nasabah yang sudah ada. Fitur ini dirancang agar administrasi data nasabah menjadi lebih efisien, akurat, dan mudah dikontrol, sehingga seluruh informasi yang tersimpan selalu sesuai dengan kondisi terkini.

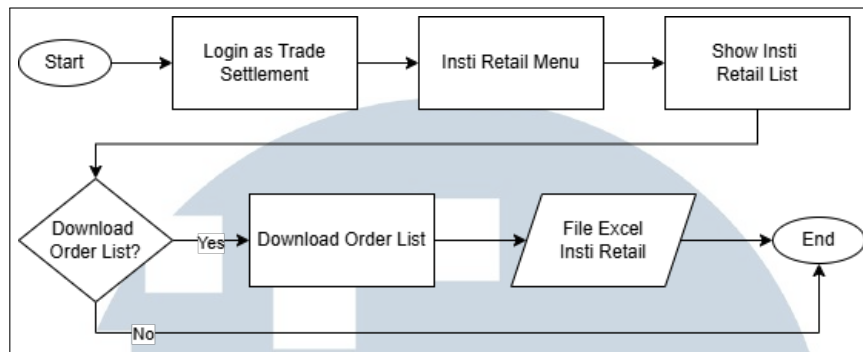


Gambar 3.7. Flowchart Client List Management

Proses pada menu ini dimulai ketika *user* memilih untuk membuat data nasabah baru melalui proses *Create Client List Management* atau memperbarui data yang sudah ada melalui *Edit Client List Management*. Selanjutnya, setelah proses pembuatan atau pengeditan selesai dilakukan, sistem akan menampilkan hasil perubahan. Tahapan ini memastikan bahwa seluruh data yang dimasukkan atau diperbarui telah tersimpan dengan benar dan siap digunakan untuk proses bisnis berikutnya. Dengan adanya alur ini, sistem dapat menjaga konsistensi serta akurasi data nasabah secara efisien, sehingga mendukung kelancaran kegiatan operasional di bagian *Trade Settlement*.

E.4 Flowchart Insti Retail

Pada Gambar 3.8 menampilkan alur proses pada menu *Insti Retail*, yang merupakan bagian dari sistem *Trade Settlement* yang berfungsi untuk menampilkan daftar transaksi ritel institusi yang telah tersimpan di dalam sistem. Menu ini menyediakan fitur *download order list* dalam format *Excel*, sehingga mempermudah proses analisis data dan pelaporan hasil transaksi. Fitur ini dibuat untuk mendukung efisiensi kerja dengan mempercepat akses terhadap data tanpa memerlukan input manual.



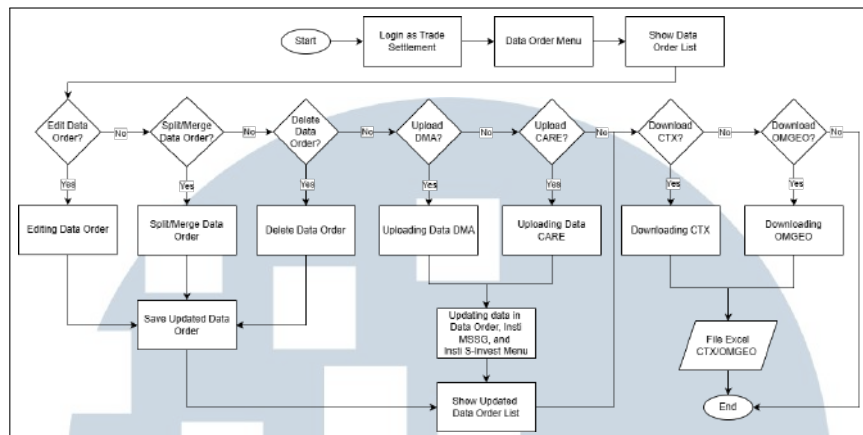
Gambar 3.8. Flowchart Insti Retail

Pada menu ini, *user* dapat melihat daftar transaksi *Insti Retail* yang tersimpan dalam sistem. Jika dibutuhkan, *user* juga dapat mengunduh *order list* dalam bentuk *file Excel* sehingga mempermudah proses analisis maupun pelaporan. Fitur ini dirancang untuk mempercepat akses terhadap data transaksi tanpa perlu melakukan pencatatan manual. Dengan demikian, alur ini mendukung efisiensi kerja sekaligus menjaga akurasi data yang digunakan dalam kegiatan operasional sehari-hari.

E.5 Flowchart Data Order

Pada Gambar 3.9 menunjukkan alur utama proses pada menu *Data Order* yang berfungsi untuk mengelola berbagai aktivitas terkait data transaksi. Melalui menu ini, *user* dapat melakukan berbagai tindakan seperti mengedit, menghapus, maupun memisahkan dan menggabungkan data order agar sesuai dengan kondisi transaksi terkini. Selain itu, *user* juga dapat melakukan *upload file DMA* dan *CARE* untuk memperbarui data transaksi secara otomatis dari sumber eksternal.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

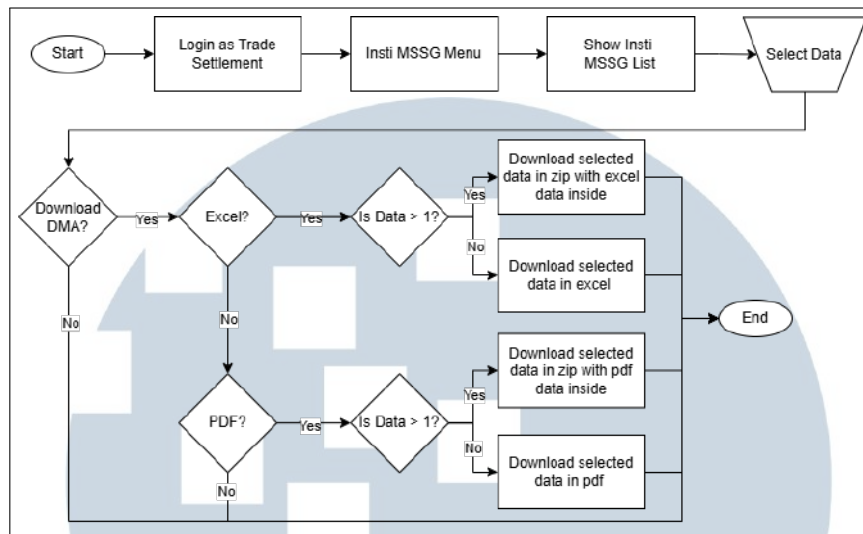


Gambar 3.9. Flowchart Data Order

Ketika daftar data order ditampilkan, *user* dapat memilih berbagai tindakan seperti melakukan *edit*, *split/merge*, atau *delete* pada data yang tersedia. Selain itu, *user* juga dapat mengunggah *file DMA* dan *CARE* untuk memperbarui data transaksi, serta *download file CTX* dan *OM-GEO* dalam format *Excel* sebagai sarana pelaporan atau verifikasi data. Setiap tindakan yang dilakukan akan memperbarui data tidak hanya pada menu *Data Order*, tetapi juga pada menu lain seperti *Insti MSSG* dan *Insti S-Invest*. Dengan adanya alur ini, seluruh proses pengelolaan data order menjadi lebih terintegrasi, efisien, dan mampu memastikan keakuratan data yang digunakan dalam kegiatan operasional perusahaan.

E.6 Flowchart Insti MSSG

Selanjutnya, Gambar 3.10 menunjukkan alur pada menu *Insti MSSG*, yang berfungsi untuk menampilkan dan mengunduh data hasil alokasi dari *upload file DMA*. Melalui menu ini, *user* dapat mengunduh data dalam berbagai format, baik *Excel* maupun *PDF*, sesuai dengan kebutuhan. Fitur ini dirancang untuk memudahkan *user* dalam memperoleh dan memverifikasi data transaksi tanpa perlu melakukan pengolahan manual.

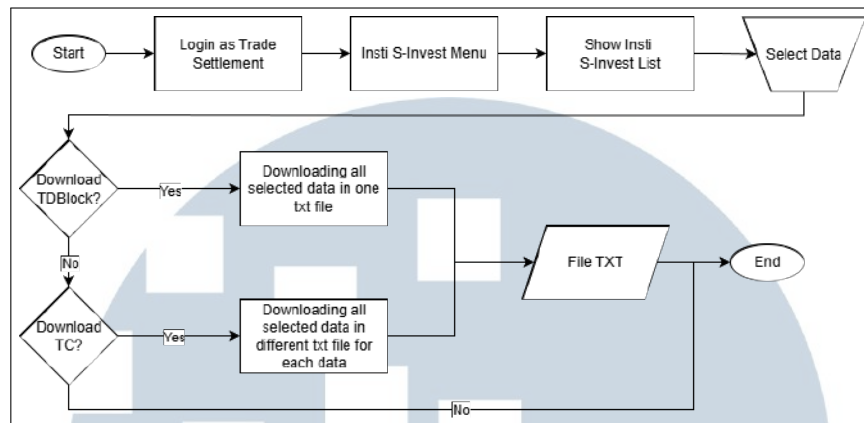


Gambar 3.10. Flowchart Insti MSSG

Berdasarkan Gambar 3.10, setelah menampilkan daftar data hasil alokasi, *user* dapat memilih data yang ingin diunduh. Jika *user* memilih untuk mengunduh *file DMA*, sistem akan menampilkan opsi format *download*, yaitu *Excel* atau *PDF*. Apabila jumlah data yang diunduh lebih dari satu, hasil *download* akan dikompresi menjadi *file zip* yang berisi beberapa *file* di dalamnya. Namun, jika hanya satu data yang dipilih, sistem langsung menghasilkan *file* tunggal sesuai format yang diminta. Dengan mekanisme ini, *user* dapat memperoleh data dengan cepat tanpa perlu melakukan pengolahan manual. Proses ini mendukung efisiensi serta meningkatkan keakuratan pengelolaan dokumen transaksi.

E.7 Flowchart Insti S-Invest

Pada Gambar 3.11 memperlihatkan alur menu *Insti S-Invest* yang merupakan salah satu fitur penting dalam sistem *Trade Settlement* yang digunakan untuk mengelola dan mengekspor data transaksi dari institusi *S-Invest*. Fitur ini dirancang untuk memastikan bahwa seluruh data transaksi dapat diakses, diproses, dan diunduh dengan format yang sesuai standar pelaporan. Melalui menu ini, *user* dapat memilih data transaksi yang diperlukan, kemudian mengekspornya ke dalam *file* teks untuk digunakan dalam proses integrasi sistem atau kebutuhan analisis lebih lanjut.

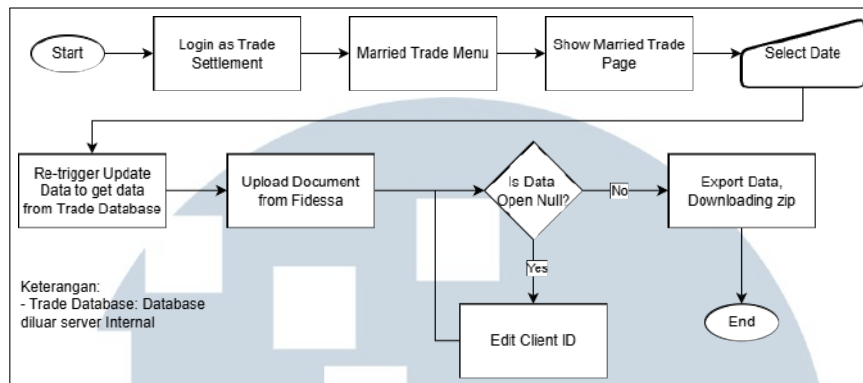


Gambar 3.11. Flowchart Insti S-Invest

Dalam proses ini, *user* dapat mengunduh data transaksi dalam format *file* teks. Sistem menyediakan dua opsi, yaitu menggabungkan seluruh data dalam satu *file* (*download TDBlock*) ataupun memisahkannya ke dalam *file* terpisah untuk setiap data (*download TC*). Fitur ini memberikan fleksibilitas sesuai dengan kebutuhan pengolahan data yang berbeda. Dengan begitu, informasi yang diunduh tetap terstruktur dan siap digunakan untuk kepentingan analisis maupun pelaporan.

E.8 Flowchart Married Trade

Kemudian Gambar 3.12 menunjukkan alur proses pada menu *Married Trade* yang berfungsi untuk melakukan proses *matching* data hasil transaksi dari *Trade Database* dan dari dokumen *Fidessa*. Melalui menu ini, *user* dapat memastikan bahwa setiap transaksi perdagangan telah tercatat dengan benar dan selaras antara dua sumber data tersebut. Fitur ini juga mendukung pengelolaan data yang fleksibel, termasuk kemampuan memperbarui, memperbaiki, dan mengekspor hasil akhir untuk keperluan pelaporan maupun audit. Dengan adanya mekanisme ini, sistem dapat menjaga konsistensi data sekaligus meningkatkan efisiensi proses penyelesaian transaksi.

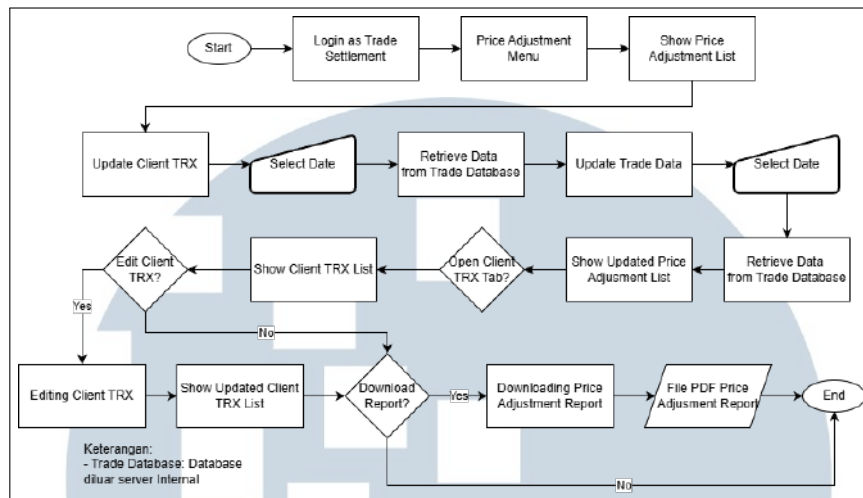


Gambar 3.12. Flowchart Married Trade

Dalam prosesnya, sistem dapat melakukan *re-trigger update data* untuk mengambil data terbaru dari *Trade Database* serta mengunggah dokumen dari *Fidessa* sebagai bagian dari integrasi antar sistem. Proses *matching data* belum dianggap berhasil jika ditemukan data yang belum lengkap atau *open null*, *user* dapat memperbaikinya dengan mengedit *Client ID*. Setelah semua data diperbarui, sistem juga menyediakan fitur ekspor dalam format *zip* dengan beberapa *filter* yang dapat digunakan agar *user* dapat menyimpan hasilnya dengan mudah sesuai dengan yang dibutuhkan. Dengan alur ini, proses pengelolaan *Married Trade* menjadi lebih efisien, terstruktur, dan mengurangi risiko kesalahan data.

E.9 Flowchart Price Adjustment

Pada Gambar 3.13 menggambarkan alur proses pada menu *Price Adjustment* yang berfungsi untuk mengelola proses pembaruan harga transaksi dan memastikan hasil penyesuaian harga. Melalui menu ini, *user* dapat melakukan sinkronisasi data dengan *Trade Database* untuk memastikan setiap transaksi memiliki nilai harga yang akurat dan terkini. Fitur ini juga mendukung proses pembaruan data baik pada sisi *client TRX* maupun *trade data*. Selain itu, sistem dilengkapi dengan kemampuan menghasilkan laporan otomatis untuk keperluan audit dan verifikasi harga.

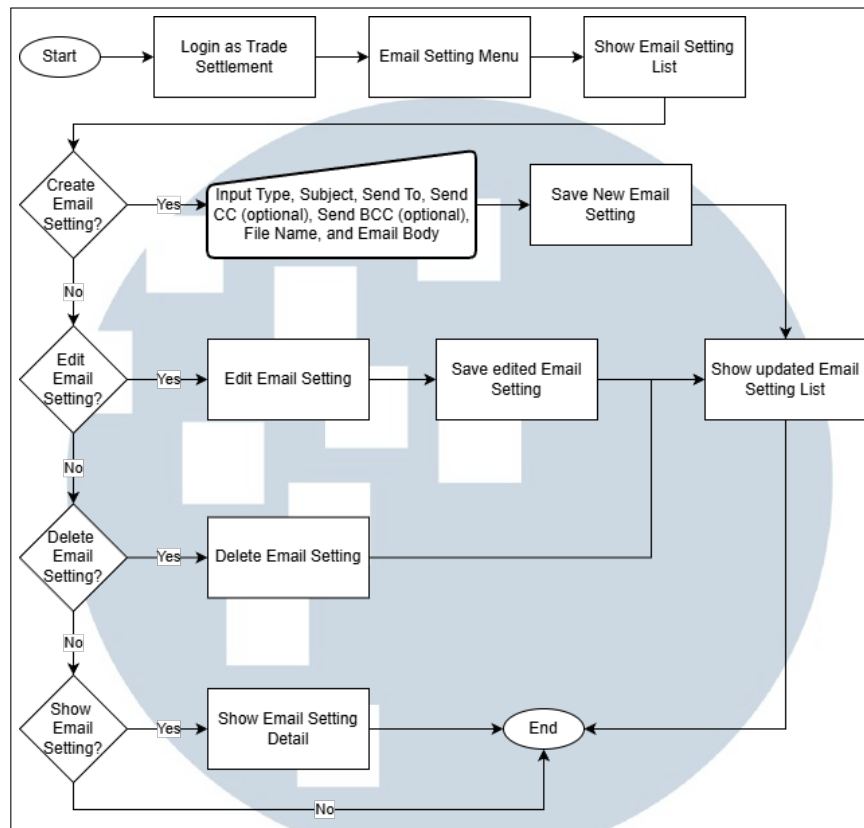


Gambar 3.13. Flowchart Price Adjustment

Menu ini dapat mengambil data dari *Trade Database* sesuai dengan tanggal yang dipilih, sehingga *user* dapat mengakses data terbaru maupun data transaksi sebelumnya. Setelah data diperbarui, sistem menampilkan daftar harga yang telah disesuaikan dan memungkinkan *user* untuk membuka tab *Client TRX* guna melakukan pengeditan jika diperlukan. Selain melakukan pembaruan, sistem juga menyediakan fitur untuk mengunduh laporan *Price Adjustment* dalam format PDF yang berfungsi sebagai dokumen pendukung administrasi dan alat *monitoring* internal. Dengan mekanisme ini, proses penyesuaian harga dapat dilakukan secara lebih cepat, akurat, dan terdokumentasi dengan baik, sehingga meningkatkan keandalan data dalam seluruh aktivitas *Trade Settlement*.

E.10 Flowchart Email Setting

Selanjutnya, Gambar 3.14 menunjukkan alur proses pada menu *Email Setting* yang berfungsi untuk mengatur konfigurasi pengiriman *email* otomatis. Melalui menu ini, *user* dapat menambahkan pengaturan *email* baru dengan menentukan berbagai parameter. Fitur ini juga memungkinkan *user* untuk menentukan nama *file* lampiran dan isi pesan yang akan dikirimkan.



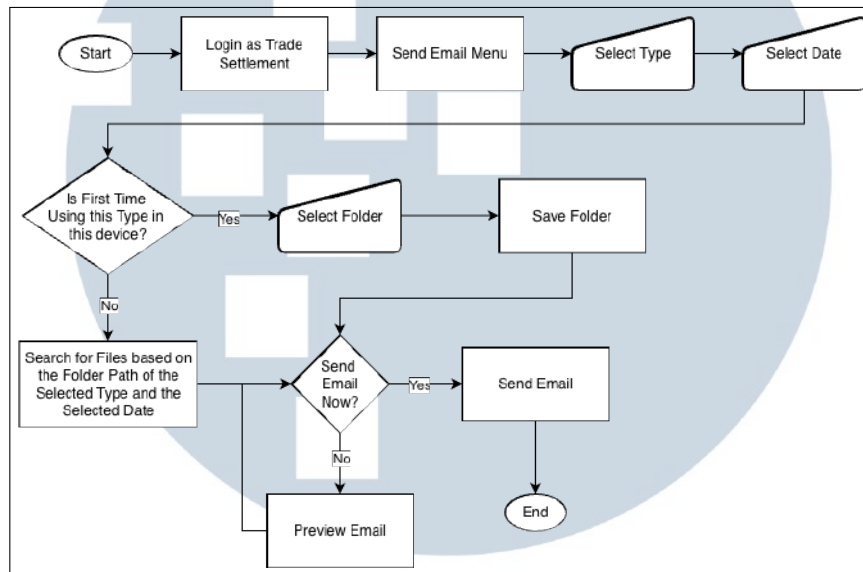
Gambar 3.14. Flowchart Email Setting

Berdasarkan Gambar 3.14, *user* dapat membuat pengaturan baru dengan mengisi detail seperti tipe *email*, subjek, penerima, serta opsi CC dan BCC jika diperlukan. Selain itu, *user* juga dapat menentukan *file name* dan isi pesan sebelum menyimpannya. Sistem juga menyediakan fitur untuk mengedit maupun menghapus pengaturan *email* yang sudah ada agar konfigurasi dapat diperbarui sesuai kebutuhan. Jika diperlukan, *user* dapat menampilkan detail pengaturan *email* untuk memastikan isinya sudah sesuai. Dengan adanya alur ini, proses pengaturan dan pengelolaan sistem pengiriman *email* menjadi lebih fleksibel, efisien, dan mudah dikontrol oleh *user*. Dengan mekanisme pengaturan ini, sistem dapat mengirimkan *email* secara otomatis berdasarkan konfigurasi yang telah disimpan, sehingga mempermudah proses notifikasi, pelaporan, dan komunikasi internal.

E.11 Flowchart Send Email

Pada Gambar 3.15 menunjukkan alur proses pada menu *Send Email* yang berfungsi untuk mengotomatisasi pengiriman *email* berdasarkan tipe dan tanggal yang dipilih secara efisien. Melalui menu ini, *user* dapat mengirimkan laporan

atau dokumen tertentu berdasarkan tipe dan tanggal yang dipilih tanpa harus melakukan proses manual. Fitur ini sangat membantu dalam memastikan bahwa komunikasi rutin, seperti pengiriman laporan harian dapat berjalan tepat waktu dan konsisten. Selain itu, sistem juga dilengkapi dengan fungsi penyimpanan lokasi folder, sehingga *user* tidak perlu menentukan ulang jalur berkas setiap kali melakukan pengiriman.



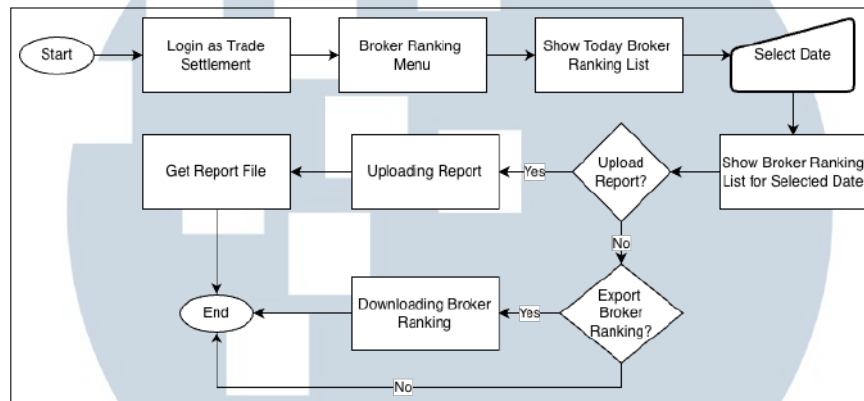
Gambar 3.15. Flowchart Send Email

Proses dimulai ketika *user* memilih tipe dan tanggal pengiriman yang diinginkan. Jika *user* pertama kali menggunakan tipe tertentu di perangkat tersebut, sistem akan meminta pemilihan *folder* agar dapat menentukan lokasi berkas yang akan dikirim. Setelah *folder* tersimpan, sistem akan melakukan pencarian berkas secara otomatis berdasarkan tipe dan tanggal yang dipilih. Setelah *file* ditemukan, *user* dapat memilih untuk langsung mengirim *email* atau melakukan *preview* terlebih dahulu untuk memastikan isi pesan sudah sesuai. Proses ini mempermudah pengiriman *email* rutin karena sistem dapat mengenali *folder* dan berkas yang relevan tanpa perlu pengaturan ulang. Dengan demikian, alur *Send Email* meningkatkan efisiensi kerja dan mengurangi risiko kesalahan dalam proses pengiriman dokumen.

E.12 Flowchart Broker Ranking

Selanjutnya, Gambar 3.16 menunjukkan alur proses pada menu *Broker Ranking* yang berfungsi untuk mengelola laporan peringkat *broker* berdasarkan

tanggal tertentu. Fitur ini membantu *user* dalam memantau dan memperbarui data peringkat *broker* berdasarkan tanggal tertentu. Melalui menu ini, *user* dapat mengunggah laporan baru yang diterima dari sumber eksternal atau mengekspor laporan peringkat yang sudah tersedia di sistem untuk kebutuhan pelaporan harian maupun arsip internal.

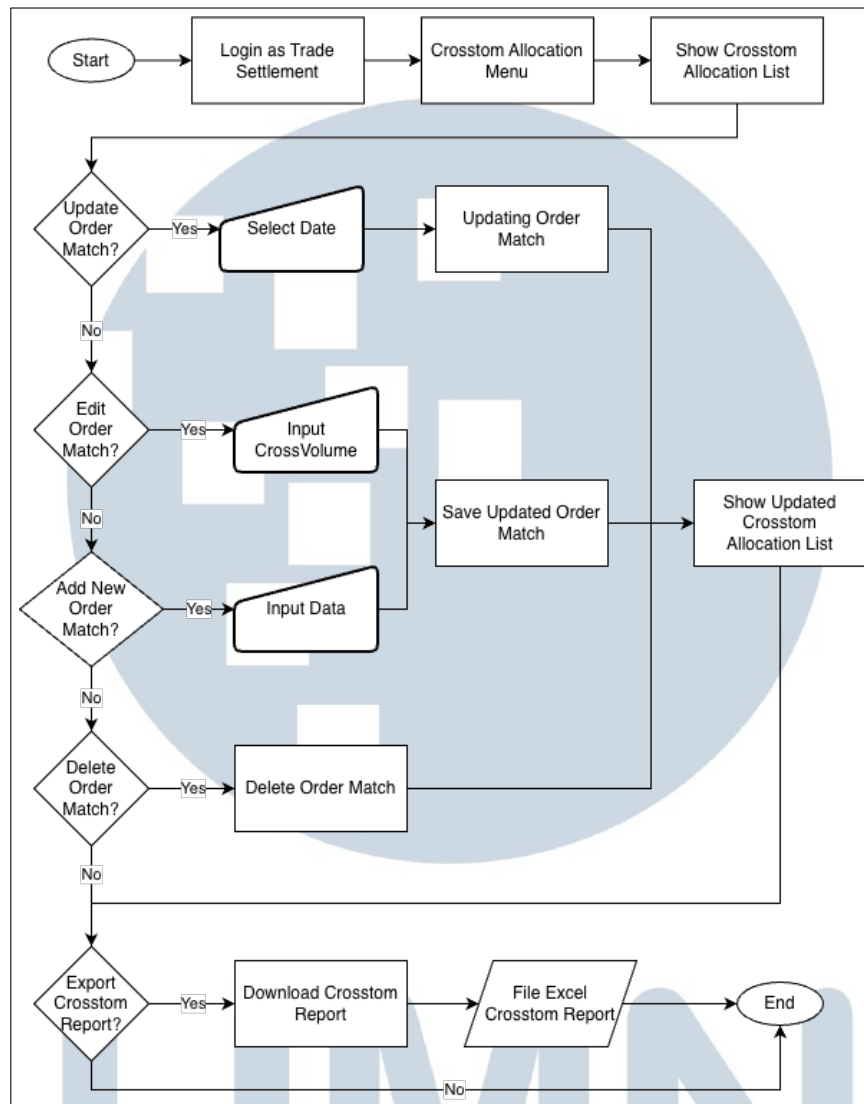


Gambar 3.16. Flowchart Broker Ranking

Dalam alur prosesnya, *user* dapat memilih tanggal laporan yang ingin ditampilkan dan melihat daftar peringkat broker pada hari tersebut. Selanjutnya, *user* dapat memilih untuk mengunggah laporan untuk mendapatkan berkas laporan yang akan dikirimkan ke institusi di Singapura. Selain itu, sistem juga menyediakan opsi untuk mengekspor data *Broker Ranking* dalam bentuk *file* agar dapat digunakan untuk keperluan pelaporan atau arsip internal. Melalui alur ini, proses pengelolaan laporan *Broker Ranking* menjadi lebih efisien dan terstruktur, sekaligus memastikan setiap data yang dikirim maupun diekspor tetap akurat dan sesuai dengan kebutuhan operasional perusahaan.

E.13 Flowchart Crosstom Allocation

Gambar 3.17 menampilkan alur proses pada menu *Crosstom Allocation* yang berfungsi untuk mengelola data *order match* secara menyeluruh. Melalui menu ini, *user* dapat melakukan *CRUD* pada *order match* agar tetap sesuai dengan kondisi aktual di pasar. Fitur ini juga dilengkapi dengan fungsi pembaruan nilai *cross volume* untuk memastikan setiap transaksi tercatat dengan benar sesuai hasil penyelesaian perdagangan.

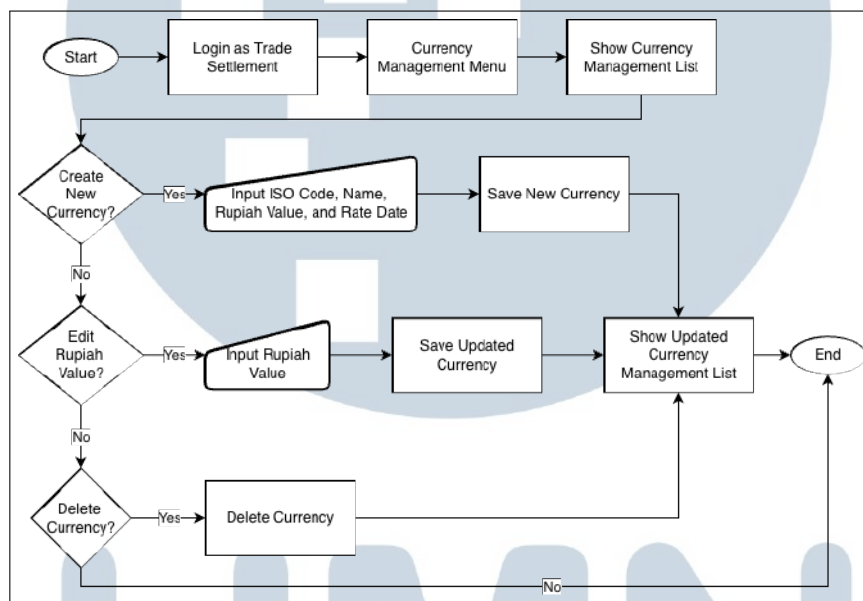


Gambar 3.17. Flowchart Crosstom Allocation

Dalam menu ini, setelah daftar data *order match* yang sudah ada ditampilkan, *user* dapat melakukan *CRUD* pada *order match* sesuai kebutuhan operasional. Proses pembaruan dan penginputan data dilakukan dengan memasukkan nilai *cross volume* atau data transaksi baru agar sistem selalu memiliki informasi terkini. Hal tersebut dilakukan jika hasil otomatisasi dari sistem masih terdapat kesalahan. Selain pengelolaan data, sistem juga menyediakan fitur untuk mengekspor laporan *Crosstom Report* dalam format *Excel*. Fitur ini memudahkan *user* untuk melakukan analisis dan dokumentasi hasil alokasi secara lebih efisien. Dengan adanya alur ini, pengelolaan *Crosstom Allocation* menjadi lebih terstruktur, akurat, dan mendukung transparansi dalam proses transaksi.

E.14 Flowchart Currency Management

Melalui Gambar 3.18 ditunjukkan alur proses pada menu *Currency Management* yang berfungsi untuk mengatur data nilai tukar mata uang yang digunakan dalam sistem. Fitur ini memungkinkan penambahan, pembaruan, serta penghapusan data mata uang agar daftar kurs tetap relevan dan konsisten dengan kebijakan internal perusahaan. Dengan adanya menu ini, proses administrasi dan pengelolaan kurs menjadi lebih efisien, terstandarisasi, dan meminimalkan potensi kesalahan dalam perhitungan nilai transaksi lintas mata uang.



Gambar 3.18. Flowchart Currency Management

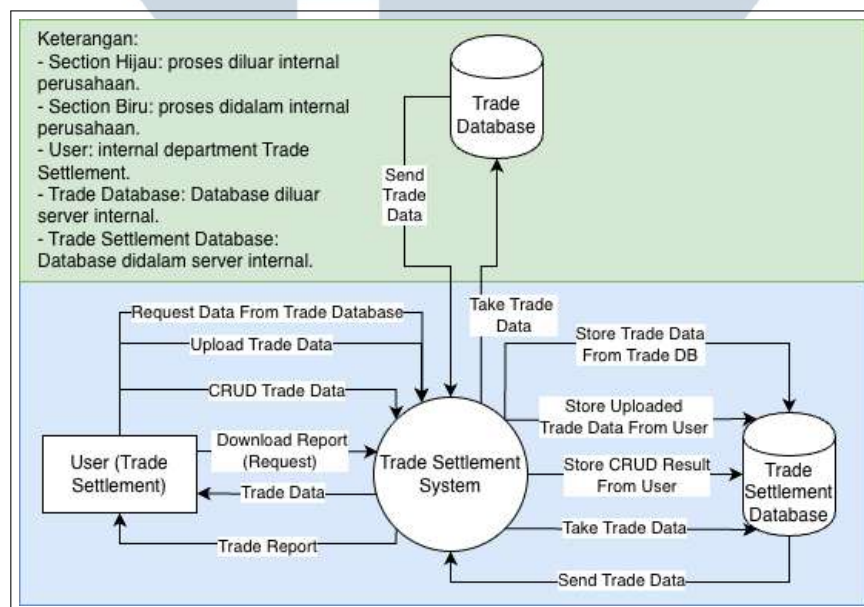
Melalui alur ini, departemen *trade settlement* dapat menambahkan data mata uang baru dengan mengisi informasi seperti *ISO Code* (kode mata uang), nama mata uang, nilai rupiah, dan tanggal kurs. Selain itu, sistem juga memungkinkan *user* memperbarui nilai rupiah jika terjadi perubahan kurs agar data tetap sesuai dengan kondisi pasar. Pada halaman ini juga terdapat fitur untuk menghapus data mata uang yang sudah tidak digunakan agar daftar kurs tetap relevan dan terkelola dengan baik. Dengan adanya alur ini, proses pengelolaan nilai tukar menjadi lebih efisien, terstruktur, dan membantu memastikan ketepatan perhitungan dalam aktivitas transaksi keuangan perusahaan.

F Data Flow Diagram

Pada bagian ini akan membahas *Data Flow Diagram* (DFD) yang digunakan untuk menggambarkan aliran data yang terjadi pada sistem serta interaksi antar entitas yang terlibat. DFD memberikan representasi visual mengenai bagaimana data berpindah, disimpan, dan diproses sehingga mempermudah pemahaman alur kerja sistem yang sedang dikembangkan.

F.1 DFD Level 0

Gambar 3.19 menunjukkan sistem *Trade Settlement* yang digambarkan sebagai satu proses utama yang berperan mengatur seluruh aliran data antara *user*, basis data internal, serta basis data eksternal. Terdapat tiga entitas yang berinteraksi langsung dengan sistem, yaitu *User (Trade Settlement)*, *Trade Database*, dan *Trade Settlement Database*.



Gambar 3.19. DFD Level 0

Bagian berwarna biru pada gambar menggambarkan proses yang terjadi di dalam lingkungan internal perusahaan, sedangkan bagian hijau menunjukkan proses yang berlangsung di luar server internal, yaitu pada *Trade Database*.

Secara umum, alur data berjalan sebagai berikut:

1. *User (Trade Settlement)* melakukan permintaan data ke sistem, seperti mengambil data dari *Trade Database*, mengunggah data transaksi, atau

menjalankan operasi *CRUD* (Create, Read, Update, Delete) terhadap data perdagangan.

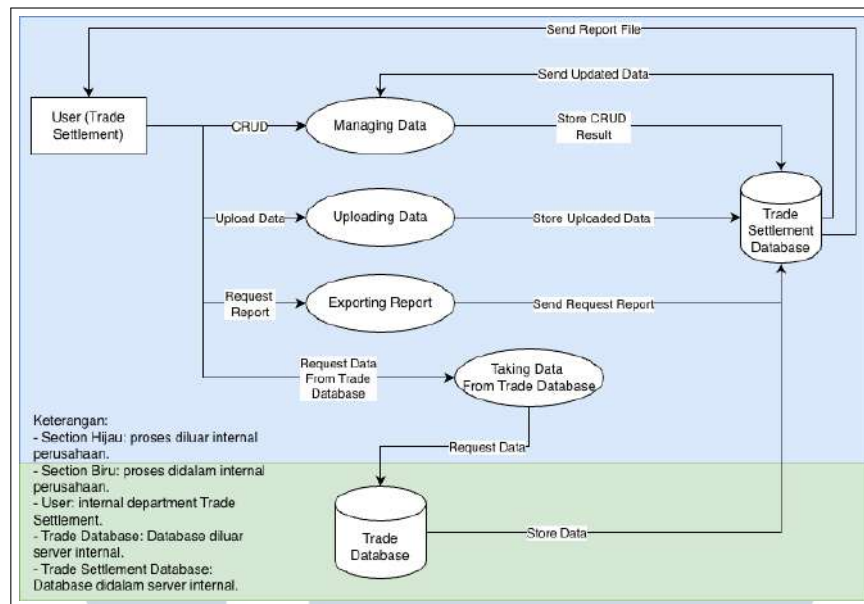
2. Sistem kemudian melakukan pemrosesan terhadap permintaan tersebut dan menyimpannya ke dalam *Trade Settlement Database*.
3. Selain berinteraksi dengan *user*, sistem juga melakukan pertukaran data secara otomatis dengan *Trade Database*. Proses ini mencakup pengambilan data perdagangan dari *Trade Database* (*Take Trade Data*) dan pengiriman data dari *Trade Database* (*Send Trade Data*).
4. Hasil dari seluruh proses tersebut kemudian disajikan kembali kepada *user* dalam bentuk laporan yang dapat diunduh melalui antarmuka aplikasi (*Trade Report*).

Dengan demikian, sistem *Trade Settlement* berfungsi sebagai pusat integrasi antara *user* internal dan dua sumber data utama (internal dan eksternal). Alur ini memastikan setiap data transaksi dapat diproses secara konsisten, disimpan dengan aman di dalam server internal, serta tetap terhubung dengan sistem eksternal untuk sinkronisasi data perdagangan.

F.2 DFD Level 1

Pada Gambar 3.20, aliran data dijabarkan lebih rinci dibandingkan DFD Level 0. Proses utama pada sistem dipecah menjadi empat subproses inti, yaitu *Managing Data*, *Uploading Data*, *Exporting Report*, dan *Taking Data from Trade Database*. Masing-masing subproses saling terhubung dengan *User (Trade Settlement)*, *Trade Settlement Database*, dan *Trade Database*.

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



Gambar 3.20. DFD Level 1

Alur data yang terbentuk pada DFD Level 1 dapat dijelaskan sebagai berikut:

1. Managing Data: *User* melakukan operasi *CRUD* (*Create, Read, Update, Delete*) terhadap data transaksi. Sistem kemudian menyimpan hasil perubahan tersebut sebagai *CRUD Result* di *Trade Settlement Database*. Setelah diperbarui, sistem juga dapat mengirimkan *Updated Data* dan *Report File* kembali ke *user*.
2. Uploading Data: Pada tahap ini, *User* mengunggah data transaksi ke sistem. Data yang diterima akan disimpan di *Trade Settlement Database* sebagai *Uploaded Data* untuk keperluan pemrosesan lebih lanjut.
3. Taking Data from Trade Database: Ketika *User* melakukan permintaan data dari *Trade Database*, sistem akan mengirimkan *Request Data* ke *Trade Database*. Setelah data diterima, sistem menyimpannya sebagai *Store Data* di *Trade Settlement Database* agar dapat digunakan pada proses lain seperti pengelolaan data dan pembuatan laporan.
4. Exporting Report: *User* mengirimkan permintaan laporan kepada sistem. Sistem kemudian mengambil data yang relevan dari *Trade Settlement Database*, mengolahnya, dan menghasilkan *Report File* yang dikirim kembali ke *User*.

Secara keseluruhan, DFD Level 1 pada sistem *Trade Settlement* memperlihatkan bagaimana setiap subprocess saling berkoordinasi untuk menjaga aliran data berjalan dengan lancar. Dengan pembagian proses ini, sistem mampu memastikan setiap data transaksi dikelola, disimpan, dan disajikan secara efisien baik dari sumber internal maupun eksternal.

G Database Schema

Sebelum sistem dikembangkan, tahap perancangan basis data menjadi aspek penting untuk memastikan setiap modul dapat saling terhubung dan menyimpan data secara efisien. Perancangan ini bertujuan agar proses transaksi dan pengelolaan informasi dapat berjalan dengan konsisten serta mudah dipelihara. Pada Gambar 3.21 menunjukkan struktur *database* dari sistem *Trade Settlement*. Struktur ini dirancang untuk mendukung pengelolaan transaksi perdagangan, alokasi data, serta administrasi yang saling terintegrasi.



Proses ini dimulai dari tabel *daily trade* yang menyimpan transaksi harian yang sudah melewati proses *filtering*. Data ini didapatkan dari *trade database* yang berada di luar server internal. Selain itu, data yang diperoleh dari *trade database* juga disimpan dalam tabel *history daily trade*, yang berfungsi sebagai arsip untuk pelacakan, koreksi, maupun perbandingan data transaksi di periode berikutnya.

Selanjutnya, pada tabel *client* digunakan untuk menyimpan daftar nasabah yang didapatkan dari *trade database*. Tabel *client trx* digunakan untuk mencatat nasabah yang melakukan transaksi pada hari tersebut, sehingga sistem dapat memisahkan antara nasabah yang aktif dan tidak pada periode tertentu, data ini ditampilkan pada menu *Trade List*. Kemudian, pada tabel *client id collection* berfungsi untuk memetakan antara data nasabah yang digunakan untuk keperluan *reporting* pada bagian *Contracting* dan *Contracted*. Tabel ini juga memiliki relasi *many-to-many* dengan tabel *fee config*, yang memungkinkan setiap nasabah memiliki konfigurasi biaya transaksi yang berbeda sesuai dengan ketentuan atau kebijakan yang berlaku. Dengan struktur ini, sistem dapat menghasilkan laporan yang lebih akurat dan fleksibel dalam pengelolaan data nasabah serta perhitungan biaya transaksi.

Kemudian, tabel *trade allocation* digunakan untuk menyimpan hasil unggahan data dari proses *upload DMA* dan *CARE*, di mana data yang tersimpan kemudian ditampilkan pada menu *Data Order*, *Insti MSSG*, dan *Insti S-Invest*. Proses ini memastikan seluruh data perdagangan yang diunggah dapat langsung diakses, diverifikasi, dan diolah sesuai kebutuhan operasional. Selanjutnya, pada tabel *mm trade* dan *mm trade history* digunakan untuk menyimpan data hasil unggahan *file CARE* yang dilakukan melalui halaman *Data Order*. *File* ini berisi data aktivitas perdagangan *Market Making* yang mencakup informasi order, harga, volume, dan detail eksekusi. Tabel *mm trade* berperan sebagai penyimpanan utama untuk data aktif yang digunakan dalam proses analisis dan pelaporan, sedangkan *mm trade history* menyimpan riwayat perubahan atau pembaruan data untuk tujuan audit dan validasi sistem. Dengan mekanisme ini, sistem dapat menjaga integritas serta transparansi terhadap seluruh aktivitas perdagangan yang melibatkan *Market Maker*.

Sementara itu, tabel *dma* berfungsi untuk menampung hasil unggahan *file DMA* (*Direct Market Access*) dari halaman *Data Order*. Data yang tersimpan pada tabel ini meliputi informasi transaksi seperti kode saham, harga, volume, nilai tukar, serta identitas nasabah yang melakukan perdagangan melalui jalur *DMA*. Proses unggahan ini memungkinkan sinkronisasi antara data eksternal dan sistem

internal agar seluruh transaksi yang dilakukan secara langsung tetap tercatat dalam sistem *Trade Settlement*. Kemudian untuk tabel *trade eod* dan *trade eod history* berfungsi untuk menyimpan data hasil proses *End of Day (EOD)*, yaitu rekapitulasi perdagangan yang dilakukan di akhir hari. Kedua tabel ini mendapatkan datanya dari hasil unggahan (*upload file*) yang dilakukan pada halaman *Married Trade*. Setelah *file* diunggah, sistem akan memproses dan menyalin informasi ke dalam tabel-tabel tersebut untuk memastikan seluruh transaksi harian tercatat secara konsisten dan dapat dilacak kembali apabila terjadi koreksi data. Tabel historis digunakan untuk menyimpan versi sebelumnya, sehingga sistem memiliki catatan lengkap dari setiap pembaruan yang terjadi.

Selanjutnya, tabel *contract trade* berfungsi untuk menyimpan data transaksi yang tidak *match* setelah proses *matching* selesai dilakukan, sehingga sistem dapat memisahkan transaksi yang masih perlu diperiksa agar tidak tercampur dengan transaksi yang telah *settled*. Tabel ini membantu menjaga integritas data dan memudahkan proses koreksi apabila ditemukan perbedaan antara catatan dan hasil aktual. Kemudian, tabel *email setting* digunakan untuk mengonfigurasi pengiriman email otomatis dalam sistem. Setiap entri berisi informasi tentang penerima (*send to*), salinan (CC dan BCC), subjek, isi pesan (*email body*), dan jenis laporan yang dikirim (*reporting type*). Tabel ini juga menyimpan nama *file* laporan (*reporting file name*) dan folder tempat *file* tersebut disimpan (*file folder name*). Selain itu, kolom status menunjukkan apakah pengaturan email aktif atau tidak. Dengan tabel ini, sistem dapat menjalankan proses pengiriman laporan secara otomatis dan terjadwal sesuai dengan konfigurasi yang disimpan oleh *user*.

Sementara itu, tabel *file history* digunakan untuk mencatat seluruh aktivitas unggahan dan unduhan berkas dalam sistem. Informasi yang disimpan meliputi nama *file* (*file name*), waktu unggah (*upload date*), jenis *file* (*file type*), serta identitas *user* yang melakukan aktivitas tersebut (*uploaded by*). Tabel ini juga menyimpan status proses *file* untuk memastikan setiap unggahan berjalan dengan benar. Dengan adanya tabel ini, sistem memiliki *audit trail* yang lengkap terhadap seluruh pergerakan *file*, sehingga memudahkan proses pelacakan, validasi, dan pengawasan aktivitas *user* di dalam sistem. Adapun tabel *monthly trade* digunakan untuk menyimpan laporan perdagangan bulanan yang menjadi dasar perhitungan dan pelaporan resmi kepada pihak institusi di Singapura. Data yang tersimpan di dalamnya merupakan hasil rekapitulasi dari transaksi harian yang telah disetujui dan diselesaikan, sehingga sistem dapat menyediakan laporan yang konsisten, akurat, dan sesuai dengan standar pelaporan lintas negara.

Pada tabel *crosstom* digunakan untuk menyimpan data transaksi yang terkait dengan aktivitas *cross trade* dan *tomorrow trade*. Struktur tabel ini mengandung informasi penting seperti *client id*, *stock id*, dan arah transaksi (*buy/sell*). Selain itu, terdapat atribut seperti *order price*, *order volume*, *done volume*, dan *cross volume*, yang berfungsi untuk mencatat rincian jumlah dan harga transaksi yang dilakukan antara pihak-pihak terkait. Tabel ini juga menyimpan data *trade date*, yang digunakan sebagai acuan waktu transaksi. Melalui tabel ini, sistem dapat memantau dan mengelola transaksi *cross* secara akurat, memastikan keseimbangan antara volume beli dan jual, serta memfasilitasi proses verifikasi alokasi transaksi.

Tabel *currency* berfungsi untuk menyimpan data nilai tukar mata uang yang digunakan dalam setiap transaksi lintas mata uang di dalam sistem. Kolom utama seperti *iso code*, *name*, *rate date*, dan *rupiah value* digunakan untuk mengidentifikasi jenis mata uang, tanggal kurs, serta nilai tukar terhadap rupiah. Tabel ini memiliki hubungan *many-to-many* dengan tabel *trade allocation*, karena satu transaksi dapat melibatkan lebih dari satu jenis mata uang, sementara satu mata uang juga dapat digunakan dalam banyak transaksi berbeda. Melalui relasi ini, sistem mampu melakukan konversi nilai secara dinamis dan menjaga akurasi perhitungan finansial antar transaksi lintas mata uang.

Tabel *right* digunakan untuk mencatat aktivitas transaksi yang berkaitan dengan pembagian hak saham atau *right issue* kepada investor. Setiap entri di tabel ini memuat informasi penting seperti identitas nasabah (*client id*), kode saham (*stock id*), pasar perdagangan (*market id*), harga transaksi (*trade price*), serta volume dan jumlah perdagangan (*trade volume*, *trade amount*). Selain itu, tabel ini juga mencatat status transaksi seperti *settled*, *upload*, dan *netting* untuk menunjukkan apakah proses perdagangan telah diselesaikan atau masih dalam tahap pemrosesan. Melalui tabel ini, sistem dapat memantau seluruh kegiatan pembagian hak saham dan memastikan bahwa setiap hak yang dimiliki investor tercatat dengan akurat dan transparan.

Sementara itu, tabel *stock positions* digunakan untuk menyimpan informasi posisi saham yang dimiliki setiap nasabah pada periode tertentu. Tabel ini mencatat berbagai atribut penting seperti identitas nasabah, kode saham, harga penutupan terakhir (*last close price*), jumlah portofolio, serta nilai transaksi seperti *buy*, *sell*, *repo*, *lend*, dan *borrow*. Selain itu, tabel ini juga memuat informasi tambahan terkait pihak penjual atau perantara seperti *office name* dan *sales person name*. Fungsi utama tabel ini adalah untuk menampilkan posisi portofolio terkini, menghitung nilai total kepemilikan, serta mendukung proses pelaporan posisi saham pada akhir

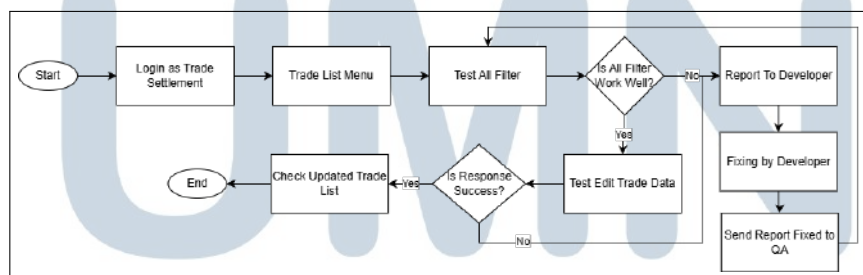
periode. Dengan demikian, tabel *stock positions* menjadi bagian penting dalam menjaga keakuratan data investasi dan performa portofolio nasabah di dalam sistem.

H Flow Testing

Pada bagian ini, dibahas mengenai tahap pengujian untuk memastikan setiap alur proses dalam sistem berjalan sesuai dengan logika bisnis dan kebutuhan *user*. Pengujian ini dilakukan secara manual oleh tim *QA*, yaitu Lidya Silitonga. Melalui tahapan ini, tim *developer* dapat mengetahui apakah sistem telah berfungsi secara integratif dan bebas dari *error* yang mengganggu operasional. Dengan demikian, *testing* menjadi dasar utama dalam menjamin stabilitas sistem sebelum digunakan pada lingkungan *production*.

H.1 Flow Testing Menu Trade List

Menu *Trade List* merupakan salah satu fitur utama dalam sistem yang berfungsi untuk menampilkan seluruh data transaksi perdagangan yang telah terjadi. Menu ini menjadi titik awal bagi *user* untuk melakukan pemantauan, analisis, serta pembaruan terhadap data transaksi harian. Dengan demikian, menu ini menjadi fondasi penting dalam kegiatan operasional sistem perdagangan. Pada Gambar 3.22 diperlihatkan *flow testing* yang dilakukan pada menu ini.



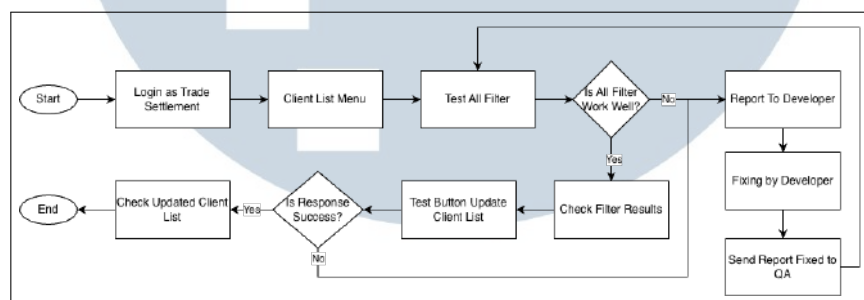
Gambar 3.22. Flow Testing Trade List

Pengujian ini dimulai dengan melakukan pengujian pada fitur *filter* yang terdapat di halaman tersebut. Tujuan dari pengujian ini adalah untuk memastikan bahwa sistem mampu menampilkan data perdagangan sesuai dengan parameter yang dimasukkan oleh *user* tanpa adanya keterlambatan atau kesalahan data. Apabila proses *filter* berjalan dengan baik, pengujian diteruskan dengan melakukan proses *edit* pada data tertentu untuk memverifikasi fungsi pembaruan informasi transaksi dan memastikan bahwa perubahan tersimpan dengan benar di basis data.

Seluruh fitur pada menu ini dianggap telah aman jika tidak ditemukan *error* atau *data mismatch* selama proses pengujian berlangsung.

H.2 Flow Testing Menu Client List

Pada menu *Client List* berfungsi untuk menampilkan daftar nasabah yang terdaftar dalam sistem perdagangan. Halaman ini menyediakan informasi lengkap terkait identitas, status, dan data transaksi nasabah. Fitur ini dirancang untuk membantu *user* internal dalam melakukan pengawasan dan verifikasi terhadap data nasabah secara cepat dan akurat. Selain itu, menu ini juga menjadi dasar bagi proses pengelolaan nasabah yang terintegrasi dengan modul lain seperti alokasi dan pelaporan transaksi. Pada Gambar 3.23 diperlihatkan *flow testing* yang dilakukan pada menu ini.



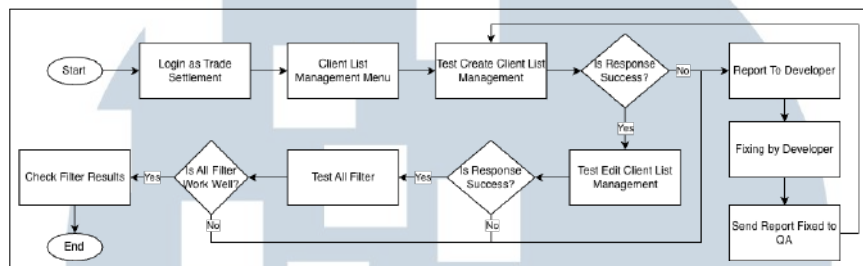
Gambar 3.23. Flow Testing Client List

Pengujian dimulai dengan memastikan fungsi *filter* berdasarkan nama nasabah, kode nasabah, serta tipe nasabah. Tujuan pengujian ini adalah untuk memverifikasi bahwa sistem mampu menampilkan daftar nasabah dengan data yang konsisten dan sesuai dengan hasil *query* dari *database*. Selain itu, dilakukan pula pengujian untuk *update client list* yang bertujuan untuk mendapatkan data nasabah terbaru. Fitur yang terdapat pada menu ini dianggap telah berjalan dengan baik jika tidak ditemukan *error* ataupun *data duplication* selama pengujian berlangsung.

H.3 Flow Testing Menu Client List Management

Kemudian pada menu *Client List Management* berfungsi untuk mengelola data nasabah secara menyeluruh, seperti penambahan dan pembaruan data. Menu ini berperan penting dalam menjaga keakuratan informasi nasabah agar tetap mutakhir dan terintegrasi dengan sistem *backend*. Melalui tampilan yang interaktif,

user dapat dengan mudah memvalidasi atau memperbaiki data yang ditemukan tidak sesuai. Dengan demikian, menu ini menjadi pusat aktivitas administrasi terkait data nasabah. Pada Gambar 3.24 ditunjukkan mengenai alur *testing* yang dilakukan pada menu ini.

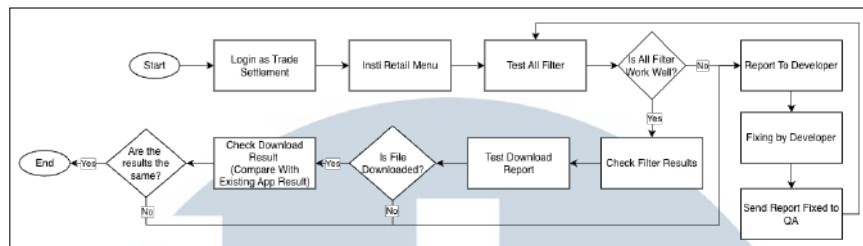


Gambar 3.24. Flow Testing Client List Management

Tahap awal pengujian dilakukan dengan menambahkan data baru dan memverifikasi apakah data tersebut tersimpan dengan benar di *database*. Selanjutnya dilakukan pengujian pembaruan data dengan mengubah salah satu atribut nasabah. Tujuan pengujian ini adalah untuk memastikan bahwa setiap perubahan data dapat tersimpan dengan baik tanpa mengakibatkan inkonsistensi. Kemudian dilanjutkan dengan melakukan pengujian *filter* untuk memastikan fungsi *filter* berjalan dengan baik. Seluruh fitur dianggap telah berhasil melewati fase *testing* jika sudah tidak ditemukan *error* pada fitur yang terkait.

H.4 Flow Testing Menu Insti Retail

Menu *Insti Retail* digunakan untuk menampilkan data transaksi yang dilakukan oleh nasabah ritel pada sistem *Trade Settlement*. Menu ini menjadi salah satu bagian penting dalam memastikan keakuratan data transaksi dan validasi laporan yang dihasilkan. Melalui pengujian pada menu ini, tim *QA* dapat memastikan bahwa seluruh fungsi utama seperti *filter* data dan *download report* dapat berjalan sesuai dengan kebutuhan *user*. Pada Gambar 3.25 ditampilkan alur *flow testing* yang diterapkan untuk memastikan performa dan keandalan fitur yang ada di dalam menu tersebut.



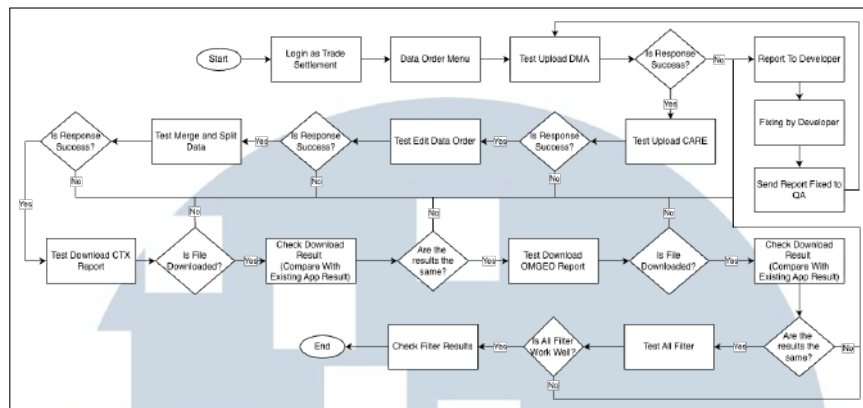
Gambar 3.25. Flow Testing Insti Retail

Berdasarkan alur pada Gambar 3.25, proses pengujian diawali melakukan pengujian terhadap seluruh fungsi *filter* untuk memastikan setiap *filter* menampilkan data yang sesuai. Apabila ditemukan kendala, pengujian akan dilanjutkan dengan pelaporan kepada *developer* untuk dilakukan perbaikan. Setelah semua *filter* berfungsi dengan baik, langkah selanjutnya adalah melakukan uji *download report* dan membandingkan hasil unduhan dengan laporan dari aplikasi sebelumnya. Fitur dianggap telah berfungsi dengan benar apabila hasil laporan yang diunduh memiliki kesesuaian data dengan aplikasi lama dan tidak ditemukan perbedaan pada hasil pengujian.

H.5 Flow Testing Menu Data Order

Pada menu *Data Order* digunakan untuk mengelola proses *upload* dan pemrosesan *file* transaksi yang berasal dari sistem eksternal, seperti *file DMA* dan *CARE*. Menu ini berperan penting dalam memastikan data transaksi yang diterima oleh sistem *Trade Settlement* dapat diolah dan ditampilkan dengan benar. Melalui pengujian ini, dilakukan verifikasi terhadap keandalan proses unggahan, validasi data, serta sinkronisasi hasil pemrosesan dengan data referensi yang sudah ada. Pada Gambar 3.26 ditampilkan alur *flow testing* yang digunakan untuk menguji keseluruhan proses pada menu *Data Order*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

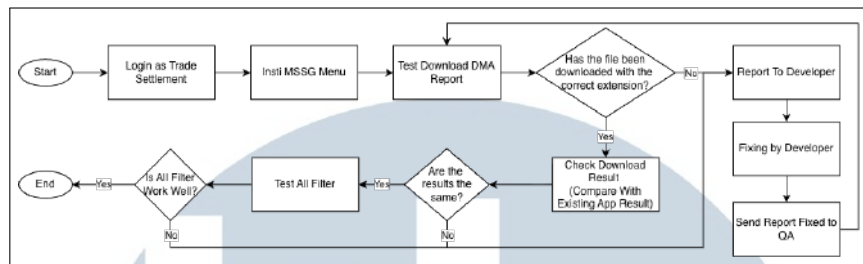


Gambar 3.26. Flow Testing Data Order

Berdasarkan alur pengujian pada Gambar 3.26, tahapan pengujian mencakup proses *upload file DMA* dan *CARE*, pengujian respon sistem, serta pengecekan hasil *upload* apakah berhasil diproses dan tersimpan ke dalam *database*. Selain itu, dilakukan pula pengujian fungsi *merge* dan *split data* untuk memastikan pemisahan serta penggabungan transaksi berjalan dengan baik. Setelah proses *upload* berhasil, dilakukan *download report CTX* dan *OMGEO* untuk dibandingkan dengan hasil dari aplikasi sebelumnya. Menu ini dianggap berfungsi dengan baik apabila seluruh proses *upload*, *filter*, dan hasil *download report* menunjukkan kesesuaian data tanpa adanya perbedaan hasil dari sistem lama.

H.6 Flow Testing Menu Insti MSSG

Pada menu *Insti MSSG* digunakan untuk menampilkan hasil alokasi transaksi institusi yang berasal dari *file upload DMA* dan *CARE*. Pengujian dilakukan untuk memastikan bahwa proses pemuatan data, fungsi *filter*, serta fitur *download report* dapat berjalan dengan benar tanpa kesalahan format *file*. Selain itu, pengujian ini juga bertujuan untuk menilai konsistensi hasil antara sistem yang baru dikembangkan dengan aplikasi sebelumnya agar performanya tetap stabil. Pada Gambar 3.27 ditampilkan alur *flow testing* yang digunakan untuk menguji keseluruhan proses pada menu *Insti MSSG*.

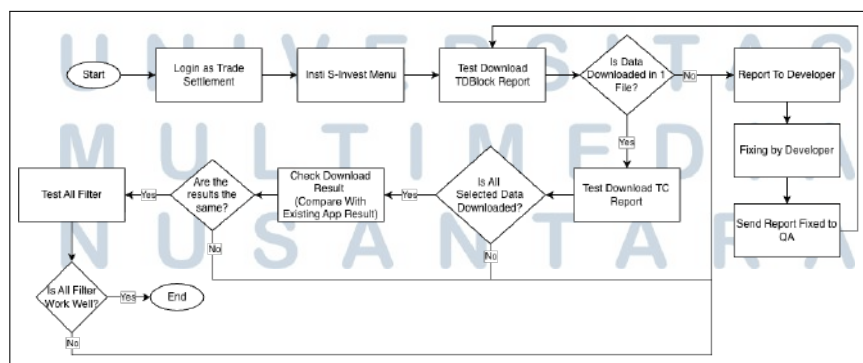


Gambar 3.27. Flow Testing Insti MSSG

Berdasarkan alur pengujian pada Gambar 3.27, proses dimulai dari *login* sebagai *user Trade Settlement*, kemudian membuka menu *Insti MSSG* untuk menguji fitur *download report DMA*. Tim *QA* akan memeriksa apakah *file* yang diunduh memiliki ekstensi yang sesuai. Jika tidak, laporan akan dikirim ke *developer* untuk diperbaiki sebelum dikirim kembali ke *QA*. Setelah *file* terunduh dengan benar, hasil *download* dibandingkan dengan data aplikasi lama. Pengujian dianggap berhasil apabila seluruh hasil data sesuai, fungsi *filter* dapat bekerja dengan baik, dan tidak ditemukan perbedaan antara hasil baru dan lama.

H.7 Flow Testing Menu Insti S-Invest

Selanjutnya, pada menu *Insti S-Invest* berfungsi untuk menampilkan hasil integrasi data antara sistem *Trade Settlement* dan institusi *S-Invest*. Pengujian dilakukan untuk memastikan proses *download report* dan fungsi *filter* berjalan dengan benar serta hasilnya konsisten dengan aplikasi sebelumnya. Selain itu, pengujian ini juga bertujuan untuk memverifikasi bahwa setiap data yang dipilih berhasil diunduh dengan format yang sesuai dan tidak terjadi kehilangan data saat proses berjalan. Pada Gambar 3.28 ditampilkan alur *flow testing* yang digunakan untuk menguji keseluruhan proses pada menu *Insti S-Invest*.

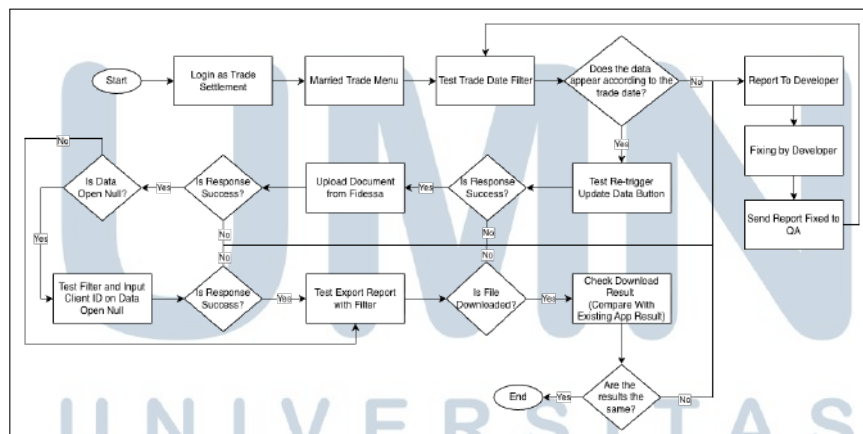


Gambar 3.28. Flow Testing Insti S-Invest

Berdasarkan alur pengujian pada Gambar 3.28, proses dimulai dengan *login* sebagai *user Trade Settlement* dan membuka menu *Insti S-Invest*. Selanjutnya dilakukan pengujian fitur unduh laporan seperti *TD Block* untuk memastikan seluruh data terunduh dalam satu *file txt* dan *TC Report* terunduh secara terpisah setiap data yang dipilih. Jika ditemukan ketidaksesuaian, laporan dikirim ke *developer* untuk diperbaiki sebelum diverifikasi ulang oleh *QA*. Pengujian dianggap berhasil apabila seluruh data terunduh dengan benar, hasil *download* sesuai dengan sistem sebelumnya, dan seluruh *filter* berfungsi tanpa kendala.

H.8 Flow Testing Menu Married Trade

Pada menu *Married Trade* digunakan untuk melakukan proses *matching* data hasil transaksi dari *Trade Database* dan dari dokumen *Fidessa*. Pengujian dilakukan untuk memastikan bahwa sistem dapat membaca data dari *Trade Database* dengan benar, serta mampu memproses hasil pembaruan dari dokumen *Fidessa*. Selain itu, pengujian juga dilakukan untuk menilai kestabilan sistem ketika melakukan ekspor laporan berdasarkan filter tertentu. Pada Gambar 3.29 ditampilkan alur *flow testing* yang digunakan untuk menguji keseluruhan proses pada menu *Married Trade*.



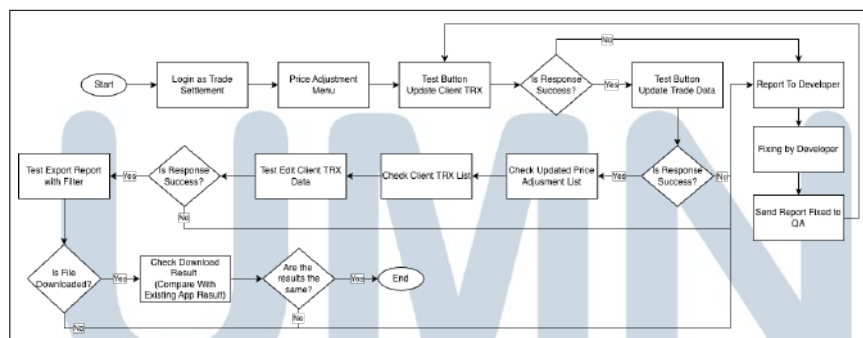
Gambar 3.29. Flow Testing Married Trade

Berdasarkan alur pengujian pada Gambar 3.29, pengujian dilakukan terhadap beberapa tahap, seperti pengujian *filter* tanggal transaksi, *upload* dokumen dari *Fidessa*, pembaruan data menggunakan tombol *Re-trigger Update Data* jika data pada hari tersebut belum ter-*update*, hingga ekspor laporan hasil transaksi. Hasil *matching* data dianggap berhasil ketika tidak ada kasus *open null* pada data.

Jika terdapat kasus *open null* maka *user* harus mengisi *client id* supaya *matching* data dapat selesai. Jika sistem gagal merespons atau hasil *download report* tidak sesuai, laporan dikirim ke *developer* untuk diperbaiki sebelum diverifikasi ulang oleh *QA*. Pengujian dinyatakan berhasil apabila seluruh data tampil sesuai tanggal perdagangan, *file* hasil ekspor terunduh dengan format yang benar, serta hasil data cocok dengan aplikasi pembanding sebelumnya.

H.9 Flow Testing Menu Price Adjustment

Kemudian, pada menu *Price Adjustment* digunakan untuk mengelola proses pembaruan harga transaksi dan memastikan hasil penyesuaian harga tersinkronisasi dengan data yang ada di *Trade Database*. Pengujian dilakukan untuk memverifikasi bahwa sistem dapat melakukan pembaruan data transaksi nasabah maupun data perdagangan secara otomatis, serta menghasilkan laporan penyesuaian harga yang akurat. Selain itu, pengujian juga mencakup validasi terhadap respon sistem, fungsi ekspor laporan, dan kesesuaian hasil unduhan dengan aplikasi sebelumnya. Pada Gambar 3.30 ditampilkan alur *flow testing* yang digunakan untuk menguji keseluruhan proses pada menu *Price Adjustment*.

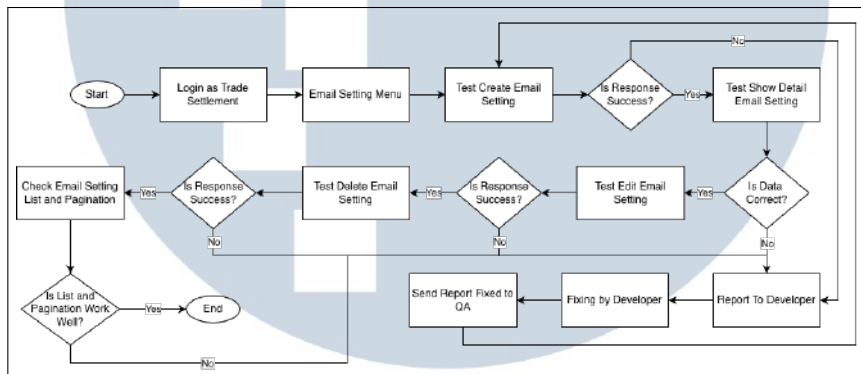


Gambar 3.30. Flow Testing Price Adjustment

Pengujian dilakukan terhadap tombol *Update Client TRX* dan *Update Trade Data* untuk memastikan sistem memberikan respon yang sukses serta memperbarui daftar transaksi dan penyesuaian harga. Selanjutnya dilakukan pengujian *editing data* terhadap *client trx list* dan dilanjutkan dengan pengujian ekspor laporan dengan *filter* tertentu untuk memastikan *file* hasil *download* memiliki isi yang benar dan format yang sesuai. Pengujian dianggap berhasil apabila *file* laporan berhasil diunduh, hasilnya sama dengan data aplikasi sebelumnya, dan seluruh fungsi berjalan tanpa kesalahan sistem.

H.10 Flow Testing Menu Email Setting

Pada menu *Email Setting* berfungsi untuk mengatur konfigurasi pengiriman email otomatis pada sistem *Trade Settlement*. Melalui menu ini, *user* dapat menambah, mengedit, menampilkan detail, maupun menghapus pengaturan *email* yang digunakan untuk notifikasi atau laporan sistem. Pengujian dilakukan untuk memastikan bahwa seluruh fungsi tersebut berjalan dengan baik, serta data yang tersimpan benar-benar digunakan dalam proses pengiriman otomatis. Pada Gambar 3.31 ditampilkan alur *flow testing* yang digunakan untuk menguji keseluruhan proses pada menu *Email Setting*.



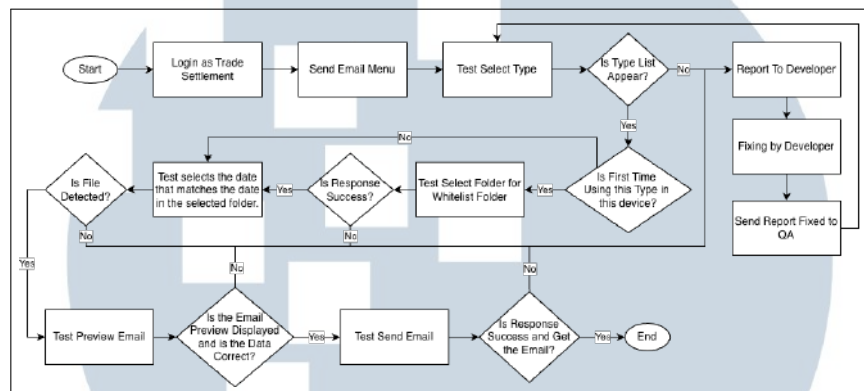
Gambar 3.31. Flow Testing Email Setting

Tahapan pengujian meliputi pembuatan konfigurasi *email* baru, menampilkan detail pengaturan, mengedit, hingga menghapus data *email* yang sudah ada. Setelah itu, sistem diuji untuk memastikan respon yang diberikan berhasil dan data tersimpan dengan benar. Jika ditemukan kesalahan, laporan dikirim ke *developer* untuk diperbaiki sebelum diverifikasi ulang oleh *QA*. Pengujian dianggap berhasil apabila seluruh proses *create*, *read*, *update*, *delete* berjalan dengan lancar, data yang ditampilkan sesuai dengan hasil input, serta daftar pengaturan dan *pagination* berfungsi tanpa kendala.

H.11 Flow Testing Menu Send Email

Menu *Send Email* berfungsi untuk mengotomatisasi proses pengiriman email berdasarkan tipe dan tanggal yang dipilih. Melalui menu ini, sistem secara otomatis mengirimkan *email* beserta lampiran laporan kepada penerima yang telah dikonfigurasi sebelumnya pada menu *Email Setting*. Pengujian dilakukan untuk memastikan bahwa proses pemilihan tipe email, deteksi *file* laporan sesuai tanggal,

serta pengiriman *email* berjalan tanpa kendala. Selain itu, diuji juga apakah sistem dapat menampilkan *preview email* dengan benar sebelum dikirimkan ke penerima. Pada Gambar 3.32 ditampilkan alur *flow testing* yang digunakan untuk menguji keseluruhan proses pada menu *Send Email*.



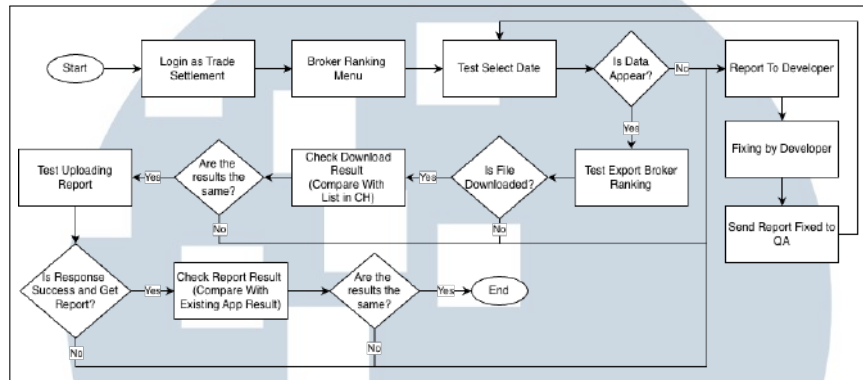
Gambar 3.32. Flow Testing Send Email

Berdasarkan alur pengujian pada Gambar 3.32, proses pengujian dimulai dengan memilih tipe *email* yang ingin dikirim, lalu sistem melakukan pengecekan apakah tipe tersebut pertama kali digunakan pada *device* tersebut atau tidak. Jika "Ya" maka sistem akan meminta *user* untuk memilih folder yang akan digunakan. Jika "Tidak" maka sistem akan otomatis memilih *file* sesuai dengan tipe dan tanggal yang dipilih. Setelah *file* laporan terdeteksi, dilakukan pengujian pada fitur *preview email* untuk memastikan isi pesan dan lampiran sudah benar. Jika semua data valid, sistem kemudian mengirimkan *email* ke daftar penerima yang telah disetujui. Pengujian dianggap berhasil apabila sistem mampu menampilkan pratinjau dengan benar, mengirim email sesuai jadwal, dan penerima menerima pesan beserta lampirannya tanpa kesalahan.

H.12 Flow Testing Menu Broker Ranking

Menu *Broker Ranking* berfungsi untuk mengelola laporan peringkat *broker* berdasarkan tanggal tertentu. Melalui menu ini, *user* dapat mengunggah laporan harian, menampilkan hasil peringkat *broker*, serta mengekspor data tersebut untuk keperluan pelaporan atau analisis lebih lanjut. Pengujian dilakukan untuk memastikan sistem dapat membaca *file* laporan dengan benar, menampilkan data sesuai urutan peringkat, dan menghasilkan *file* ekspor yang sesuai dengan format standar. Selain itu, pengujian juga mencakup validasi terhadap konsistensi hasil data

antara sistem yang diuji dan aplikasi sebelumnya. Pada Gambar 3.33 ditampilkan alur *flow testing* yang digunakan untuk menguji keseluruhan proses pada menu *Broker Ranking*.

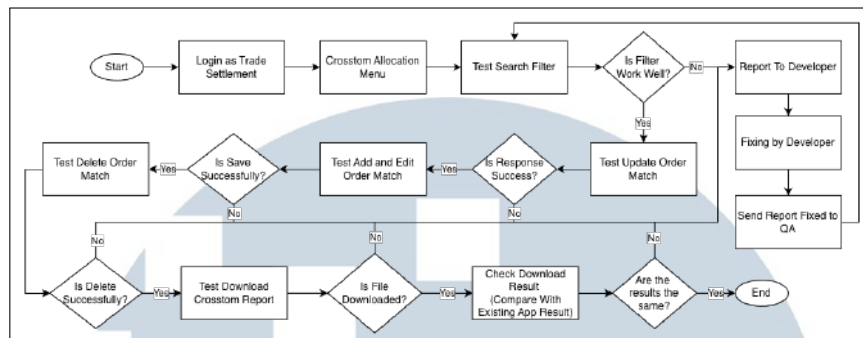


Gambar 3.33. Flow Testing Broker Ranking

Berdasarkan alur pengujian pada Gambar 3.33, Tahapan pengujian mencakup pemilihan tanggal laporan, kemudian dilakukan uji fungsi ekspor *broker ranking*, dan dilakukan perbandingan dengan list yang ada pada halaman *broker ranking*. Selanjutnya, dilakukan juga pengujian *upload file* laporan harian, untuk mendapatkan hasil laporan yang akan diberikan ke institusi di Singapura. Jika ditemukan ketidaksesuaian hasil, laporan dikirim ke *developer* untuk diperbaiki dan diverifikasi ulang oleh *QA*. Pengujian dianggap berhasil apabila sistem dapat mengunggah *file* dengan benar, menampilkan data peringkat sesuai urutan, serta menghasilkan *file* ekspor yang identik dengan hasil dari sistem sebelumnya.

H.13 Flow Testing Menu Crosstom Allocation

Menu *Crosstom Allocation* berfungsi untuk mengelola data *order match* secara menyeluruh, mulai dari proses pencarian, penambahan, pembaruan, hingga penghapusan data *crosstom*. Pengujian dilakukan untuk memastikan bahwa seluruh fungsi tersebut dapat berjalan dengan baik dan menghasilkan data yang konsisten. Selain itu, dilakukan juga pengujian pada fitur ekspor laporan untuk memastikan file yang dihasilkan sesuai dengan data yang tersimpan di sistem. Pada Gambar 3.34 ditampilkan alur *flow testing* yang digunakan untuk menguji keseluruhan proses pada menu *Crosstom Allocation*.

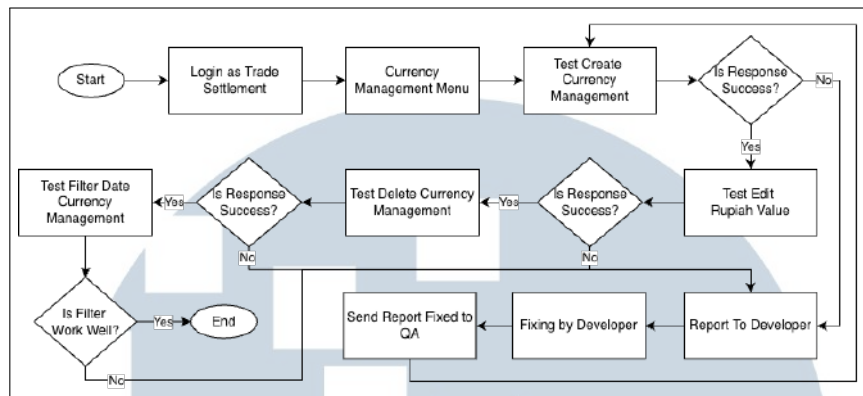


Gambar 3.34. Flow Testing Crosstom Allocation

Pengujian meliputi pencarian data menggunakan *filter*, pengujian fitur *add* dan *edit order match*, pembaruan data, serta penghapusan data. Selanjutnya dilakukan pengujian *download report* untuk memverifikasi apakah *file* hasil *download* telah sesuai dengan data pada sistem dan aplikasi sebelumnya. Pengujian dianggap berhasil apabila seluruh fungsi (*filter*, *CRUD* dan *download report*) berjalan dengan lancar, *file* laporan berhasil diunduh, serta hasil datanya sesuai tanpa adanya kesalahan format atau kehilangan informasi.

H.14 Flow Testing Menu Currency Management

Pada menu *Currency Management* berfungsi untuk mengatur data nilai tukar mata uang yang digunakan dalam sistem *Trade Settlement*. Melalui menu ini, *user* dapat melakukan *CRUD* untuk kurs mata uang yang menjadi acuan dalam perhitungan transaksi. Pengujian dilakukan untuk memastikan setiap fungsi pada menu ini dapat berjalan dengan baik, termasuk validasi terhadap perubahan nilai tukar rupiah dan fitur penyaringan data berdasarkan tanggal. Sebelum pengujian dilakukan, data simulasi kurs diinputkan ke dalam sistem untuk memastikan keakuratan hasil uji. Pada Gambar 3.35 ditampilkan alur *flow testing* yang digunakan untuk menguji keseluruhan proses pada menu *Currency Management*.

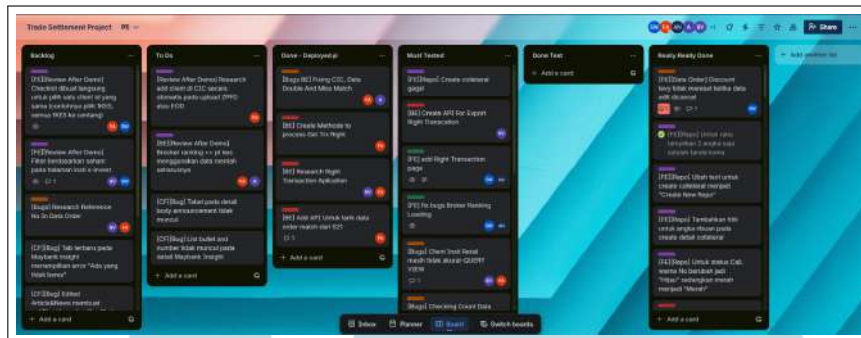


Gambar 3.35. Flow Testing Currency Management

Pengujian dilakukan dengan pengetesan mekanisme *CRUD* pada data kurs mata uang. Selain itu, dilakukan pengujian terhadap *filter* tanggal untuk memastikan data kurs dapat ditampilkan sesuai periode yang dipilih. Jika ditemukan respon gagal atau kesalahan perhitungan, laporan akan dikirim ke *developer* untuk diperbaiki sebelum diverifikasi ulang oleh *QA*. Pengujian dinyatakan berhasil apabila sistem mampu melakukan mekanisme *CRUD* dengan benar, dan seluruh fungsi *filter* berjalan dengan baik.

H.15 Hasil Testing

Trello digunakan sebagai alat bantu untuk memantau perkembangan sistem *Trade Settlement*. Platform ini membantu tim *developer* dalam mengatur tugas, menentukan prioritas, serta memastikan setiap fitur berjalan sesuai dengan jadwal yang telah ditetapkan. Dengan tampilan *board*, *Trello* memungkinkan pembagian tugas secara visual dan terstruktur, sehingga seluruh anggota tim dapat memantau progres pekerjaan dengan mudah. Setiap kolom mewakili tahap tertentu dalam siklus pengembangan, mulai dari perencanaan, pengerjaan, pengujian, hingga penyelesaian fitur. Pada Gambar 3.36, ditunjukkan mengenai *progress development* sistem *trade settlement*.



Gambar 3.36. Hasil Testing

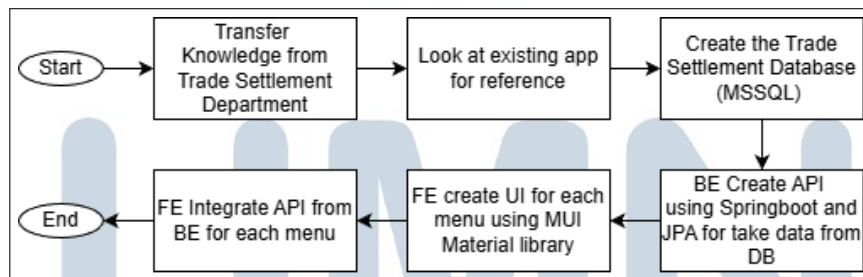
Proses pengembangan dan pengujian dilakukan dengan memantau perpindahan setiap tugas melalui beberapa tahap seperti:

1. **Backlog:** berguna untuk menampilkan daftar perubahan, ide, ataupun perbaikan yang belum dijadwalkan, sehingga tim dapat mengurutkan prioritas dan kompleksitasnya.
2. **To Do:** berguna untuk menampilkan daftar tugas yang sudah ditetapkan untuk dikerjakan pada siklus *development* saat ini. *Task* yang ada di kolom ini siap dikerjakan oleh tim *developer* sesuai pembagian tanggung jawab yang telah ditentukan.
3. **Done-Deployed:** berguna untuk menunjukkan bahwa suatu fitur atau perbaikan telah berhasil dikembangkan dan telah dilakukan *deployment*. Pada tahap ini, fitur telah siap diuji namun belum melalui proses validasi kualitas secara menyeluruh.
4. **Must Tested:** berfungsi untuk menampilkan daftar fitur atau perbaikan yang siap untuk diuji oleh tim *QA* yang berguna untuk memastikan fitur bebas dari *bug*.
5. **Done Test:** berfungsi untuk menampilkan daftar fitur atau perbaikan yang telah selesai dilakukan pengujian oleh tim *QA* dan siap untuk masuk ke lingkungan *production*.
6. **Really Really Done:** merupakan tahap akhir yang menandakan bahwa fitur telah sepenuhnya selesai, diuji, disetujui, dan diterapkan di lingkungan produksi. Semua verifikasi telah dilakukan dan task tersebut dianggap tuntas tanpa memerlukan tindak lanjut tambahan. Tahap ini juga menjadi arsip akhir sebagai dokumentasi keberhasilan penyelesaian setiap bagian proyek.

Setiap kolom menunjukkan status perkembangan tugas dan membantu memastikan bahwa seluruh fitur telah melewati tahapan validasi sebelum dinyatakan selesai. Alur kerja ini memudahkan tim untuk mengidentifikasi hambatan, mengevaluasi hasil uji, serta memastikan stabilitas sistem sebelum diterapkan ke lingkungan produksi. Dengan metode ini, proses pengembangan sistem *Trade Settlement* menjadi lebih transparan, efisien, dan mudah dikontrol secara menyeluruh.

I Hasil Implementasi

Pada bagian ini menjelaskan hasil implementasi dari setiap menu yang terdapat pada sistem *Trade Settlement*. Seluruh menu telah berhasil dikembangkan dan diintegrasikan dengan sistem *backend* menggunakan *API*. Implementasi dilakukan melalui proses pengujian berulang untuk memastikan setiap fungsi berjalan stabil, dan akurat dalam menampilkan serta memproses data. Gambar 3.37 menunjukkan alur pembuatan setiap menu secara garis besarnya. Pembuatan menu secara detail akan ditampilkan pada masing-masing menu, dimana detail pembahasan akan berfokus pada bagian *frontend*.



Gambar 3.37. Flow pembuatan halaman

Proses dimulai dengan *transfer knowledge* dari departemen *trade settlement* kepada para *developer* dengan tujuan untuk memahami kebutuhan bisnis dan alur kerja yang akan diimplementasikan. Kemudian dilakukan analisis terhadap aplikasi terdahulu sebagai referensi dalam merancang fitur *trade settlement*. Tahap berikutnya dibuatlah *database* khusus untuk *trade settlement* menggunakan *MSSQL*. Selanjutnya, tim *backend* memulai pembuatan *API* menggunakan *Java Springboot* dengan *library JPA* yang digunakan untuk mengelola data dan menghubungkan logika bisnis dengan *database*. Disaat yang bersamaan, tim *frontend* mulai mengerjakan *interface* untuk setiap halaman yang akan digunakan dengan membuat beberapa komponen menggunakan bantuan *library MUI Material*.

agar tampilannya lebih konsisten. Setelah *backend* berhasil membuat *API*, tim *frontend* langsung melakukan integrasi terhadap *API* tersebut supaya setiap fitur dapat berfungsi sepenuhnya.

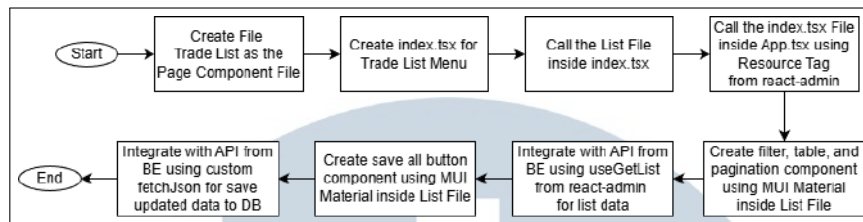
I.1 Menu Trade List

Pada Gambar 3.38 telah menunjukkan bahwa menu *Trade List* berhasil diimplementasikan dengan baik dalam sistem *Trade Settlement*. Menu ini berperan penting dalam menampilkan seluruh data transaksi perdagangan yang telah terjadi. Seluruh proses pengambilan data dilakukan melalui integrasi langsung dengan *Trade Database*, memastikan setiap transaksi yang muncul sudah tervalidasi dan sesuai dengan konfigurasi dari *backend*. Dengan adanya fitur ini, *user* dapat melakukan pemantauan, verifikasi, dan pelaporan data perdagangan dengan lebih cepat dan akurat.

| Buy Order ID | Broker ID | Buy Sell | Client ID | Client HD | Investor Type | Market ID | Market Name | Market Order ID | Market Reference | Market Trade ID | Order ID | Stock ID | Trade Date | Trade ID |
|--------------|-----------|----------|-----------|-----------|---------------|-----------|-------------|-----------------|------------------|-----------------|----------|----------|------------|----------|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |

Gambar 3.38. Hasil implementasi halaman Trade List

Pada menu ini, *user* dapat melihat *trade* yang terjadi pada tanggal tertentu. Selain itu, *user* juga dapat melakukan perubahan data untuk dilakukan penyesuaian apabila terdapat kesalahan pada saat pengambilan data dari *database*. Disediakan juga beberapa *filter* supaya *user* dapat melihat data *trade* secara lebih spesifik. Pada Gambar 3.39 ditunjukkan alur pembuatan *interface* halaman *trade list* yang berfokus pada pengerjaan *frontend*.



Gambar 3.39. Flow pembuatan halaman Trade List

Proses pembuatan menu *trade list* dimulai dengan mempersiapkan *file TradeList.tsx* yang digunakan sebagai *page component*. Kemudian *file index.tsx* dibuat secara khusus untuk memanggil *file TradeList.tsx*. Setelah itu, *file index.tsx* tadi dipanggil ke *file App.tsx* dengan bantuan *tag Resource* dari *library react-admin* supaya dapat menampilkan seluruh *file* seperti *list*, *create*, *show*, dan *edit*. Selanjutnya, komponen seperti *filter*, *table*, dan *pagination* dibuat dengan bantuan *library MUI Material*. Kemudian, *button Save All* dibuat supaya dapat menyimpan perubahan yang ada. Langkah terakhir adalah melakukan integrasi antara *API* dari *backend* dengan *interface* yang dibuat oleh *frontend*. Untuk komponen *filter*, *button* dan *pagination* dibuat menjadi *reusable* komponen supaya *code* menjadi lebih rapi.

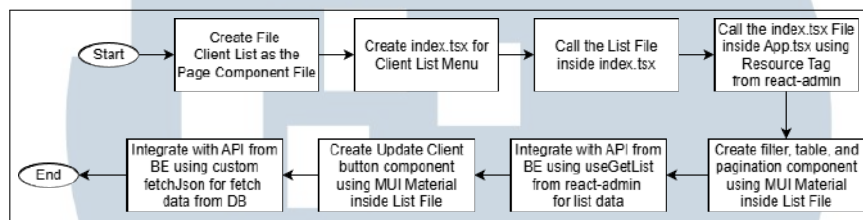
I.2 Menu Client List

Kemudian, Gambar 3.40 menunjukkan menu *Client List* telah berhasil diimplementasikan dengan baik. Menu ini berfungsi untuk menampilkan daftar nasabah yang terdaftar dalam sistem perdagangan. Implementasi yang baik memastikan bahwa seluruh data nasabah langsung tersinkronisasi dengan database *trade database*. Keberhasilan ini telah memudahkan dalam proses audit dan validasi data nasabah.

| Client ID | Name | Email | Category | Type | Trading ID |
|-----------|------|-------|----------|---------------|------------|
| 001 | BEFO | | EQ | INSTITUTIONAL | 11 |
| 002 | BEFO | | EQ | INDIVIDUAL | 11 |
| 003 | BEFO | | EQ | INDIVIDUAL | 11 |
| 004 | BEFO | | EQ | INDIVIDUAL | 11 |
| 005 | SABA | | EQ | INDIVIDUAL | 11 |
| 006 | PRD | | EQ | INDIVIDUAL | 11 |
| 007 | PRD | | EQ | INDIVIDUAL | 11 |
| 008 | PRD | | EQ | INDIVIDUAL | 11 |
| 009 | PRD | | EQ | INDIVIDUAL | 11 |
| 010 | PRD | | EQ | INDIVIDUAL | 11 |
| 011 | PRD | | EQ | INDIVIDUAL | 11 |
| 012 | PRD | | EQ | INDIVIDUAL | 11 |
| 013 | PRD | | EQ | INDIVIDUAL | 11 |
| 014 | PRD | | EQ | INDIVIDUAL | 11 |
| 015 | PRD | | EQ | INDIVIDUAL | 11 |
| 016 | PRD | | EQ | INDIVIDUAL | 11 |
| 017 | PRD | | EQ | INDIVIDUAL | 11 |
| 018 | PRD | | EQ | INDIVIDUAL | 11 |
| 019 | PRD | | EQ | INDIVIDUAL | 11 |
| 020 | PRD | | EQ | INDIVIDUAL | 11 |

Gambar 3.40. Hasil implementasi halaman Client List

Pada menu ini, *user* dapat melihat nasabah yang telah melakukan *trading* dan terdaftar di dalam *trade database* yang berada diluar *server internal*. Pada halaman ini disediakan beberapa *filter* yang dapat digunakan untuk keperluan mencari *user* secara spesifik. Selain itu, disediakan juga tombol untuk melakukan *update* terhadap daftar nasabah supaya dapat menampilkan daftar nasabah terbaru yang didapat dari *trade database*. Pada Gambar 3.41 ditunjukkan alur pembuatan menu *client list* secara detail.



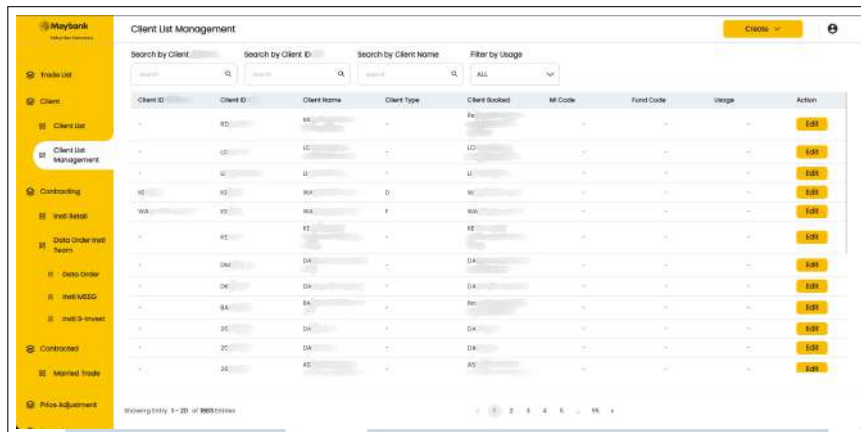
Gambar 3.41. Flow pembuatan halaman Client List

Pembuatan menu ini dimulai dengan mempersiapkan *file ClientList.tsx* yang digunakan sebagai komponen halaman utama menu ini dan dipanggil di dalam *file index.tsx* yang khusus untuk menu *client list*. Berikutnya, *file index.tsx* ini ditaruh pada *tag Resource* milik *library react-admin* yang berada pada *file App.tsx*. Kemudian dibuatlah komponen seperti *filter*, dan *table* khusus untuk menu ini. Komponen *pagination* menggunakan komponen *reusable* yang digunakan di setiap halaman yang membutuhkan *pagination*. Digunakan juga *useGetList* dari *react-admin* yang berfungsi untuk menghubungkan *API* supaya mendapatkan data yang berada dalam *database*. Langkah terakhir yaitu dibuatlah sebuah *button* yang berfungsi untuk melakukan *update* terhadap daftar nasabah yang diintegrasikan dengan menggunakan *custom fetchJson*, dimana *frontend* membuat sebuah *file* konfigurasi untuk menggunakan *library fetchJson* dari *react* supaya sesuai dengan kebutuhan sistem.

I.3 Menu Client List Management

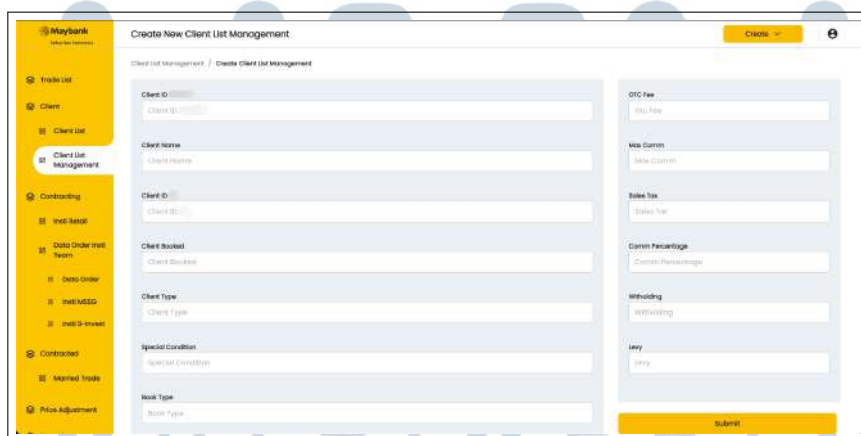
Selanjutnya pada Gambar 3.42 menunjukkan bahwa menu *Client List Management* telah berhasil diimplementasi dengan baik. Menu ini berfungsi untuk mengelola data nasabah secara menyeluruh, seperti penambahan dan pembaruan data. Proses pengembangannya berjalan dengan sukses karena mampu mengintegrasikan fungsi *CRUD* secara aman dan efisien. Selain itu, *interface* dirancang agar *user* dapat melakukan perubahan data dengan mudah tanpa risiko

kehilangan informasi penting.



Gambar 3.42. Hasil implementasi halaman Client List Management

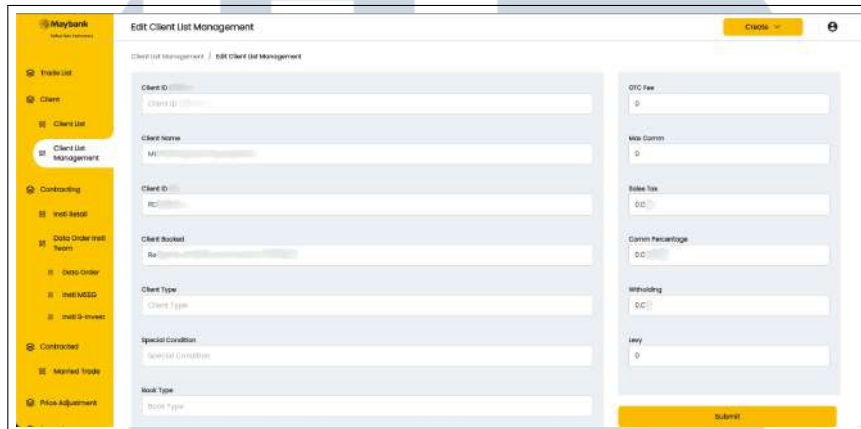
Pada halaman ini, daftar *mapping* yang telah dibuat oleh *user* ditampilkan. Diberikan beberapa *filter* supaya *user* dapat mencari *mapping* nasabah secara spesifik untuk keperluan melihat detail *mapping* ataupun melakukan *edit*. Adanya fitur *search* dan *filter* ini membantu *user* dalam mempercepat proses pengelolaan data nasabah, terutama ketika jumlah data yang ditampilkan cukup besar. Pada Gambar 3.43 menunjukkan halaman untuk *create client mapping* supaya data nasabah dapat digunakan dengan baik.



Gambar 3.43. Hasil implementasi halaman create Client List Management

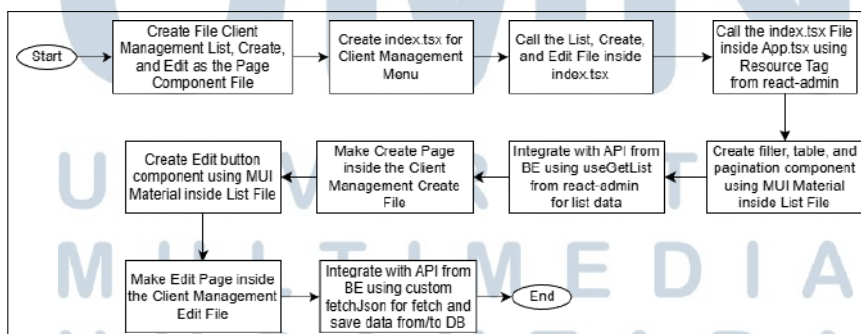
Pada halaman *Create Client List Management* tidak ada *field* yang wajib diisi karena tidak semua nasabah memiliki *field* tertentu. Contohnya pada *field client type* belum tentu nasabah tersebut memiliki tipe khusus, tergantung pada kebijakan atau kategori tertentu dari perusahaan. Hal ini memberikan fleksibilitas

kepada *user* dalam melakukan pengisian data sesuai kebutuhan aktual tanpa harus mematuhi pola input yang kaku. Selanjutnya pada Gambar 3.44 merupakan halaman untuk melakukan pembaruan *mapping data* nasabah. Hal ini telah berhasil diimplementasi dengan baik dan merupakan salah satu fitur penting untuk menjaga data nasabah selalu terbaru.



Gambar 3.44. Hasil implementasi halaman edit Client List Management

Tampilan pada halaman *edit* kurang lebih sama seperti halaman *create*, hanya saja pada halaman ini telah terisi data nasabah yang telah dibuat sebelumnya. *User* dapat memperbarui informasi dengan mudah karena setiap *field* menampilkan data terkini yang tersimpan dalam *database*. Dengan desain ini, keandalan dan akurasi data nasabah dapat terus terjaga, mendukung kebutuhan analisis maupun pelaporan internal secara berkala. Pada Gambar 3.45 ditunjukkan alur pembuatan halaman *client list management* secara detail.



Gambar 3.45. Flow pembuatan halaman Client List Management

Proses pembuatan dimulai dengan mempersiapkan beberapa *file* seperti *ClientManagementList.tsx*, *ClientManagementCreate.tsx*, dan

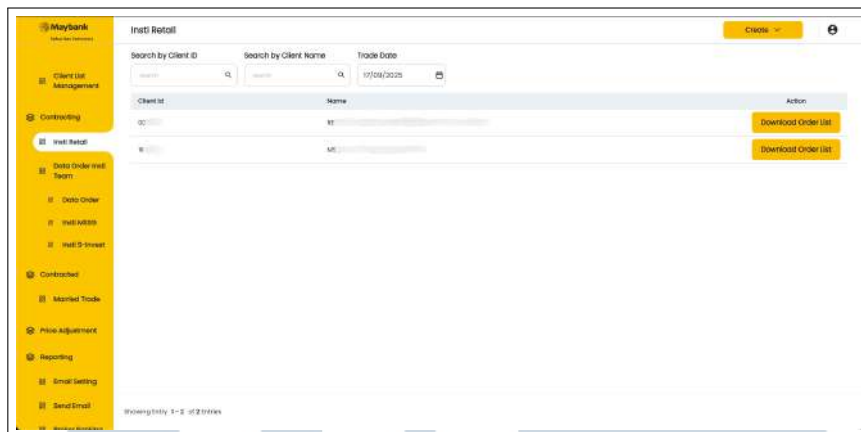
ClientManagementEdit.tsx. Ketiga *file* tersebut kemudian ditaruh pada *file index.tsx* khusus untuk menu *client management*, yang kemudian akan dipanggil pada *file App.tsx* dengan bantuan tag *Resource* dari *library react-admin*. Selanjutnya, pembuatan *table* untuk *client management* dilakukan, sedangkan untuk pembuatan *filter* dan *pagination* menggunakan komponen yang sudah ada pada menu sebelumnya dengan melakukan beberapa penyesuaian sesuai dengan kebutuhan menu ini. Tahap berikutnya melakukan integrasi untuk mendapatkan daftar data *client management* menggunakan bantuan *useGetList* dari *react-admin*.

Tahap berikutnya adalah pembuatan halaman *create*. Untuk halaman *create* tidak ada *field* yang wajib diisi karena tidak semua nasabah memiliki *field* tertentu. Selanjutnya, dibuatlah sebuah tombol *edit* untuk mengarahkan *user* pada halaman *edit*. Pembuatan halaman *edit* sama seperti halaman *create*, hanya saja pada halaman *edit* menggunakan bantuan tag *EditBase* dari *library react-admin* supaya mendapatkan data yang sesuai dengan *client* yang dipilih. Tahap terakhir dilakukan integrasi dengan *API* untuk melakukan *save* dan *fetch* data ke *database* menggunakan *custom fetchJson*.

I.4 Menu Insti Retail

Gambar 3.46 menunjukkan bahwa halaman *Insti Retail* telah berhasil diimplementasikan dengan baik. Menu ini berfungsi untuk menampilkan data transaksi yang dilakukan oleh nasabah ritel pada sistem *Trade Settlement*. Dengan pengaturan *interface* yang intuitif, *user* dapat dengan mudah memfilter dan menelusuri transaksi sesuai kebutuhan pelaporan. Implementasi menu ini mendukung kecepatan proses *monitoring* dan akurasi data yang dibutuhkan dalam proses penyelesaian perdagangan.

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



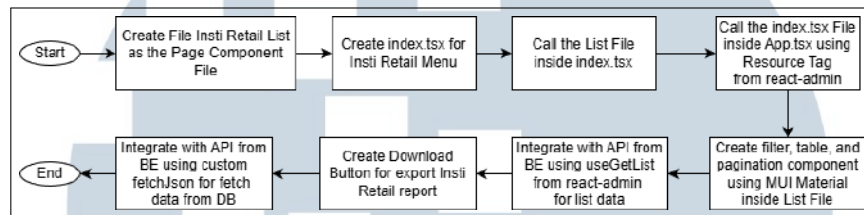
Gambar 3.46. Hasil implementasi halaman Insti Retail

Pada halaman ini telah disediakan beberapa *filter* seperti pada halaman lainnya. *Filter trade date* juga dapat menampilkan data secara historis jika *user* memang memerlukan laporan pada tanggal tertentu. Selanjutnya, Gambar 3.47 menunjukkan hasil *export* yang digunakan untuk kebutuhan laporan operasional perusahaan.

| | A | B | C | D | E | F | G | H | I | J | K | L |
|----|-------------|----------|---------------|---------|----------|---------|---------|----------|------------|-------------|-----------|----------|
| 1 | OrderDate | MarketID | MarketOrderID | BoardID | BrokerID | BuySell | StockID | ClientID | OrderPrice | OrderVolume | NGRReport | DayTrade |
| 2 | 17-Sep-25 B | | 11 | R | X | B | GOTO 1K | | 56 | 19 | 0 | 0 |
| 3 | 17-Sep-25 B | | 11 | R | X | B | GOTO 1K | | 56 | 1 | 0 | 0 |
| 4 | 17-Sep-25 B | | 11 | R | Z | B | GOTO 1K | | 56 | 53 | 0 | 0 |
| 5 | 17-Sep-25 B | | 11 | R | A | B | GOTO 1K | | 56 | 64 | 0 | 0 |
| 6 | 17-Sep-25 B | | 11 | R | C | B | GOTO 1K | | 56 | 2 | 0 | 0 |
| 7 | 17-Sep-25 B | | 11 | R | A | B | GOTO 1K | | 56 | 12 | 0 | 0 |
| 8 | 17-Sep-25 B | | 11 | R | X | B | GOTO 1K | | 56 | 30 | 0 | 0 |
| 9 | 17-Sep-25 B | | 25 | R | C | S | AMMS 1K | | 232 | 30 | 0 | 0 |
| 10 | 17-Sep-25 B | | 22 | R | C | S | AMMS 1K | | 232 | 30 | 0 | 0 |
| 11 | 17-Sep-25 B | | 11 | R | D | B | GOTO 1K | | 56 | 29 | 0 | 0 |
| 12 | 17-Sep-25 B | | 13 | R | G | S | AMMS 1K | | 230 | 10 | 0 | 0 |
| 13 | 17-Sep-25 B | | 11 | R | P | B | GOTO 1K | | 56 | 10 | 0 | 0 |
| 14 | 17-Sep-25 B | | 11 | R | C | B | GOTO 1K | | 56 | 20 | 0 | 0 |
| 15 | 17-Sep-25 B | | 11 | R | Z | B | GOTO 1K | | 56 | 18 | 0 | 0 |
| 16 | 17-Sep-25 B | | 11 | R | Z | B | GOTO 1K | | 56 | 11 | 0 | 0 |
| 17 | 17-Sep-25 B | | 11 | R | Z | B | GOTO 1K | | 56 | 18 | 0 | 0 |
| 18 | 17-Sep-25 B | | 11 | R | Z | B | GOTO 1K | | 56 | 45 | 0 | 0 |
| 19 | 17-Sep-25 B | | 11 | R | Z | B | GOTO 1K | | 56 | 14 | 0 | 0 |
| 20 | 17-Sep-25 B | | 11 | R | X | B | GOTO 1K | | 56 | 15 | 0 | 0 |
| 21 | 17-Sep-25 B | | 11 | R | P | B | GOTO 1K | | 56 | 3 | 0 | 0 |
| 22 | 17-Sep-25 B | | 11 | R | P | B | GOTO 1K | | 56 | 50 | 0 | 0 |
| 23 | 17-Sep-25 B | | 11 | R | Z | B | GOTO 1K | | 56 | 58 | 0 | 0 |
| 24 | 17-Sep-25 B | | 11 | R | Z | B | GOTO 1K | | 56 | 94 | 0 | 0 |
| 25 | 17-Sep-25 B | | 13 | R | C | S | AMMS 1K | | 228 | 10 | 0 | 0 |
| 26 | 17-Sep-25 B | | 22 | R | C | B | MDIY 1K | | 1070 | 11 | 0 | 0 |
| 27 | 17-Sep-25 B | | 11 | R | X | B | GOTO 1K | | 56 | 15 | 0 | 0 |
| 28 | 17-Sep-25 B | | 11 | R | K | B | GOTO 1K | | 56 | 10 | 0 | 0 |
| 29 | 17-Sep-25 B | | 11 | R | R | B | GOTO 1K | | 56 | 76 | 0 | 0 |
| 30 | 17-Sep-25 B | | 11 | R | Z | B | GOTO 1K | | 56 | 56 | 0 | 0 |
| 31 | 17-Sep-25 B | | 11 | R | C | B | GOTO 1K | | 56 | 10 | 0 | 0 |
| 32 | 17-Sep-25 B | | 11 | R | Z | B | GOTO 1K | | 56 | 90 | 0 | 0 |
| 33 | 17-Sep-25 B | | 11 | R | X | B | GOTO 1K | | 56 | 2 | 0 | 0 |
| 34 | 17-Sep-25 B | | 11 | R | X | B | GOTO 1K | | 56 | 1 | 0 | 0 |
| 35 | 17-Sep-25 B | | 11 | R | Z | B | GOTO 1K | | 56 | 22 | 0 | 0 |
| 36 | 17-Sep-25 B | | 11 | R | K | B | GOTO 1K | | 56 | 20 | 0 | 0 |
| 37 | 17-Sep-25 B | | 22 | R | C | B | MDIY 1K | | 1070 | 22 | 0 | 0 |
| 38 | 17-Sep-25 B | | 11 | R | C | B | GOTO 1K | | 56 | 4 | 0 | 0 |
| 39 | 17-Sep-25 B | | 11 | R | X | B | GOTO 1K | | 56 | 3 | 0 | 0 |
| 40 | 17-Sep-25 B | | 11 | R | Z | B | GOTO 1K | | 56 | 10 | 0 | 0 |
| 41 | 17-Sep-25 B | | 11 | R | Z | B | GOTO 1K | | 56 | 11 | 0 | 0 |
| 42 | 17-Sep-25 B | | 13 | R | G | S | AMMS 1K | | 230 | 20 | 0 | 0 |
| 43 | 17-Sep-25 B | | 11 | R | X | B | GOTO 1K | | 56 | 2 | 0 | 0 |
| 44 | 17-Sep-25 B | | 13 | R | C | S | AMMS 1K | | 228 | 20 | 0 | 0 |
| 45 | 17-Sep-25 B | | 11 | R | X | B | GOTO 1K | | 56 | 2 | 0 | 0 |
| 46 | 17-Sep-25 B | | 11 | R | P | B | GOTO 1K | | 56 | 6 | 0 | 0 |
| 47 | 17-Sep-25 B | | 11 | R | A | B | GOTO 1K | | 56 | 12 | 0 | 0 |
| 48 | 17-Sep-25 B | | 11 | R | X | B | GOTO 1K | | 56 | 30 | 0 | 0 |

Gambar 3.47. Hasil report Insti Retail

Seperti yang terlihat pada Gambar 3.47, laporan yang dihasilkan sudah sesuai dengan yang dibutuhkan oleh departemen terkait. Data pada laporan dianggap sudah benar ketika data yang dihasilkan sudah sama dengan hasil laporan pada aplikasi terdahulu. Selanjutnya, pada Gambar 3.48 ditunjukkan *flow* untuk pembuatan halaman ini.

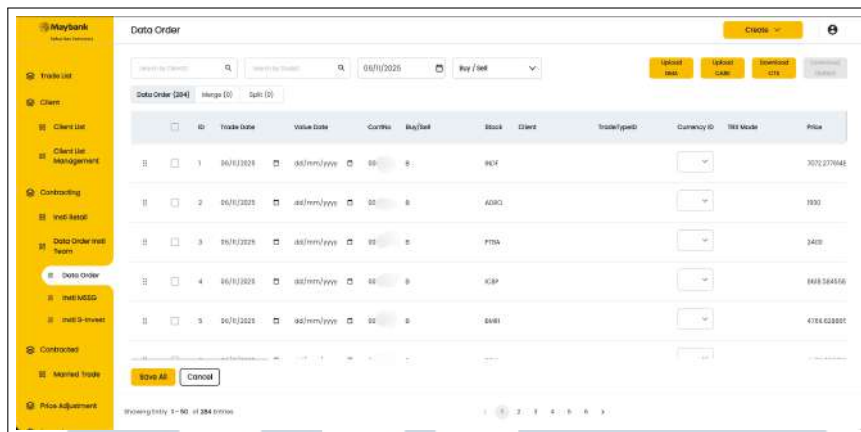


Gambar 3.48. Flow pembuatan halaman Insti Retail

Pembuatan halaman *insti retail* dimulai dengan pembuatan *file InstiRetail.tsx* sebagai komponen utama halaman ini. Kemudian *file* tersebut dipanggil didalam *file index.tsx* milik menu *insti retail*. Selanjutnya, sama seperti menu lainnya *file index.tsx* dipanggil di dalam *file App.tsx*. Kemudian dilakukan pembuatan *filter*, *table* dan *pagination* menggunakan *library mui-material*, dan di integrasikan dengan *API* menggunakan *useGetList* dari *library react-admin*. Tahap terakhir yaitu pembuatan tombol *download report* agar *user* bisa mendapatkan *report* yang diperlukan untuk keperluan operasional perusahaan. Integrasi *download report* menggunakan *custom fetchJson* supaya dapat menghasilkan *report* sesuai yang diinginkan.

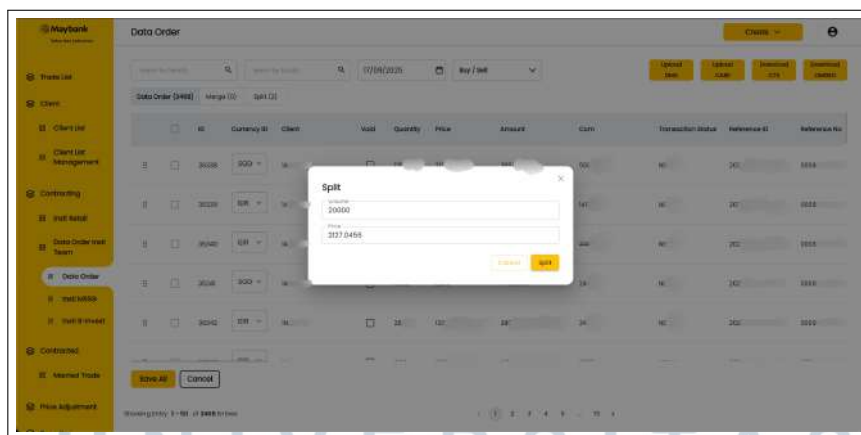
I.5 Menu Data Order

Pada Gambar 3.49 menunjukkan bahwa menu *Data Order* telah berhasil diimplementasikan dengan baik. Menu ini berfungsi untuk mengelola proses *upload* dan pemrosesan *file* transaksi yang berasal dari sistem eksternal, seperti *file DMA* dan *CARE*. Keberhasilan implementasi menu ini membuktikan bahwa sistem mampu menangani data dalam jumlah besar dengan efisien serta menjaga integritas informasi transaksi. Dengan demikian, menu ini menjadi komponen penting dalam fitur *Trade Settlement*, terutama dalam tahap awal pemrosesan data transaksi.



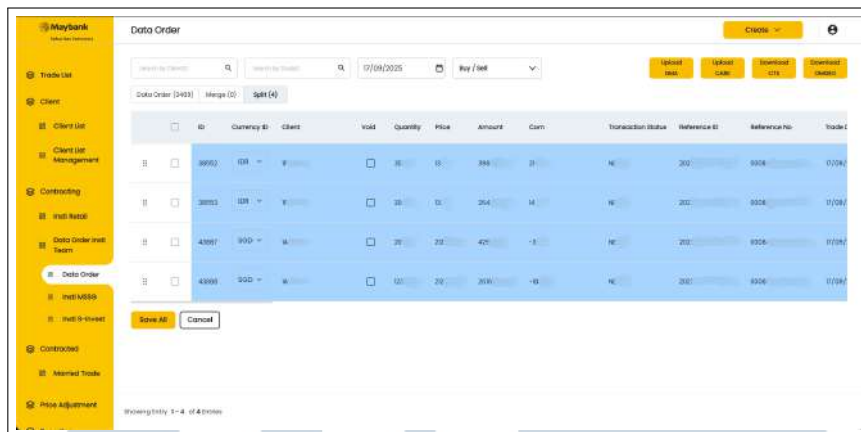
Gambar 3.49. Hasil implementasi halaman Data Order

Sama seperti halaman lainnya, di halaman ini disediakan *filter* yang dapat digunakan *user* untuk mempermudah pencarian data secara spesifik. Selain itu terdapat juga tombol *Upload DMA* dan *CARE* yang berfungsi untuk *upload* data dari *trade database* ke *trade settlement database*, serta tombol *Download CTX* dan *OMGEO* yang berguna untuk mendapatkan *report file* untuk keperluan operasional perusahaan. Selanjutnya pada Gambar 3.50 ditunjukkan *pop up* saat melakukan *split data*.



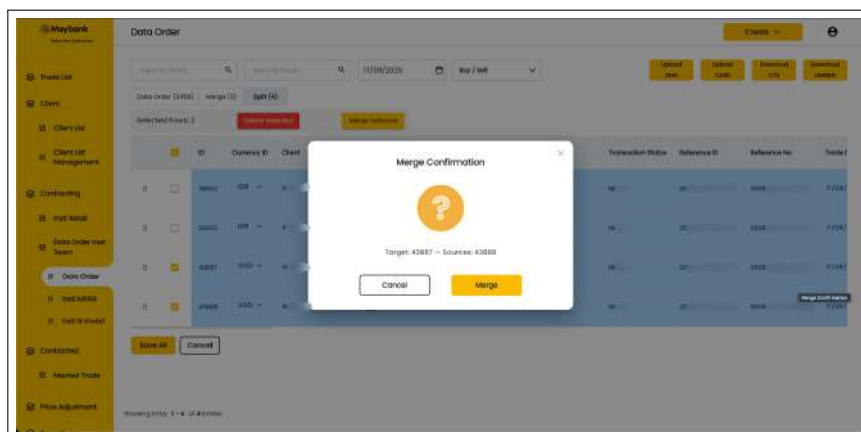
Gambar 3.50. Confirm split data order

Pada Gambar 3.50, *user* dapat memasukkan nilai *volume* dan *price* yang ingin dibagi. Contohnya, jika *volume* data tersebut sebesar dua ribu, kemudian di *split* sebesar seribu lima ratus *volume*, maka hasilnya akan menjadi dua data dengan *volume* data pertama sebesar seribu lima ratus *volume*, dan data kedua sebesar lima ratus *volume*. Kemudian pada Gambar 3.51 ditunjukkan *tab split* yang berisi data yang telah berhasil dipisahkan.



Gambar 3.51. Hasil implementasi halaman split data order

Di bagian ini, ditunjukkan data hasil pemisahan (*split*) yang ditandai dengan *background* berwarna biru. Hal tersebut dilakukan supaya membedakan hasil *split* dengan *merge* karena pada bagian *Data Order* juga menampilkan data hasil *split* dan *merge*. Tujuan dilakukan *split* data yaitu untuk memisahkan satu transaksi besar menjadi beberapa transaksi kecil sesuai dengan kebutuhan pembagian alokasi kepada nasabah. Selanjutnya pada Gambar 3.52 ditunjukkan *pop up* konfirmasi untuk melakukan *merge data*.



Gambar 3.52. Confirm merge data order

Pop up konfirmasi ini dimunculkan agar memitigasi kesalahan *user* ketika sedang melihat data. Selain itu, terdapat syarat untuk melakukan *merge data*, yaitu *client id*, *stock id*, dan *buy sell* harus sama, jika tidak maka *merge* tidak dapat dilakukan. Tujuan dilakukan *merge data* yaitu untuk menggabungkan beberapa transaksi yang memiliki karakteristik serupa menjadi satu entitas data

yang lebih ringkas dan konsisten. Pada Gambar 3.53 ditunjukkan *tab merge* untuk menampilkan seluruh data hasil *merge*.

| ID | Currency ID | Client | Valid | Quantity | Price | Amount | Com | Transaction Status | Reference ID | Reference No | Trade |
|-------|-------------|--------|-------|----------|-------|--------|-----|--------------------|--------------|--------------|-------|
| 43887 | 300 | JA | | 0.0 | 207 | 200 | -W | NE | 203 | 9038 | 2104 |

Gambar 3.53. Hasil implementasi halaman merge data order

Pada bagian *tab merge* menampilkan seluruh data hasil *merge* yang ditandai dengan *background* berwarna *orange*. Pada setiap data di halaman *data order*, *user* dapat melakukan *edit* terhadap kolom tertentu seperti *price*, *quantity*, *amount*, dan masih banyak lagi. Hal tersebut dapat dilakukan ketika terjadi kesalahan data, *user* dapat segera memperbaikinya. Selanjutnya pada Gambar 3.54 ditunjukkan laporan hasil *download CTX*.

| Order ID | Order Date | Order Time | Order Type | Order Status | Order Price | Order Quantity | Order Amount | Order Commission | Order Fee | Order Tax | Order Total | Order Currency | Order Unit | Order Symbol | Order Description | Order Remarks | Order Created By | Order Created At | Order Modified By | Order Modified At | Order Deleted By | Order Deleted At |
|----------------------|------------|------------|------------|--------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| 10000000000000000000 | 2023-09-15 | 10:00:00 | B | Open | 10000000000000000000 | 10000000000000000000 | 10000000000000000000 | 10000000000000000000 | 10000000000000000000 | 10000000000000000000 | 10000000000000000000 | 10000000000000000000 | 10000000000000000000 | 10000000000000000000 | 10000000000000000000 | 10000000000000000000 | 10000000000000000000 | 10000000000000000000 | 10000000000000000000 | 10000000000000000000 | 10000000000000000000 | 10000000000000000000 |

Gambar 3.54. Hasil report CTX

Hasil *report CTX* sudah sesuai dengan laporan yang dihasilkan oleh aplikasi sebelumnya. Laporan ini nantinya akan diberikan kepada pengelola *trade database* untuk diproses lebih lanjut. Hal ini menunjukkan bahwa sistem baru mampu mereplikasi sekaligus meningkatkan keakuratan dan kecepatan proses

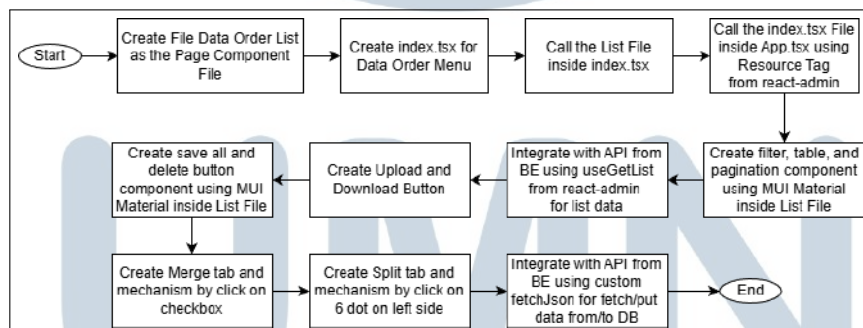
pembuatan laporan transaksi. Selain itu, format dan struktur data yang dihasilkan telah disesuaikan agar kompatibel dengan kebutuhan sistem *backend* serta mudah dianalisis oleh bagian terkait. Berikutnya, pada Gambar 3.55 menampilkan hasil *download report OMGEO*.



| Client ID | Stock ID | Quantity | Price | Direction | Transaction Date | Transaction Time | Transaction Status | Transaction Type | Transaction Category | Transaction Subcategory | Transaction Description | Transaction Remark |
|-----------|-----------|----------|-------|-----------|------------------|------------------|--------------------|------------------|----------------------|-------------------------|--|--------------------|
| 123456789 | 987654321 | 100 | 1000 | BUY | 2023-10-27 | 10:00:00 | Success | Normal | Normal | Normal | Buy 100 shares of Stock ID 987654321 at price 1000. | |
| 123456789 | 987654321 | 100 | 1000 | SELL | 2023-10-27 | 10:05:00 | Success | Normal | Normal | Normal | Sell 100 shares of Stock ID 987654321 at price 1000. | |

Gambar 3.55. Hasil report OMGEO

Laporan *OMGEO* berisi data hasil konfirmasi transaksi antara pihak *buy-side* dan *sell-side* yang telah melalui proses pencocokan (*trade matching*). Proses ini memastikan bahwa data transaksi seperti *Client ID*, *Stock ID*, jumlah saham, harga, serta arah transaksi (*buy/sell*) telah sesuai. Dengan adanya laporan ini, sistem *Trade Settlement* dapat menjamin integritas dan konsistensi data sebelum tahap penyelesaian transaksi dilakukan. Dengan demikian, fitur ini berperan penting dalam meningkatkan efisiensi operasional dan mengurangi risiko kesalahan dalam proses penyelesaian transaksi. Pada Gambar 3.56 menunjukkan alur halaman ini dibuat yang berfokus pada bagian *frontend*.



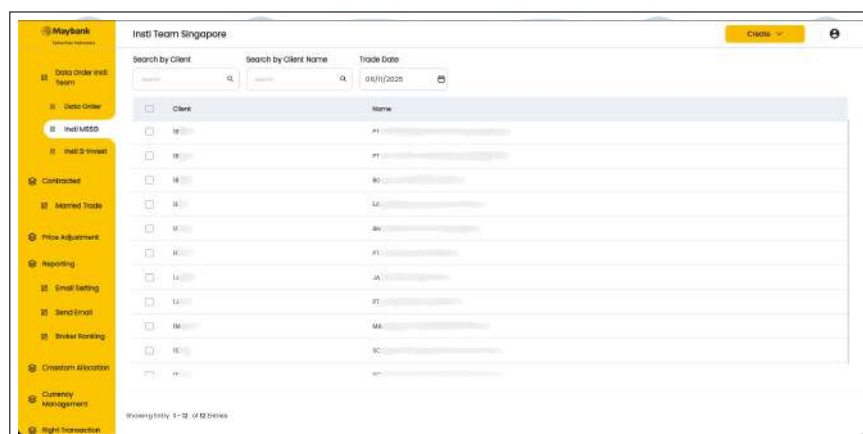
Gambar 3.56. Flow pembuatan halaman Data Order

Seperti menu sebelumnya, proses pembuatan dimulai dengan menyiapkan *file DataOrderList.tsx* sebagai komponen halaman utama menu ini. Kemudian, halaman tersebut dipanggil kedalam *file index.tsx* khusus untuk menu *data order* dan *file index.tsx* tersebut dipanggil lagi kedalam *file App.tsx*. Berikutnya, dibuat komponen *table* untuk halaman ini, untuk komponen *filter* dan *pagination* menggunakan komponen yang sudah ada, hanya disesuaikan pengaturannya. Integrasi data menggunakan *useGetList* dari *library react-admin* dengan menyesuaikan *data*, *sorting*, dan *pagination*.

Selanjutnya dilakukan pembuatan untuk tombol *upload*, *download*, *save all*, dan *delete* menggunakan *library mui-material* supaya *style* tombol seragam. Kemudian *frontend* membuat mekanisme *merge data* dengan cara memilih minimal dua data dengan melakukan centang pada *checkbox*, sekaligus membuat *tab merge* untuk melihat seluruh data hasil *merge*. Setelah itu, mekanisme *split* dibuat dengan cara menekan *dot* yang berada di paling kiri tabel, kemudian akan muncul *pop up* untuk mengisi *volume* dan *price* yang ingin dipisahkan. Semua mekanisme ini diintegrasikan dengan menggunakan *custom fetchJson* agar mendapatkan hasil yang sesuai.

I.6 Menu Insti MSSG

Pada menu *Insti MSSG* berfungsi untuk menampilkan hasil alokasi transaksi dari proses *upload file DMA* maupun *CARE*. Melalui halaman ini, *user* dapat mencari data berdasarkan nama atau kode *client*, serta memilih tanggal perdagangan yang ingin ditampilkan. *Interface* halaman dirancang agar mudah dipahami dan efisien digunakan untuk meninjau data dalam jumlah besar. Setiap entri transaksi ditampilkan secara terstruktur sehingga *user* dapat dengan cepat melakukan pemeriksaan dan analisis terhadap hasil alokasi yang telah diproses sistem, seperti terlihat pada Gambar 3.57.



Gambar 3.57. Hasil implementasi halaman Insti MSSG

Hasil implementasi menunjukkan bahwa sistem berhasil menampilkan seluruh data alokasi dengan cepat dan akurat. Fitur *search* dan pemilihan *trade date* membantu *user* menemukan data secara spesifik tanpa perlu melakukan proses manual. Selain itu, halaman ini telah diintegrasikan dengan sistem *backend* untuk

mengambil data dari *trade database*, sehingga seluruh informasi yang muncul selalu diperbarui sesuai data terakhir yang tersedia. Selanjutnya, Gambar 3.58 menampilkan hasil laporan *Insti MSSG* dalam format *PDF*.

Maybank
Santitas Indonesia

TRADE CONFIRMATION

General Enquiry: 0 2244 1100
J. Raya Pribadi: 0 2244 1101 - 1102
Telp: 021-221 8888 18883, Fax: 021-221 8888 18881

To: [Redacted]
From: [Redacted]
Trade Date: 17-Sep-25
Value Date: 19-Sep-25

We have executed the following trade(s) for account:

| No | CN | B/S | IDN | Code | Quantity | Avg Price | Gross Amount | Comm | Levy | VAT | Sales Tax | PPH22 (%) | Net Amount |
|----------------------------|----|-----|-----|------|----------|-----------|--------------|----------|-----------------|-----|-----------|-----------|------------|
| For the account of: | | | | | | | | | | | | | |
| 1 | 00 | B | ID1 | MYOR | 1.2 | 2.0 | 2.4 | 4.8 | 0.00 | 400 | 2.5 | MF | 2.0 |
| 2 | 00 | B | ID1 | AMRT | 1.2 | 2.1 | 2.5 | -19 | 0.00 | 26 | -3 | | 2.0 |
| | | | | | | | SGD Rate: | 1.234 08 | SGD Equivalent: | | 2.5 | | |
| Total Net Amount Buy | | | | | | | | | | | | | 5.2 |
| Total | | | | | | | | | | | | | 5.2 |
| Total Converted Amount SGD | | | | | | | | | | | | | 4.3 |
| 1 | 00 | S | | BRS | 32 | 2.8 | 89.6 | 1.4 | 0.00 | 10 | 98 | 25 | 86.1 |
| Total Net Amount Sell | | | | | | | | | | | | | 86.1 |
| Total | | | | | | | | | | | | | 86.1 |

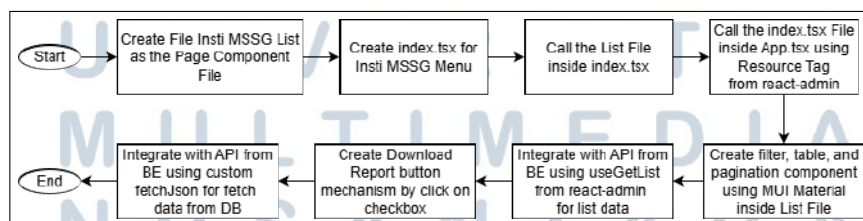
For settlement (GR) Currency, Please transfer the payment to:
Bank Maybank Indonesia (GR) A/C No.: [Redacted]
PT: [Redacted]
Swift Code: [Redacted]

Contents of this statement will be considered correct if no discrepancy is reported within 24 hours. Please contact operation department at 021-221 8888 18883 for any claim.
This is an authorized print out, therefore no signature needed.

Page 1 of 2

Gambar 3.58. Hasil report Insti MSSG

Laporan ini merupakan dokumen resmi *Trade Confirmation* yang berisi rincian transaksi seperti *stock code*, *quantity*, *average price*, *sales tax*, dan *net amount*. Dokumen ini dapat langsung di *export* melalui tombol *Download Report*, yang mempermudah *user* dalam memperoleh laporan tanpa harus mengolah data secara manual. Terakhir, pada Gambar 3.59 ditunjukkan alur proses pembuatan halaman *Insti MSSG*.



Gambar 3.59. Flow pembuatan halaman Insti MSSG

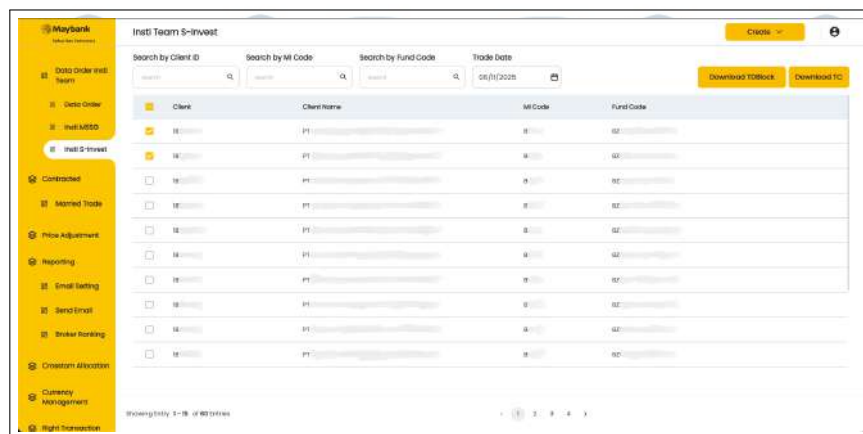
Proses *development* dimulai dari pembuatan *file InstiMSSGList.tsx* sebagai komponen halaman utama menu ini. Selanjutnya, *file index.tsx* dibuat untuk

memanggil file *InstiMSSGList.tsx*. Sama seperti halaman sebelumnya, tag *Resource* digunakan untuk memanggil *index.tsx* milik menu *insti mssg*. Tag *Resource* ini dapat memanggil halaman *list*, *create*, *edit*, dan *show* jika di dalam file *index.tsx* terdapat halaman tersebut. Integrasi menggunakan *useGetList* dari *react-admin* digunakan untuk mengambil data lewat *api* yang diberikan oleh *backend*.

Selanjutnya, pembuatan tombol *download report* diimplementasikan dengan cara mencentang *checkbox* yang disediakan di setiap *row* data. Setelah itu, *user* dapat melakukan *download* data *pdf* maupun *xlsx* sesuai tombol yang digunakan. Integrasi dengan *API* menggunakan *custom fetchJson* supaya respon yang diterima dapat mudah dibaca dan disesuaikan dengan kebutuhan.

I.7 Menu Insti S-Invest

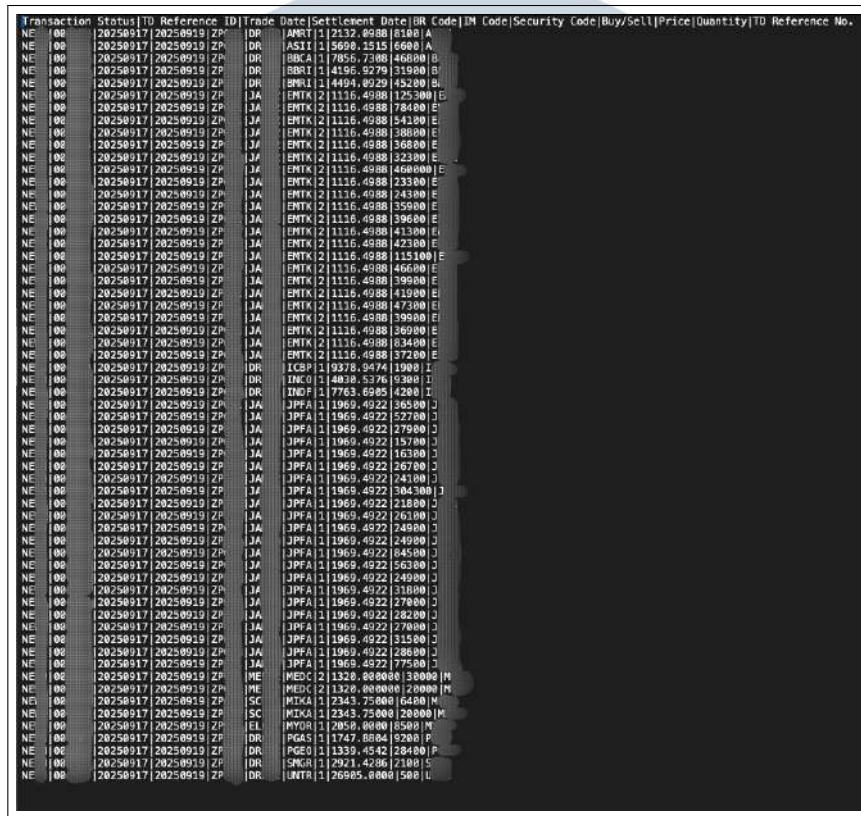
Menu berikutnya yaitu *Insti S-Invest*, digunakan untuk mengelola dan menampilkan data transaksi dari institusi *S-Invest*. Melalui halaman ini, *user* dapat melakukan pencarian berdasarkan *Client ID*, *MI Code*, dan *Fund Code*, serta menentukan *trade date* yang diinginkan. Selain itu, disediakan dua tombol utama yaitu *Download TDBlock* dan *Download TC* untuk mempermudah proses ekspor data ke dalam format teks sesuai standar sistem pelaporan. Tampilan antarmuka dari menu ini ditunjukkan pada Gambar 3.60.



Gambar 3.60. Hasil implementasi halaman Insti S-Invest

Selanjutnya pada Gambar 3.61 ditampilkan hasil laporan *TD Block* yang dihasilkan dari menu *Insti S-Invest*. Laporan ini berisi rincian transaksi seperti *trade date*, *stock code*, *price*, dan *quantity* yang telah dilakukan oleh nasabah institusi. Data tersebut diperoleh langsung dari sistem *backend* setelah proses pemrosesan

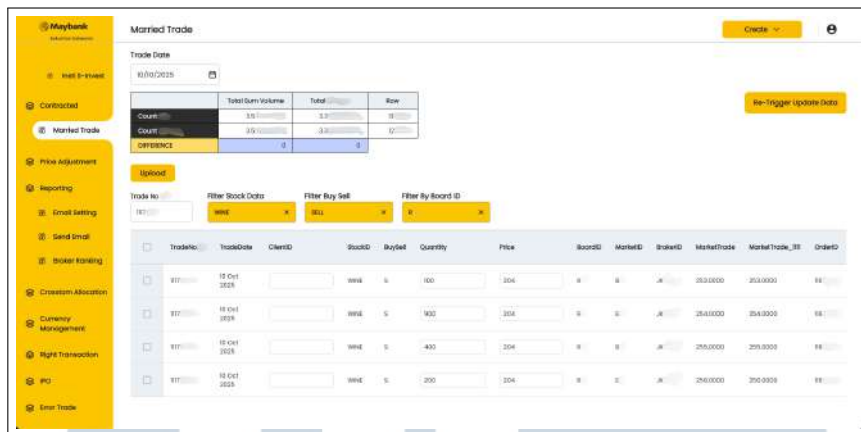
selesai, kemudian diekspor dalam format teks agar dapat digunakan untuk integrasi atau pelaporan lanjutan.



| Transaction | Status | ID | Reference | ID | Trade Date | Settlement Date | BR Code | IN Code | Security Code | Buy/Sell | Price | Quantity | ID | Reference No. |
|-------------|--------|----|-----------|----------|------------|-----------------|---------|---------|---------------|----------|-------|----------|----|---------------|
| NE | 00 | | 20250917 | 20250919 | ZP | DR | AMRT | 1 | 2132.0000 | 8100 | A | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | DR | ASII | 1 | 5600.1515 | 6600 | A | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | DR | BKIA | 1 | 7050.7200 | 4600 | B | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | DR | BBRI | 1 | 4590.5270 | 3100 | B | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | DR | BMRI | 1 | 4490.0020 | 4520 | B | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | ENTK | 2 | 1110.4980 | 12500 | E | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | ENTK | 2 | 1110.4980 | 7000 | E | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | ENTK | 2 | 1110.4980 | 5420 | E | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | ENTK | 2 | 1110.4980 | 3000 | E | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | ENTK | 2 | 1110.4980 | 3600 | E | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | ENTK | 2 | 1110.4980 | 3230 | E | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | ENTK | 2 | 1110.4980 | 4000 | E | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | ENTK | 2 | 1110.4980 | 2330 | E | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | ENTK | 2 | 1110.4980 | 2430 | E | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | ENTK | 2 | 1110.4980 | 3500 | E | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | ENTK | 2 | 1110.4980 | 3900 | E | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | ENTK | 2 | 1110.4980 | 4130 | E | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | ENTK | 2 | 1110.4980 | 4230 | E | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | ENTK | 2 | 1110.4980 | 11500 | E | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | ENTK | 2 | 1110.4980 | 4000 | E | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | ENTK | 2 | 1110.4980 | 3900 | E | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | ENTK | 2 | 1110.4980 | 4190 | E | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | ENTK | 2 | 1110.4980 | 4730 | E | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | ENTK | 2 | 1110.4980 | 3900 | E | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | ENTK | 2 | 1110.4980 | 3000 | E | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | ENTK | 2 | 1110.4980 | 8340 | E | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | ENTK | 2 | 1110.4980 | 3720 | E | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | DR | ICBP | 1 | 9370.9470 | 1900 | I | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | DR | INDO | 1 | 4030.5370 | 9300 | I | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | DR | INDO | 1 | 7763.8900 | 4200 | I | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | JPPA | 1 | 1960.4022 | 3050 | J | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | JPPA | 1 | 1960.4022 | 5270 | J | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | JPPA | 1 | 1960.4022 | 2790 | J | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | JPPA | 1 | 1960.4022 | 1570 | J | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | JPPA | 1 | 1960.4022 | 1630 | J | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | JPPA | 1 | 1960.4022 | 2670 | J | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | JPPA | 1 | 1960.4022 | 2410 | J | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | JPPA | 1 | 1960.4022 | 3040 | J | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | JPPA | 1 | 1960.4022 | 2180 | J | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | JPPA | 1 | 1960.4022 | 2610 | J | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | JPPA | 1 | 1960.4022 | 2490 | J | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | JPPA | 1 | 1960.4022 | 2490 | J | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | JPPA | 1 | 1960.4022 | 8450 | J | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | JPPA | 1 | 1960.4022 | 5630 | J | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | JPPA | 1 | 1960.4022 | 2490 | J | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | JPPA | 1 | 1960.4022 | 3100 | J | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | JPPA | 1 | 1960.4022 | 2700 | J | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | JPPA | 1 | 1960.4022 | 2820 | J | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | JPPA | 1 | 1960.4022 | 2700 | J | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | JPPA | 1 | 1960.4022 | 3150 | J | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | JPPA | 1 | 1960.4022 | 2200 | J | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | JA | JPPA | 1 | 1960.4022 | 7750 | J | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | ME | MEDC | 2 | 1320.0000 | 3000 | M | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | ME | MEDC | 2 | 1320.0000 | 2000 | M | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | ME | MEDC | 2 | 1320.0000 | 6000 | M | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | SC | MIKA | 1 | 2343.7500 | 2000 | M | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | EL | MYOR | 1 | 2050.0000 | 8500 | M | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | DR | PGAS | 1 | 1747.8000 | 9200 | P | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | DR | PGEI | 1 | 1339.4542 | 2000 | P | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | DR | SNGR | 1 | 2011.4200 | 2100 | S | | | |
| NE | 00 | | 20250917 | 20250919 | ZP | DR | UNITR | 1 | 26905.0000 | 500 | U | | | |

Gambar 3.61. Hasil report TD Block

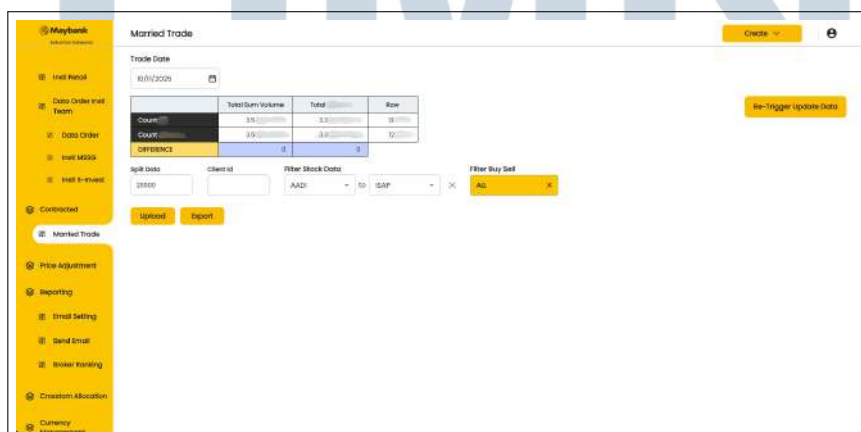
Hasil report *TD Block* ini berisi seluruh data nasabah institusi yang bertransaksi pada hari itu. Seluruh data ini membantu memastikan bahwa hasil perdagangan yang dikirim oleh pihak manajer investasi telah sesuai dengan data yang diterima oleh broker. Selain itu, format laporan disusun sesuai standar pertukaran data antar sistem agar dapat langsung diolah tanpa perlu konversi tambahan. Berikutnya, pada Gambar 3.62 diperlihatkan hasil laporan *Trade Confirmation* (TC) yang merupakan dokumen pelengkap dari *TD Block*. Laporan ini digunakan untuk memastikan bahwa seluruh transaksi telah dikonfirmasi antara pihak *broker* dan *investment manager*.



Gambar 3.64. Hasil implementasi halaman Married Trade ketika terdapat opennull

Hasil implementasi pada kondisi ini menunjukkan bahwa sistem mampu mengidentifikasi data yang tidak *match* saat proses *matching*, ditandai dengan nilai kosong pada kolom *ClientID* tertentu. Fitur ini membantu *user* untuk mendeteksi anomali sebelum melakukan proses *matching*, sehingga data dapat diperbaiki terlebih dahulu. Selain itu, sistem memberikan fleksibilitas untuk memilih data yang ingin diperbaiki, sehingga proses validasi menjadi lebih cepat dan efisien.

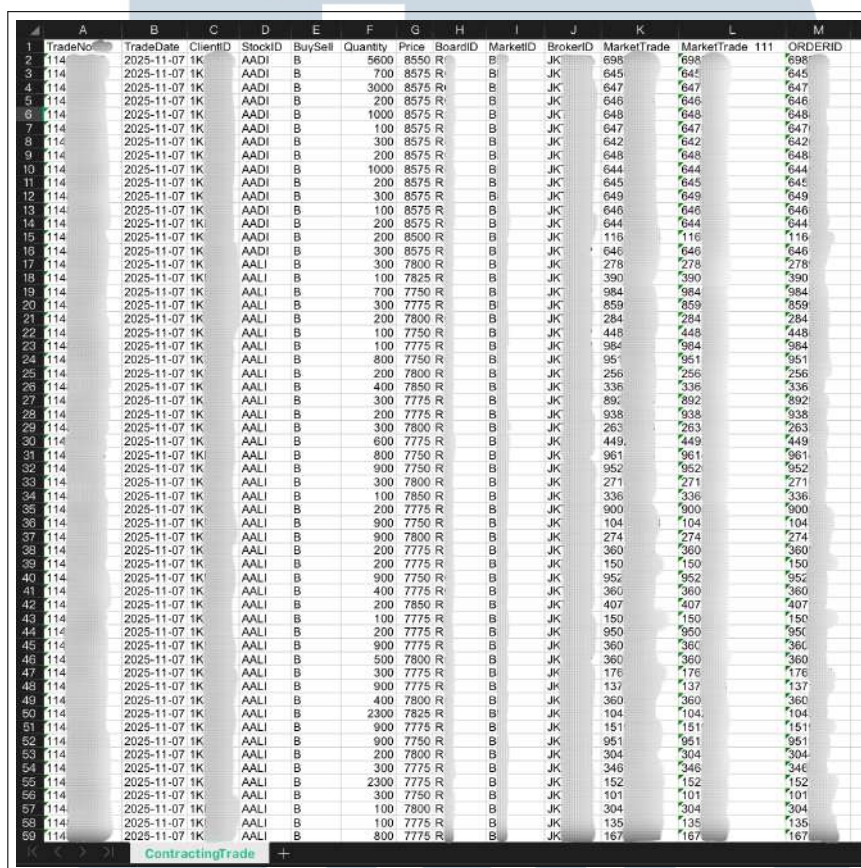
Setelah *user* mengisi seluruh nilai kosong pada kolom *client id* di setiap data yang terdeteksi sebagai *opennull*, halaman *Married Trade* akan menampilkan kondisi data yang telah diperbarui. Pada tahap ini, data dari *Trade Database* dan *Fidessa* telah selesai dilakukan proses pencocokan untuk memastikan kesesuaian jumlah, harga, dan detail transaksi lainnya. Tampilan sistem setelah *opennull* berhasil diperbaiki ditunjukkan pada Gambar 3.65.



Gambar 3.65. Hasil implementasi halaman Married Trade setelah mengisi opennull

Hasil tersebut menunjukkan bahwa sistem mampu memperbaiki nilai-

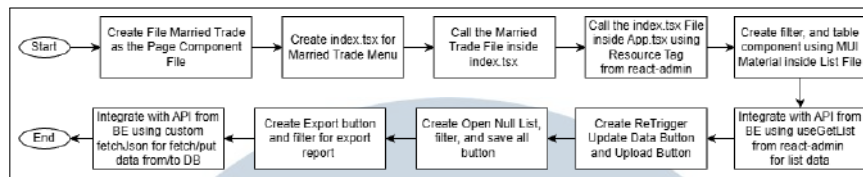
nilai yang sebelumnya kosong, sekaligus melakukan penyajian ulang data secara otomatis. Dengan kondisi data yang telah lengkap, *user* dapat melakukan *export report*. Pada tahap berikutnya, *user* dapat melakukan *export* data yang telah melalui proses verifikasi ke dalam format laporan tertentu untuk kepentingan pelaporan maupun audit. Sistem menyediakan fitur *Export* yang menghasilkan laporan *Married Trade* dalam format standar agar dapat diolah lebih lanjut. Hasil laporan tersebut ditampilkan pada Gambar 3.66.



| # | A | B | C | D | E | F | G | H | I | J | K | L | M |
|----|---------|------------|----------|---------|---------|----------|--------|---------|----------|----------|-------------|-------------|---------|
| 1 | TradeNo | TradeDate | ClientID | StockID | BuySell | Quantity | Price | BoardID | MarketID | BrokerID | MarketTrade | MarketTrade | ORDERID |
| 2 | 114 | 2025-11-07 | 1K | AADI | B | 5600 | 8550 R | B | JK | 698 | 698 | 698 | |
| 3 | 114 | 2025-11-07 | 1K | AADI | B | 700 | 8575 R | B | JK | 645 | 645 | 645 | |
| 4 | 114 | 2025-11-07 | 1K | AADI | B | 3000 | 8575 R | B | JK | 647 | 647 | 647 | |
| 5 | 114 | 2025-11-07 | 1K | AADI | B | 200 | 8575 R | B | JK | 646 | 646 | 646 | |
| 6 | 114 | 2025-11-07 | 1K | AADI | B | 1000 | 8575 R | B | JK | 648 | 648 | 648 | |
| 7 | 114 | 2025-11-07 | 1K | AADI | B | 100 | 8575 R | B | JK | 647 | 647 | 647 | |
| 8 | 114 | 2025-11-07 | 1K | AADI | B | 300 | 8575 R | B | JK | 642 | 642 | 642 | |
| 9 | 114 | 2025-11-07 | 1K | AADI | B | 200 | 8575 R | B | JK | 648 | 648 | 648 | |
| 10 | 114 | 2025-11-07 | 1K | AADI | B | 1000 | 8575 R | B | JK | 644 | 644 | 644 | |
| 11 | 114 | 2025-11-07 | 1K | AADI | B | 200 | 8575 R | B | JK | 645 | 645 | 645 | |
| 12 | 114 | 2025-11-07 | 1K | AADI | B | 300 | 8575 R | B | JK | 649 | 649 | 649 | |
| 13 | 114 | 2025-11-07 | 1K | AADI | B | 100 | 8575 R | B | JK | 646 | 646 | 646 | |
| 14 | 114 | 2025-11-07 | 1K | AADI | B | 200 | 8575 R | B | JK | 644 | 644 | 644 | |
| 15 | 114 | 2025-11-07 | 1K | AADI | B | 200 | 8500 R | B | JK | 116 | 116 | 116 | |
| 16 | 114 | 2025-11-07 | 1K | AADI | B | 300 | 8575 R | B | JK | 646 | 646 | 646 | |
| 17 | 114 | 2025-11-07 | 1K | AALI | B | 300 | 7800 R | B | JK | 278 | 278 | 278 | |
| 18 | 114 | 2025-11-07 | 1K | AALI | B | 100 | 7825 R | B | JK | 390 | 390 | 390 | |
| 19 | 114 | 2025-11-07 | 1K | AALI | B | 700 | 7750 R | B | JK | 984 | 984 | 984 | |
| 20 | 114 | 2025-11-07 | 1K | AALI | B | 300 | 7775 R | B | JK | 859 | 859 | 859 | |
| 21 | 114 | 2025-11-07 | 1K | AALI | B | 200 | 7800 R | B | JK | 284 | 284 | 284 | |
| 22 | 114 | 2025-11-07 | 1K | AALI | B | 100 | 7750 R | B | JK | 448 | 448 | 448 | |
| 23 | 114 | 2025-11-07 | 1K | AALI | B | 100 | 7775 R | B | JK | 984 | 984 | 984 | |
| 24 | 114 | 2025-11-07 | 1K | AALI | B | 800 | 7750 R | B | JK | 951 | 951 | 951 | |
| 25 | 114 | 2025-11-07 | 1K | AALI | B | 200 | 7800 R | B | JK | 256 | 256 | 256 | |
| 26 | 114 | 2025-11-07 | 1K | AALI | B | 400 | 7850 R | B | JK | 336 | 336 | 336 | |
| 27 | 114 | 2025-11-07 | 1K | AALI | B | 300 | 7775 R | B | JK | 892 | 892 | 892 | |
| 28 | 114 | 2025-11-07 | 1K | AALI | B | 200 | 7775 R | B | JK | 938 | 938 | 938 | |
| 29 | 114 | 2025-11-07 | 1K | AALI | B | 300 | 7800 R | B | JK | 263 | 263 | 263 | |
| 30 | 114 | 2025-11-07 | 1K | AALI | B | 600 | 7775 R | B | JK | 449 | 449 | 449 | |
| 31 | 114 | 2025-11-07 | 1K | AALI | B | 800 | 7750 R | B | JK | 961 | 961 | 961 | |
| 32 | 114 | 2025-11-07 | 1K | AALI | B | 900 | 7750 R | B | JK | 952 | 952 | 952 | |
| 33 | 114 | 2025-11-07 | 1K | AALI | B | 300 | 7800 R | B | JK | 271 | 271 | 271 | |
| 34 | 114 | 2025-11-07 | 1K | AALI | B | 100 | 7850 R | B | JK | 336 | 336 | 336 | |
| 35 | 114 | 2025-11-07 | 1K | AALI | B | 200 | 7775 R | B | JK | 900 | 900 | 900 | |
| 36 | 114 | 2025-11-07 | 1K | AALI | B | 900 | 7750 R | B | JK | 104 | 104 | 104 | |
| 37 | 114 | 2025-11-07 | 1K | AALI | B | 900 | 7800 R | B | JK | 274 | 274 | 274 | |
| 38 | 114 | 2025-11-07 | 1K | AALI | B | 200 | 7775 R | B | JK | 360 | 360 | 360 | |
| 39 | 114 | 2025-11-07 | 1K | AALI | B | 200 | 7775 R | B | JK | 150 | 150 | 150 | |
| 40 | 114 | 2025-11-07 | 1K | AALI | B | 900 | 7750 R | B | JK | 952 | 952 | 952 | |
| 41 | 114 | 2025-11-07 | 1K | AALI | B | 400 | 7775 R | B | JK | 360 | 360 | 360 | |
| 42 | 114 | 2025-11-07 | 1K | AALI | B | 200 | 7850 R | B | JK | 407 | 407 | 407 | |
| 43 | 114 | 2025-11-07 | 1K | AALI | B | 100 | 7775 R | B | JK | 150 | 150 | 150 | |
| 44 | 114 | 2025-11-07 | 1K | AALI | B | 200 | 7775 R | B | JK | 950 | 950 | 950 | |
| 45 | 114 | 2025-11-07 | 1K | AALI | B | 900 | 7775 R | B | JK | 360 | 360 | 360 | |
| 46 | 114 | 2025-11-07 | 1K | AALI | B | 500 | 7800 R | B | JK | 360 | 360 | 360 | |
| 47 | 114 | 2025-11-07 | 1K | AALI | B | 300 | 7775 R | B | JK | 176 | 176 | 176 | |
| 48 | 114 | 2025-11-07 | 1K | AALI | B | 900 | 7775 R | B | JK | 137 | 137 | 137 | |
| 49 | 114 | 2025-11-07 | 1K | AALI | B | 400 | 7800 R | B | JK | 360 | 360 | 360 | |
| 50 | 114 | 2025-11-07 | 1K | AALI | B | 2300 | 7825 R | B | JK | 104 | 104 | 104 | |
| 51 | 114 | 2025-11-07 | 1K | AALI | B | 900 | 7775 R | B | JK | 151 | 151 | 151 | |
| 52 | 114 | 2025-11-07 | 1K | AALI | B | 900 | 7750 R | B | JK | 951 | 951 | 951 | |
| 53 | 114 | 2025-11-07 | 1K | AALI | B | 200 | 7800 R | B | JK | 304 | 304 | 304 | |
| 54 | 114 | 2025-11-07 | 1K | AALI | B | 300 | 7775 R | B | JK | 346 | 346 | 346 | |
| 55 | 114 | 2025-11-07 | 1K | AALI | B | 2300 | 7775 R | B | JK | 152 | 152 | 152 | |
| 56 | 114 | 2025-11-07 | 1K | AALI | B | 300 | 7750 R | B | JK | 101 | 101 | 101 | |
| 57 | 114 | 2025-11-07 | 1K | AALI | B | 100 | 7800 R | B | JK | 304 | 304 | 304 | |
| 58 | 114 | 2025-11-07 | 1K | AALI | B | 100 | 7775 R | B | JK | 135 | 135 | 135 | |
| 59 | 114 | 2025-11-07 | 1K | AALI | B | 800 | 7775 R | B | JK | 167 | 167 | 167 | |

Gambar 3.66. Hasil report Married Trade

Laporan *Married Trade* mencakup informasi penting seperti nomor transaksi, kode saham, jumlah transaksi, harga, *board ID*, *market ID*, dan hasil perhitungan *market trade*. Data dalam laporan ini telah melalui proses *matching* antara dua sumber sehingga dapat dipercaya untuk kebutuhan internal perusahaan maupun keperluan regulasi. Fitur ekspor ini memberikan kemudahan bagi *user* untuk menyimpan arsip transaksi dan melaksanakan analisis lanjutan apabila diperlukan. Selanjutnya, proses pembuatan menu *Married Trade* digambarkan pada alur pengembangan yang ditunjukkan pada Gambar 3.67.



Gambar 3.67. Flow pembuatan halaman Married Trade

Proses pengembangan dimulai dari pembuatan *file MarriedTrade.tsx* sebagai komponen halaman, dilanjutkan dengan integrasi melalui *index.tsx* dan *Resource Tag* dari *React Admin*. Setelah itu, dikembangkan komponen filter, tabel, serta tombol aksi seperti *Upload*, *Export*, *Open Null List*, dan *Re-Trigger Update Data*. Integrasi dengan *API* dilakukan menggunakan *fetchJson* dan *useGetList* untuk memastikan data dapat dimuat dan diperbarui secara dinamis. Alur tersebut menunjukkan bahwa pembuatan menu *Married Trade* dilakukan secara sistematis dengan memperhatikan kebutuhan *user* dalam mengelola data perdagangan. Setiap komponen dibangun menggunakan *Material UI* untuk menjaga konsistensi tampilan, sementara integrasi *API* memastikan bahwa data yang tampil selalu terbaru. Dengan pengembangan ini, fitur *Married Trade* dapat berfungsi secara optimal dalam mendukung proses penyelesaian transaksi dan menjaga kualitas data perusahaan.

I.9 Menu Price Adjustment

Pada menu *Price Adjustment* digunakan untuk melakukan penyesuaian harga transaksi antara data perdagangan yang terjadi di pasar dengan harga pemesanan awal dari klien. Melalui halaman ini, *user* dapat melakukan *filter* data berdasarkan *stock* dan *trade date* untuk mengetahui apakah terdapat selisih harga maupun jumlah antara data transaksi aktual dan data pesanan. Sistem dirancang untuk menampilkan ringkasan per saham yang mencakup total kuantitas, harga rata-rata, nilai transaksi, nilai order, dan selisih nilai (*difference amount*). Tampilan awal hasil implementasi halaman ini dapat dilihat pada Gambar 3.68.

| No | Stock Name | QTY | Trade Price | Amount Trade | Order Price | Amount Order | Diff Amount |
|----|----------------|-----------|-------------|---------------|-------------|---------------|-------------|
| 1 | AMD | | | | | | |
| | Sub Total Buy | 11,400 | 8553,4328 | 97,535,059 | 8553,4328 | 97,535,059 | - |
| | Sub Total Sell | 228,705 | 8,808,3,976 | 1,995,452,890 | 8808,3976 | 1,995,452,890 | 0 |
| | Sub Total AMD | 240,105 | 8560,9345 | 2,092,987,959 | 8808,3976 | 2,092,987,959 | 0 |
| 2 | ARI | | | | | | |
| | Sub Total Buy | 176,780 | 7773,9382 | 1,369,395,300 | 7773,9382 | 1,369,395,300 | - |
| | Sub Total Sell | 285,301 | 7,783,2,318 | 1,997,887,500 | 7783,238 | 1,997,887,480 | 0 |
| | Sub Total ARI | 462,081 | 7778,7076 | 2,367,282,800 | 7778,7076 | 2,367,282,780 | 0 |
| 3 | ARI | | | | | | |
| | Sub Total Buy | 2,705,483 | 458,78 | 998,075,855 | 458,78 | 998,075,789 | -12 |
| | Sub Total Sell | 14,300 | 452 | 6,458,400 | 452 | 6,458,400 | - |
| | Sub Total ARI | 17,008 | 459,6389 | 1,004,534,255 | 459,6389 | 1,004,534,197 | -12 |
| 4 | ARI | | | | | | |
| | Sub Total Buy | 338,800 | 1428,0259 | 481,865,500 | 1428,0259 | 481,865,485 | -15 |
| | Sub Total Sell | 827,300 | 1,428,0259 | 1,185,414,400 | 1,428,0259 | 1,185,414,400 | 0 |
| | Sub Total ARI | 1,166,100 | 1,428,0259 | 1,667,280,000 | 1,428,0259 | 1,667,279,885 | -115 |

Gambar 3.68. Hasil implementasi halaman Price Adjustment

Hasil implementasi menunjukkan bahwa sistem mampu menampilkan rekapitulasi per saham secara terstruktur, termasuk subtotal pada transaksi *buy* dan *sell*. Selisih harga yang terdeteksi ditandai secara jelas sehingga memudahkan *user* dalam melakukan analisis dan pengambilan keputusan. Fitur ini membantu tim operasional untuk menilai apakah perbedaan harga perlu ditindaklanjuti ataupun diverifikasi ulang sebelum masuk ke proses pelaporan.

Selanjutnya, halaman *show detail Price Adjustment* pada Gambar 3.69 menampilkan rincian transaksi per nasabah untuk saham tertentu. Melalui halaman ini, *user* dapat melihat setiap transaksi satu per satu mulai dari *trade date*, *client*, kuantitas, harga pasar, harga order, hingga selisih nilai yang muncul dari perbedaan harga tersebut. Halaman ini berfungsi sebagai tampilan detail yang membantu proses pemeriksaan lanjutan secara spesifik.

| No | Transaction Type | Trade Date | Client ID | Client CTX | Quantity Trade | Quantity Order | Trade Price | Amount Trade | Order Price | Amount Order | Diff Amount | Subtotal |
|----|------------------|------------|-----------|------------|----------------|----------------|-------------|--------------|-------------|--------------|-------------|----------|
| 1 | Buy | 21/10/2025 | W | 28 | 81 | 81 | 85 | 6880 | 85 | 6880 | 0 | 0 |
| 2 | Buy | 21/10/2025 | W | 28 | 58 | 58 | 852 | 49316 | 85 | 49316 | -36 | 0 |
| 3 | Buy | 21/10/2025 | W | 28 | 126 | 126 | 809 | 102 | 80 | 650 | 0 | 0 |
| 4 | Buy | 21/10/2025 | W | 28 | 54 | 54 | 82 | 4458 | 82 | 4458 | 0 | 0 |
| 5 | Buy | 21/10/2025 | W | 28 | 33 | 33 | 830 | 2739 | 83 | 2739 | -1 | 0 |
| 6 | Buy | 21/10/2025 | W | 28 | 518 | 518 | 597 | 3092 | 75 | 477 | 0 | 0 |
| 7 | Buy | 21/10/2025 | W | 27 | 82 | 82 | 75 | 6150 | 75 | 6150 | -12 | 0 |
| 8 | Buy | 21/10/2025 | W | 28 | 29 | 29 | 842 | 24418 | 842 | 24418 | 0 | 0 |
| 9 | Buy | 21/10/2025 | W | 27 | 300 | 300 | 775 | 23250 | 77 | 23250 | -480 | 0 |
| 10 | Buy | 21/10/2025 | W | 27 | 50 | 50 | 765 | 3825 | 74 | 3670 | -485 | 0 |
| 11 | Buy | 21/10/2025 | W | 28 | 20 | 20 | 842 | 16840 | 84 | 16840 | -16 | 0 |
| 12 | Buy | 21/10/2025 | W | 28 | 32 | 32 | 82 | 2624 | 82 | 2624 | 0 | 0 |
| 13 | Buy | 21/10/2025 | W | 27 | 290 | 290 | 808 | 23432 | 808 | 23432 | 0 | 0 |
| 14 | Buy | 21/10/2025 | W | 28 | 18 | 18 | 858 | 15444 | 858 | 15444 | 0 | 0 |
| 15 | Buy | 21/10/2025 | W | 28 | 42 | 42 | 82 | 3456 | 762 | 3456 | 1 | 0 |
| 16 | Buy | 21/10/2025 | W | 27 | 81 | 81 | 775 | 62775 | 75 | 61875 | 9 | 0 |
| 17 | Buy | 21/10/2025 | W | 28 | 512 | 512 | 597 | 305664 | 75 | 48480 | -16 | 0 |
| 18 | Buy | 21/10/2025 | W | 28 | 17 | 17 | 822 | 13974 | 822 | 13974 | 0 | 0 |
| 19 | Buy | 21/10/2025 | W | 27 | 81 | 81 | 824 | 66764 | 754 | 62010 | -4754 | 0 |
| 20 | Buy | 21/10/2025 | W | 27 | 26 | 26 | 762 | 19812 | 762 | 19812 | 0 | 0 |
| 21 | Buy | 21/10/2025 | W | 28 | 212 | 212 | 767 | 162604 | 767 | 162604 | -6 | 0 |
| 22 | Buy | 21/10/2025 | W | 28 | 93 | 93 | 824 | 76632 | 824 | 76632 | 0 | 0 |

Gambar 3.69. Hasil implementasi halaman show detail Price Adjustment

Halaman ini memungkinkan *user* untuk memeriksa setiap baris transaksi

yang berkontribusi pada perbedaan harga. Informasi yang ditampilkan sangat lengkap, sehingga proses validasi dapat dilakukan secara menyeluruh. Fitur ini juga penting untuk mengidentifikasi apakah selisih berasal dari perubahan harga pasar, perbedaan jumlah order, ataupun faktor lainnya. Dengan demikian, laporan akhir dapat dipastikan keakuratannya.

Selain penyesuaian harga, modul ini juga menyediakan halaman *Client TRX* yang berfungsi untuk menampilkan rincian transaksi yang telah disesuaikan berdasarkan masing-masing nasabah, yang ditampilkan pada Gambar 3.70. *User* dapat melihat dan mengubah nilai *buy avg price*, *sell avg price*, hingga biaya transaksi seperti *buy fee* dan *sell fee*. Fitur ini memberikan keleluasaan bagi *user* untuk melakukan koreksi data apabila diperlukan.

| No | NO | Trade Date | Client ID | Round | Stock | Buy/Sell | Buy Avg Price | Sell Avg Price | Buy Fee | Sell Fee | Net Amount |
|----|----|------------|-----------|-------|-------|----------|---------------|----------------|---------|----------|------------|
| 1 | 28 | 07/03/25 | 81000 | P | ADN4 | S | 0 | 14 | 0 | 33 | -306 |
| 2 | 28 | 07/03/25 | 81000 | P | ADN4 | S | 0 | 14 | 0 | 33 | -306 |
| 3 | 28 | 07/03/25 | 81000 | P | ADN4 | S | 0 | 14 | 0 | 33 | -306 |
| 4 | 28 | 07/03/25 | 81000 | P | ADN4 | S | 455 | 0 | 68 | 0 | 387 |
| 5 | 28 | 07/03/25 | 81000 | P | ADN4 | S | 455 | 0 | 31 | 0 | 426 |
| 6 | 28 | 07/03/25 | 81000 | P | ADN4 | S | 429 | 0 | 68 | 0 | 361 |
| 7 | 28 | 07/03/25 | 81000 | P | ADN4 | S | 0 | 14 | 0 | 68 | -82 |
| 8 | 28 | 07/03/25 | 81000 | P | ADN4 | S | 0 | 65 | 0 | 37 | -102 |

Gambar 3.70. Hasil implementasi halaman Client TRX

Hasil implementasi menunjukkan bahwa halaman ini mendukung proses *edit* nilai transaksi secara cepat dan akurat. Dengan adanya fitur ini, *user* dapat memastikan bahwa seluruh nilai transaksi yang digunakan dalam proses *settlement* telah sesuai dengan perhitungan yang benar. Informasi yang detail dan terstruktur membantu menurunkan risiko ketidaksesuaian laporan antara pihak nasabah dan broker. Berikutnya, sistem menyediakan fitur untuk *download report price adjustment* untuk keperluan pelaporan internal maupun audit. Laporan ini memuat data lengkap mulai dari transaksi per klien hingga subtotal per saham beserta total keseluruhan. Tampilan laporan hasil ekspor tersebut dapat dilihat pada Gambar 3.71.

Report Price Adjustment Detail

| No | StockID | Date | ClientID | ClientC | QTY | TradePrice | AmountTrade | OrderPrice | AmountOrder | DiffAmount |
|----------------|---------|------------|----------|---------|-----|------------|-------------|------------|-------------|------------|
| AADI | | | | | | | | | | |
| 1 | B | 07/11/2025 | 1K | 28 | 1 | 8.5 | 11 | 8.5 | 11 | 0 |
| Sub Total Buy | | | | | 1 | 8.5 | 11 | 8.5 | 11 | 0 |
| 1 | S | 07/11/2025 | 1K | 28 | 2 | 8.5 | 1.9 | 8.5 | 1.9 | 10 |
| Sub Total Sell | | | | | 2 | 8.5 | 1.9 | 8.5 | 1.9 | 10 |
| Sub Total AADI | | | | | 3 | 8.5 | 2.8 | 8.5 | 2.8 | 10 |
| AALI | | | | | | | | | | |
| 1 | B | 07/11/2025 | 1K | 28 | 8 | 7 | 9 | 7 | 8 | 0 |
| 2 | B | 07/11/2025 | 1K | 28 | 7 | 7 | 5 | 7 | 5 | 0 |
| 3 | B | 07/11/2025 | 1K | 28 | 2 | 7 | 1 | 7 | 1 | 0 |
| Sub Total Buy | | | | | 17 | 7 | 1.3 | 7 | 1.3 | 0 |
| 1 | S | 07/11/2025 | 1K | 28 | 8 | 7 | 5 | 7 | 6 | 4 |
| 2 | S | 07/11/2025 | 1A | 28 | 2 | 7 | 2 | 7 | 2 | 2 |
| 3 | S | 07/11/2025 | 1A | 28 | 2 | 7 | 1 | 7 | 1 | -3 |
| 4 | S | 07/11/2025 | 1A | 28 | 9 | 7 | 7 | 7 | 7 | 6 |
| 5 | S | 07/11/2025 | 1A | 28 | 9 | 7 | 7 | 7 | 7 | -1 |
| 6 | S | 07/11/2025 | 1A | 28 | 7 | 7 | 5 | 7 | 5 | 5 |
| 7 | S | 07/11/2025 | 1A | 28 | 6 | 7 | 4 | 7 | 4 | 4 |
| 8 | S | 07/11/2025 | 1A | 28 | 5 | 7 | 3 | 7 | 3 | 3 |
| 9 | S | 07/11/2025 | 1A | 28 | 4 | 7 | 3 | 7 | 3 | -2 |
| 10 | S | 07/11/2025 | 1A | 28 | 4 | 7 | 3 | 7 | 3 | -3 |
| 11 | S | 07/11/2025 | 1A | 28 | 3 | 7 | 2 | 7 | 2 | 2 |
| 12 | S | 07/11/2025 | 1A | 28 | 3 | 7 | 2 | 7 | 2 | 2 |
| 13 | S | 07/11/2025 | 1A | 28 | 3 | 7 | 2 | 7 | 2 | 2 |
| 14 | S | 07/11/2025 | 1A | 28 | 2 | 7 | 2 | 7 | 2 | 1 |
| 15 | S | 07/11/2025 | 1A | 28 | 2 | 7 | 2 | 7 | 2 | 1 |
| 16 | S | 07/11/2025 | 1A | 28 | 2 | 7 | 2 | 7 | 2 | 1 |
| 17 | S | 07/11/2025 | 1A | 28 | 2 | 7 | 2 | 7 | 2 | 1 |
| 18 | S | 07/11/2025 | 1A | 28 | 1 | 7 | 1 | 7 | 1 | 1 |
| 19 | S | 07/11/2025 | 1A | 28 | 1 | 7 | 1 | 7 | 1 | 1 |
| 20 | S | 07/11/2025 | 1A | 28 | 1 | 7 | 8 | 7 | 8 | 7 |
| Sub Total Sell | | | | | 20 | 7 | 1.9 | 7 | 1.5 | 10 |
| Sub Total AALI | | | | | 38 | 7 | 2.8 | 7.7 | 2.5 | 10 |
| ACES | | | | | | | | | | |
| 1 | B | 07/11/2025 | 1K | 28 | 70 | 45 | 36 | 45 | 36 | -24 |
| 2 | B | 07/11/2025 | 1K | 28 | 35 | 45 | 16 | 45 | 16 | 12 |
| 3 | B | 07/11/2025 | 1A | 28 | 1 | 45 | 45 | 45 | 45 | 0 |
| Sub Total Buy | | | | | 2.1 | 45 | 98 | 45 | 98 | -12 |
| 1 | S | 07/11/2025 | 1K | 28 | 1 | 45 | 6 | 45 | 6 | 0 |
| Sub Total Sell | | | | | 1 | 45 | 6 | 45 | 6 | 0 |
| Sub Total ACES | | | | | 2.1 | 45 | 98 | 45 | 98 | -12 |
| ADMR | | | | | | | | | | |
| 1 | B | 07/11/2025 | 1K | 28 | 32 | 1.4 | 45 | 1.4 | 4 | -5 |
| Sub Total Buy | | | | | 32 | 1.4 | 45 | 1.4 | 4 | -5 |
| 1 | S | 07/11/2025 | 1C | 28 | 20 | 1.4 | 28 | 1.4 | 2 | 0 |
| 2 | S | 07/11/2025 | 1C | 28 | 20 | 1.4 | 28 | 1.4 | 2 | 0 |
| 3 | S | 07/11/2025 | 1C | 28 | 20 | 1.4 | 28 | 1.4 | 2 | 0 |
| 4 | S | 07/11/2025 | 1K | 28 | 32 | 1.4 | 45 | 1.4 | 4 | 14 |
| Sub Total Sell | | | | | 92 | 1.4 | 1.3 | 1.4 | 1.3 | 14 |
| Sub Total ADMR | | | | | 1.2 | 1.4 | 1.7 | 1.4 | 1.7 | 9 |
| Grand Total | | | | | 1.8 | 7.9 | 1.8 | 7.9 | 17 | |

Monday, November 10, 2025

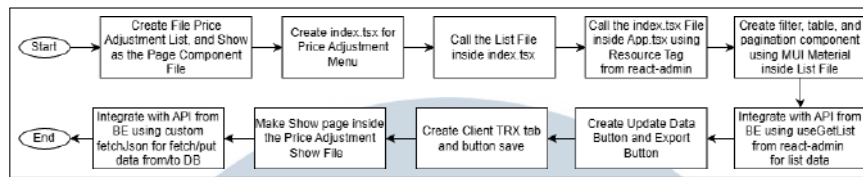
Page 1 of 1

Monday, November 10, 2025

Page 1 of 1

Gambar 3.71. Hasil report Price Adjustment

Laporan ini memberikan ringkasan yang sangat komprehensif mulai dari *buy/sell*, tanggal transaksi, kuantitas, harga pasar, harga order, hingga selisih nilai. Format tabel yang rapi membantu mempercepat proses analisis oleh tim operasional. Selain itu, laporan ini berfungsi sebagai dokumen resmi yang dapat digunakan dalam proses rekonsiliasi lintas departemen maupun keperluan pemeriksaan internal. Laporan ini juga dapat di *download* sesuai dengan *stock* yang dipilih pada *filter stock*, sehingga *user* bisa mendapatkan laporan yang spesifik jika dibutuhkan. Berikutnya, alur pengembangan modul *Price Adjustment* ditunjukkan pada Gambar 3.72.



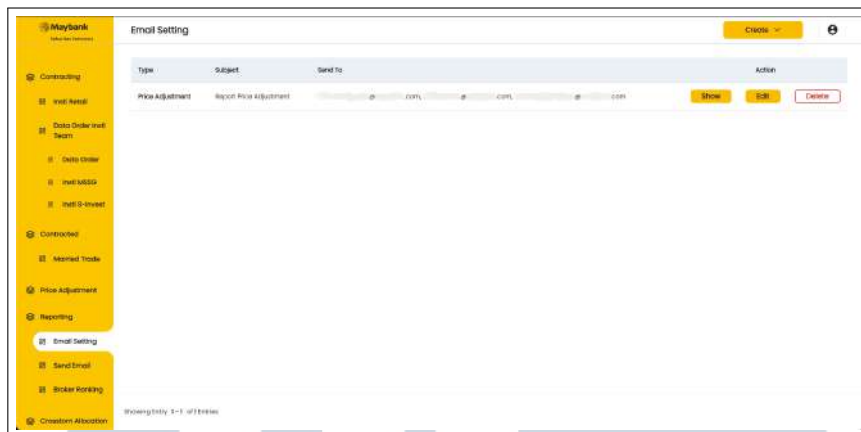
Gambar 3.72. Flow pembuatan halaman Price Adjustment

Proses dimulai dari pembuatan *PriceAdjustmentList.tsx* dan *PriceAdjustmentShow.tsx* sebagai komponen halaman menu ini. Kemudian sama seperti menu lainnya, kedua komponen halaman tersebut dipanggil di dalam *file index.tsx* dan diintegrasikan ke *App.tsx* menggunakan *tag resource*. Pembuatan *filter* dan *table* dilakukan menggunakan *MUI Material* supaya *style* tetap seragam. Selanjutnya, list diintegrasikan dengan *API* dari *backend* menggunakan *useGetList* dari *react-admin* dengan melakukan penyesuaian *sorting* dan *filter*nya.

Fitur tombol seperti *Update Trade Data*, *Update Client TRX*, dan *Download Report* juga ditambahkan untuk mendukung proses. Selanjutnya dibuatlah *client trx tab* yang dapat di *edit* secara langsung pada *list tab*nya dan dapat disimpan menggunakan tombol *save*. Berikutnya pembuatan halaman *show detail price adjustment* dibuat dengan cara yang sama yaitu menggunakan *MUI Material* untuk *tablenya*. Tahap terakhir dilakukan integrasi menggunakan *custom fetchJson* untuk setiap komponen guna memastikan data dapat dimuat secara dinamis dan efisien. Dengan demikian, modul ini menjadi salah satu komponen penting dalam memastikan keakuratan data perdagangan.

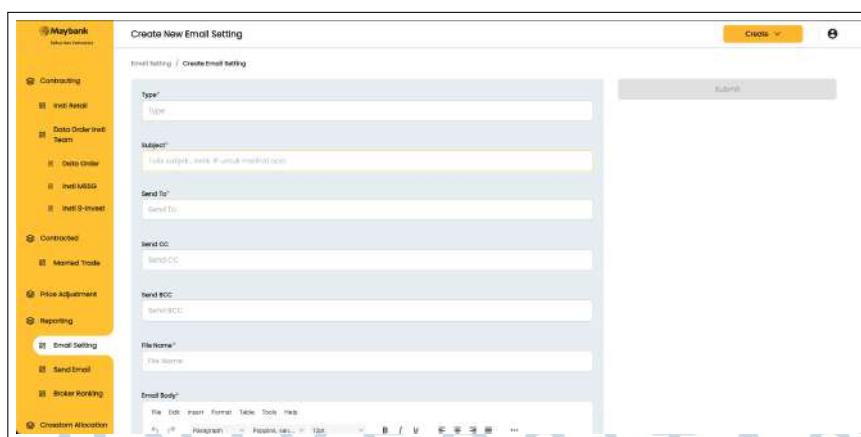
I.10 Menu Email Setting

Menu *Email Setting* merupakan fitur yang digunakan untuk mengatur konfigurasi pengiriman *email* otomatis pada sistem *Trade Settlement*. Melalui halaman ini, *user* dapat melakukan aksi *CRUD* pada pengaturan *email* yang berhubungan dengan proses pelaporan, seperti laporan *Price Adjustment*, *Trade Confirmation*, dan laporan operasional lainnya. Setiap konfigurasi mencakup penentuan tipe laporan, daftar penerima, *CC/BCC*, nama *file* lampiran, serta isi pesan yang akan dikirim. Tampilan daftar utama dari menu ini diperlihatkan pada Gambar 3.73.



Gambar 3.73. Hasil implementasi halaman Email Setting

Hasil implementasi halaman daftar menunjukkan bahwa sistem berhasil menampilkan semua konfigurasi *email* yang tersedia secara jelas dan terstruktur. Pengguna dapat melihat tipe laporan, subjek *email*, serta daftar penerimanya dalam satu tabel. Tersedia pula tombol untuk melihat detail, melakukan *edit* konfigurasi, atau menghapus data, sehingga mempermudah proses pengelolaan pengaturan *email* dalam satu halaman terpusat. Selanjutnya, *user* dapat menambahkan konfigurasi baru melalui halaman *Create* yang dapat dilihat pada Gambar 3.74.



Gambar 3.74. Hasil implementasi halaman create Email Setting

Pada halaman ini, *user* diminta mengisi tipe *email*, subjek, penerima, serta isi pesan yang akan dikirim. Sistem juga menyediakan kolom untuk menentukan nama *file* lampiran agar laporan yang dikirim sesuai dengan standar penamaan dokumen yang digunakan oleh perusahaan. Halaman ini membantu memastikan bahwa setiap pengaturan *email* yang diinput telah lengkap dan sesuai kebutuhan proses pelaporan. *Email setting* ini nantinya akan digunakan pada menu *Send Email*

untuk pengiriman *email* sesuai tipe yang sudah diatur. Selanjutnya, Gambar 3.75 menampilkan halaman *Show Email*, yaitu halaman yang digunakan untuk melihat detail lengkap dari satu konfigurasi *email*.

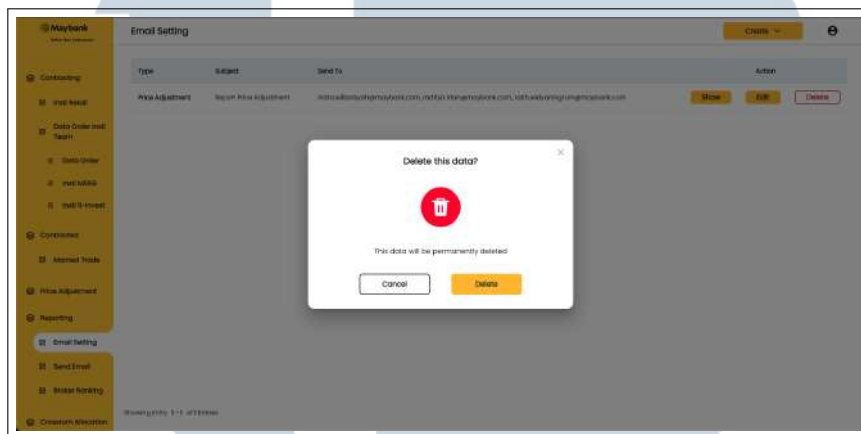
Gambar 3.75. Hasil implementasi halaman show detail Email Setting

Informasi yang ditampilkan termasuk tipe laporan, subjek, daftar penerima email utama, daftar *CC*, *BCC*, nama *file*, serta isi dari pesan yang dikirim. Halaman ini memberikan gambaran lengkap bagi *user* tanpa perlu masuk ke mode edit. Halaman *show* ini berfungsi sebagai referensi cepat bagi *user* sebelum melakukan perubahan atau pengecekan terhadap konfigurasi yang sedang digunakan. Dengan tampilan yang bersih dan terstruktur, *user* dapat memahami isi laporan yang akan dikirim beserta pengaturan penerimanya tanpa perlu membuka beberapa halaman berbeda. Berikutnya, pada Gambar 3.76 ditampilkan halaman *Edit Email Setting*.

Gambar 3.76. Hasil implementasi halaman edit Email Setting

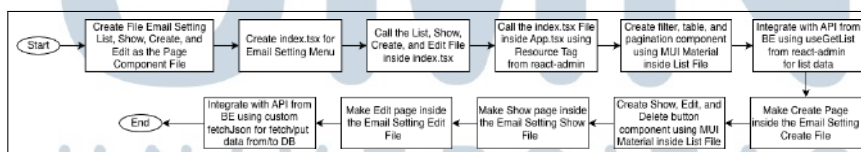
Halaman ini digunakan untuk *edit email setting* apabila terjadi perubahan

penerima, perubahan subjek, perubahan isi pesan, atau perubahan jenis laporan yang dikirimkan. Melalui halaman ini, *user* dapat melakukan revisi data secara langsung dan melakukan penyimpanan kembali agar perubahan dapat diterapkan dalam proses pengiriman laporan berikutnya. Fitur ini memastikan bahwa seluruh pengiriman *email* selalu menggunakan konfigurasi terbaru yang telah disetujui *user*. Pada Gambar 3.77 ditunjukkan tampilan proses penghapusan konfigurasi *email*.



Gambar 3.77. Hasil implementasi delete Email Setting

Fitur ini memungkinkan *user* untuk menghapus pengaturan yang tidak lagi digunakan agar daftar konfigurasi tetap bersih dan terorganisir. Fitur *delete* ini dilengkapi dengan mekanisme konfirmasi sehingga menghindari penghapusan data secara tidak sengaja. Dengan cara ini, konsistensi dan keamanan data tetap terjaga dalam proses pengelolaan konfigurasi *email*. Kemudian, alur pengembangan halaman *Email Setting* digambarkan pada Gambar 3.78.



Gambar 3.78. Flow pembuatan halaman Email Setting

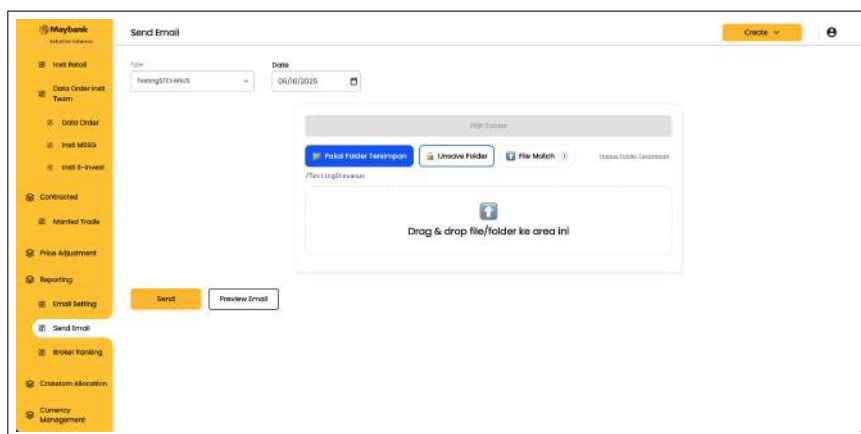
Alur ini dimulai dari pembuatan *EmailSettingList.tsx*, *EmailSettingCreate.tsx*, *EmailSettingShow.tsx* dan *EmailSettingEdit.tsx* komponen halaman. Kemudian, seluruh komponen halaman tersebut dipanggil di dalam file *index.tsx* dengan *setting list*, *create*, *show* dan *edit*. Tahap berikutnya sama seperti menu lainnya, file *index.tsx* dipanggil kedalam *App.tsx*. Pembuatan *table*

dilakukan menggunakan *mui-material* dan diintegrasikan dengan *useGetList* supaya mendapatkan data dari *endpoint api* yang diberikan oleh *backend*.

Selanjutnya, pembuatan tombol *show*, *edit* dan *delete* pada halaman *list*, dilanjutkan pembuatan halaman *show* dan *edit* yang memiliki tampilan mirip dengan halaman *create*. Tahap terakhir, dilakukan integrasi dengan *backend* menggunakan *custom fetchJson*, sama seperti halaman lainnya. Dengan alur tersebut, proses implementasi menu *Email Setting* dapat berjalan secara sistematis dan konsisten dengan modul-modul lainnya.

I.11 Menu Send Email

Berikutnya, menu *Send Email* disediakan untuk mendukung proses pengiriman laporan secara manual oleh *user*. Menu ini menggunakan tipe dari *Email Setting* yang telah dikonfigurasi oleh *user*. Menu ini memungkinkan *user* memilih tipe laporan, menentukan tanggal, dan mengunggah *file* atau *folder* laporan yang ingin dikirimkan. Selain itu, *user* juga dapat melihat isi *email* sebelum dikirimkan melalui fitur *preview*. Tampilan awal menu ini dapat dilihat pada Gambar 3.79.



Gambar 3.79. Hasil implementasi halaman Send Email

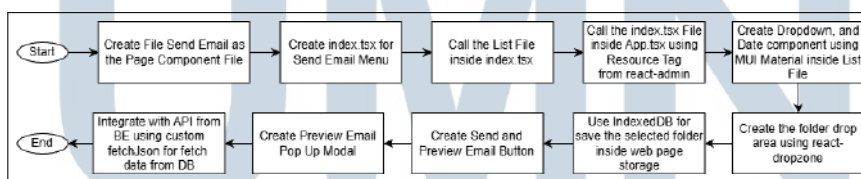
Pada halaman utama, *user* dapat memilih tipe laporan, menentukan tanggal laporan, serta mengunggah *folder* melalui fitur *drag and drop*. Sistem juga menyediakan opsi untuk menggunakan *folder* yang telah tersimpan sebelumnya di dalam *IndexedDB*, sehingga *user* tidak perlu melakukan *upload* ulang pada penggunaan tipe yang sama berikutnya. Tombol *Send* dan *Preview Email* disediakan agar *user* dapat mengirim laporan secara langsung atau memeriksa

kembali isi *email* sebelum dikirim. Selanjutnya, *user* dapat melihat isi *email* secara lengkap melalui fitur *preview* yang ditampilkan pada Gambar 3.80.



Gambar 3.80. Hasil implementasi preview Send Email

Pada halaman *preview*, sistem menampilkan daftar penerima *email* (To, CC, dan BCC), subjek pesan, isi badan *email*, serta daftar *file attachment* yang akan dikirimkan. Dengan adanya tampilan ringkas seperti ini, *user* dapat memastikan bahwa seluruh informasi telah benar sebelum menekan tombol *Send*. Fitur ini sangat membantu untuk meminimalkan kesalahan pengiriman dan memastikan laporan dikirim kepada pihak yang tepat dengan lampiran yang sesuai. Tahapan pengembangan dan alur proses pembuatan fitur *Send Email* divisualisasikan pada Gambar 3.81.



Gambar 3.81. Flow pembuatan halaman Send Email

Flow tersebut menunjukkan bahwa pengembangan fitur dilakukan secara bertahap, dimulai dari pembuatan *file SendEmail.tsx* sebagai komponen utama dan komponen *dropdown*, *date picker*, serta area unggah file berbasis *react-dropzone*. Selanjutnya, fitur diintegrasikan dengan *API backend* untuk mengambil data konfigurasi penerima *email* dan mengeksekusi proses pengiriman. Implementasi *IndexedDB* digunakan untuk menyimpan *folder* laporan pada *storage website* agar dapat digunakan kembali tanpa harus mengunggah ulang, sehingga dapat meningkatkan efisiensi proses operasional harian. Dengan alur yang sistematis

ini, fitur *Send Email* mampu bekerja secara optimal dan konsisten dengan modul lainnya.

I.12 Menu Broker Ranking

Menu *Broker Ranking* dikembangkan untuk memudahkan tim internal dalam memantau performa masing-masing *broker* berdasarkan aktivitas transaksi harian. Fitur ini menyajikan tampilan data yang ringkas namun tetap informatif, sehingga *user* dapat dengan mudah memahami pergerakan volume, nilai total transaksi, hingga frekuensi perdagangan dari setiap *broker*. Selain itu, proses pemilihan tanggal sebagai parameter penyaringan data dapat berjalan dengan lancar tanpa mengganggu tampilan keseluruhan. Hasil implementasi halaman utama ditunjukkan pada Gambar 3.82.

| No | Code | Total Value | Total Volume | Total Frequency |
|----|------|-------------|--------------|-----------------|
| 1 | X | 9,194.7 | 19,917 | 1,219 |
| 2 | Y | 3,574.0 | 11,200 | 451 |
| 3 | C | 3,187.4 | 11,000 | 391 |
| 4 | M | 2,595.0 | 10,311 | 380 |
| 5 | Y | 1,719.90 | 9,246 | 100 |
| 6 | X | 899.90 | 9,044 | 100 |
| 7 | C | 1,602.50 | 5,200 | 200 |
| 8 | A | 1,404.00 | 3,000 | 30 |
| 9 | A | 3,973.2 | 4,899 | 240 |
| 10 | X | 1,055.0 | 4,571 | 373 |
| 11 | P | 3,463.71 | 4,186 | 359 |
| 12 | Z | 3,254.7 | 3,000 | 100 |
| 13 | K | 766.04 | 2,44 | 61 |
| 14 | A | 349.99 | 2,2 | 10 |
| 15 | L | 1,700.00 | 2,00 | 10 |
| 16 | A | 791.94 | 2,00 | 40 |
| 17 | S | 900.00 | 1,99 | 84 |
| 18 | B | 2,200.0 | 1,90 | 84 |
| 19 | Y | 430.00 | 1,80 | 100 |
| 20 | H | 899.8 | 1,70 | 100 |
| 21 | B | 49.80 | 1,50 | 70 |

Gambar 3.82. Hasil implementasi halaman Broker Ranking

Halaman ini menampilkan daftar peringkat *broker* berdasarkan nilai transaksi, volume perdagangan, serta frekuensi aktivitas yang berhasil diolah dari laporan harian. Data ditampilkan secara berurutan dari peringkat tertinggi sehingga memudahkan proses analisis performa *broker*. Tampilan tabel yang luas memungkinkan *user* untuk melihat seluruh parameter pada satu layar sekaligus tanpa perlu membuka halaman tambahan. Selain itu, integrasi dengan *filtering* tanggal memungkinkan *user* untuk melihat data secara historikal. Selain melihat data langsung, *user* juga dapat melakukan *download report* dalam format yang siap diproses lebih lanjut. Untuk memenuhi kebutuhan tersebut, sistem menyediakan fitur *Export Broker Ranking* yang memungkinkan *user* mendapatkan *file* laporan secara cepat hanya melalui satu klik. Hasil implementasi fitur ekspor dapat dilihat

pada Gambar 3.83.

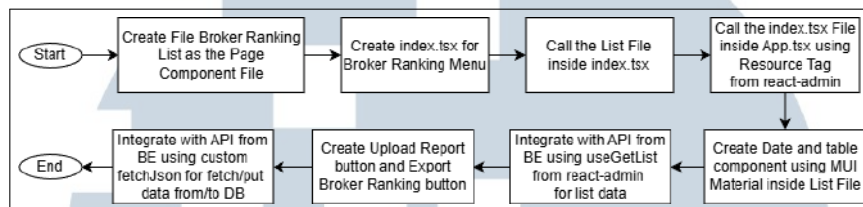
| No | Broker | Name | Buy | BuyV | BuyF | Sell | SellV | SellF | Net | NetV | NetF | Total | YTD |
|----|--------|--|-------|-------|------|-------|-------|-------|-----|---------|-------------------|-------|---------------|
| 1 | 1 | Miras Asset Sekuritas Indonesia | 1,149 | 2,241 | 121 | 1,207 | 2,290 | 191 | 191 | 255,494 | 2,205,188,144,220 | 0.00 | 4,371,91,574 |
| 2 | 2 | MANDIRI SECURITIES | 1,749 | 2,165 | 160 | 1,589 | 1,941 | 162 | 162 | 214,267 | 2,202,960,020,000 | 0.00 | 4,101,402,961 |
| 3 | 3 | KB Sekuritas | 369 | 2,162 | 54 | 315 | 1,846 | 17 | 174 | 117,068 | 1,619,651,091,760 | 0.00 | 3,716,368,063 |
| 4 | 4 | Tanahgajah Sekuritas Indonesia Tbk. | 499 | 2,040 | 56 | 1,984 | 1,990 | 66 | 66 | 141,884 | 1,606,923,074,760 | 0.00 | 3,567,485,064 |
| 5 | 5 | CBS International Sekuritas Indonesia | 861 | 1,556 | 49 | 1,507 | 1,458 | 48 | 215 | 81,389 | 1,479,028,214,840 | 0.00 | 3,360,372,708 |
| 6 | 6 | Semesta Indonesia Sekuritas | 385 | 1,211 | 46 | 1,165 | 1,170 | 51 | 97 | 107,877 | 1,374,078,033,000 | 0.00 | 3,249,646,000 |
| 7 | 7 | USG Sekuritas Indonesia | 1,144 | 1,000 | 100 | 1,044 | 1,044 | 48 | 48 | 164,683 | 2,201,973,819,000 | 0.00 | 3,050,609,000 |
| 8 | 8 | Indo Premier Sekuritas | 525 | 920 | 100 | 771 | 1,048 | 148 | 148 | 201,280 | 1,339,023,091,272 | 0.00 | 3,000,000,000 |
| 9 | 9 | Maybank Sekuritas Indonesia | 809 | 847 | 42 | 765 | 1,044 | 15 | 15 | 200,209 | 1,040,187,000,000 | 0.00 | 2,870,000,000 |
| 10 | 10 | Asah Sekuritas Asia | 527 | 875 | 146 | 729 | 1,044 | 158 | 158 | 278,414 | 1,445,16,523,700 | 0.00 | 2,858,383,000 |
| 11 | 11 | BN Sekuritas | 316 | 800 | 42 | 758 | 1,044 | 17 | 17 | 98,837 | 1,223,890,865,800 | 0.00 | 2,800,000,000 |
| 12 | 12 | J.P. Morgan Sekuritas Indonesia | 475 | 426 | 25 | 401 | 1,044 | 11 | 11 | 481,239 | 963,746,362,000 | 0.00 | 2,812,224,000 |
| 13 | 13 | Philips Sekuritas Indonesia | 229 | 411 | 201 | 210 | 1,044 | 101 | 101 | 82,031 | 476,713,868,500 | 0.00 | 2,668,111,011 |
| 14 | 14 | Korea Investment and Sekuritas Indonesia | 142 | 407 | 25 | 382 | 1,044 | 10 | 10 | 17,281 | 266,762,781,000 | 0.00 | 2,646,000,000 |
| 15 | 15 | Bucer Sekuritas | 264 | 401 | 27 | 374 | 1,044 | 10 | 10 | 86,789 | 382,509,344,000 | 0.00 | 2,604,444,243 |
| 16 | 16 | Citidana Sekuritas Asia | 107 | 376 | 6 | 370 | 1,044 | 11 | 11 | 16,872 | 324,896,840,000 | 0.00 | 2,511,508,000 |
| 17 | 17 | MNC Sekuritas | 102 | 376 | 26 | 350 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 18 | 18 | BGA Sekuritas | 202 | 207 | 26 | 180 | 1,044 | 10 | 10 | 10,837 | 303,686,811,400 | 0.00 | 2,510,682,000 |
| 19 | 19 | BN Sekuritas | 216 | 201 | 21 | 195 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 20 | 20 | RHB Sekuritas Indonesia | 102 | 211 | 22 | 189 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 21 | 21 | BN Sekuritas | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 22 | 22 | Yakini Sekuritas | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 23 | 23 | SAMUEL SEKURITAS INDONESIA | 202 | 207 | 26 | 180 | 1,044 | 10 | 10 | 10,837 | 303,686,811,400 | 0.00 | 2,510,682,000 |
| 24 | 24 | PANIN SEKURITAS Tbk. | 123 | 211 | 22 | 189 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 25 | 25 | KGI Sekuritas Indonesia | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 26 | 26 | OCBC Sekuritas Indonesia | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 27 | 27 | SHAMMA SEKURITAS | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 28 | 28 | Daratama Makmur Sekuritas | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 29 | 29 | Leana Andean Sekuritas | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 30 | 30 | Sura Sekuritas Indonesia | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 31 | 31 | Meas Perdana Sekuritas | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 32 | 32 | Dividende Sekuritas | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 33 | 33 | DOMINANT SEKURITAS | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 34 | 34 | USG Kay Man Sekuritas | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 35 | 35 | Pharmaco Sekuritas | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 36 | 36 | CLSA Sekuritas Indonesia | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 37 | 37 | DBS Sekuritas Indonesia | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 38 | 38 | Banka Perdana Sekuritas | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 39 | 39 | Banka Perdana Sekuritas | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 40 | 40 | Banka Perdana Sekuritas | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 41 | 41 | Banka Perdana Sekuritas | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 42 | 42 | Banka Perdana Sekuritas | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 43 | 43 | Banka Perdana Sekuritas | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 44 | 44 | Banka Perdana Sekuritas | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 45 | 45 | Banka Perdana Sekuritas | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 46 | 46 | Banka Perdana Sekuritas | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 47 | 47 | Banka Perdana Sekuritas | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 48 | 48 | Banka Perdana Sekuritas | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 49 | 49 | Banka Perdana Sekuritas | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |
| 50 | 50 | Banka Perdana Sekuritas | 102 | 201 | 21 | 180 | 1,044 | 11 | 11 | 16,789 | 303,686,761,474 | 0.00 | 2,510,682,000 |

Gambar 3.83. Hasil export Broker Ranking

Melalui fitur *export* ini, seluruh data peringkat broker diproses ke dalam format *spreadsheet* sehingga dapat langsung digunakan untuk analisis lanjutan atau kebutuhan pelaporan. Sistem memastikan bahwa angka-angka terkait nilai transaksi, volume, dan frekuensi tersusun dengan benar dan mudah dibaca. Dengan demikian, proses pelaporan harian maupun bulanan dapat dilakukan secara lebih efisien karena *user* tidak perlu lagi melakukan pengolahan data secara manual. Selain itu, pada menu *Broker Ranking* disediakan fitur *Upload Report* yang memungkinkan *user* memasukkan laporan dari pihak eksternal ke dalam *database*. Hasil *upload* tersebut ditunjukkan pada Gambar 3.84.

| RANK | | LEADER | CHALLENGER | | | | | | | | | | |
|---------------------------|-----|---|------------|---------|---------|---------|---------|---------|---------|---------|---------|--|--|
| (1) | | (2) | (3) | | | | | | | | | | |
| MARKET SHARE | | Previous 12 Months | | 2017-12 | 2018-12 | 2019-12 | 2020-12 | 2021-12 | 2022-12 | 2023-12 | 2024-12 | | |
| (1) | | (1) | | (1) | (1) | (1) | (1) | (1) | (1) | (1) | (1) | | |
| TRADING VOLUME OF CLIENTS | | Previous 12 Months | | 2017-12 | 2018-12 | 2019-12 | 2020-12 | 2021-12 | 2022-12 | 2023-12 | 2024-12 | | |
| Online | (1) | (1) | (1) | 16,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | | |
| Branch | (1) | (1) | (1) | 16,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | | |
| Retail | (1) | (1) | (1) | 16,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | | |
| Other | (1) | (1) | (1) | 16,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | | |
| Corporate | (1) | (1) | (1) | 16,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | | |
| Institutional | (1) | (1) | (1) | 16,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | | |
| CAPEX COMPLETION | | Previous 12 Months | | 2017-12 | 2018-12 | 2019-12 | 2020-12 | 2021-12 | 2022-12 | 2023-12 | 2024-12 | | |
| (1) | | (1) | | (1) | (1) | (1) | (1) | (1) | (1) | (1) | (1) | | |
| Online | (1) | (1) | (1) | 16,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | | |
| Branch | (1) | (1) | (1) | 16,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | | |
| Retail | (1) | (1) | (1) | 16,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | | |
| Other | (1) | (1) | (1) | 16,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | | |
| Corporate | (1) | (1) | (1) | 16,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | | |
| Institutional | (1) | (1) | (1) | 16,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | | |
| FINANCIAL RISK MANAGEMENT | | Previous 12 Months | | 2017-12 | 2018-12 | 2019-12 | 2020-12 | 2021-12 | 2022-12 | 2023-12 | 2024-12 | | |
| (1) | | (1) | | (1) | (1) | (1) | (1) | (1) | (1) | (1) | (1) | | |
| Task | (1) | (1) | (1) | 16,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | | |
| Weight | (1) | (1) | (1) | 16,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | | |
| Customer Satisfaction | (1) | (1) | (1) | 16,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | | |
| Score for Benchmarking | (1) | (1) | (1) | 16,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | 17,000 | | |
| NOTES - | | | | | | | | | | | | | |
| (1) | | Must update daily based on Bursa website or any transparent websites. | | | | | | | | | | | |
| (2) | | Must update daily based on the Data Services from ISSI report. | | | | | | | | | | | |
| (3) | | Only update once a month. | | | | | | | | | | | |
| (4) | | Only update once a month for the previous month closing. Can use the last day QRS report for the month. For example if in August their should update July column. | | | | | | | | | | | |
| (5) | | Must update once a month for the previous month closing. For example if in August their should update July column. | | | | | | | | | | | |
| (6) | | Means the last month before the current month. For example if in August then the should be July. | | | | | | | | | | | |

supaya *user* dapat langsung memberikannya kepada pihak terkait. Dengan mekanisme unggah ini, proses pembaruan data menjadi lebih cepat, terstruktur, dan mendukung *audit trail* yang jelas bagi tim operasional. Selanjutnya, alur pengembangan menu *broker ranking* dapat dilihat pada Gambar 3.85.



Gambar 3.85. Flow pembuatan halaman Broker Ranking

Proses dimulai dengan membuat *page component* utama, yaitu *BrokerRankingList.tsx*, kemudian dilanjutkan dengan pembuatan *file index.tsx* yang berfungsi untuk memanggil dan menampilkan komponen tersebut pada aplikasi. Selanjutnya, *file App.tsx* mengimpor *index.tsx* menggunakan *Resource Tag* dari *react-admin* sehingga modul dapat tampil dan terhubung dengan struktur navigasi sistem. Pada Gambar 3.85 juga memperlihatkan bahwa proses pengembangan melibatkan pembuatan *table* data dan komponen tanggal menggunakan *MUI Material*. Setelah struktur halaman lengkap, integrasi *backend* dilakukan menggunakan *useGetList* dan *custom fetchJson* untuk memproses permintaan data. Tahapan berikutnya adalah penambahan tombol *Upload Report* dan *Export Broker Ranking* yang memungkinkan *user* mengunggah laporan dan mengeksport hasil *broker ranking* secara langsung dari *database*. Dengan rangkaian pengembangan yang sistematis ini, modul *Broker Ranking* dapat berjalan dengan stabil, menampilkan data secara akurat, serta memenuhi kebutuhan operasional perusahaan.

I.13 Menu Crosstom Allocation

Halaman *Crosstom Allocation* berhasil diimplementasikan sebagai fitur yang memudahkan proses pengecekan, pengelolaan, dan penyesuaian data alokasi transaksi harian. *Interface* halaman dirancang agar sederhana namun tetap informatif, sehingga *user* dapat memfilter data berdasarkan *client id*, meninjau setiap entri, serta melakukan pembaruan data secara langsung pada tabel. Fitur seperti *Add New Row* dan *Save Changes* menambah fleksibilitas dalam memperbarui informasi tanpa harus berpindah halaman. Selain itu, keseluruhan

tampilan mengikuti standar desain sistem agar konsisten dengan modul lainnya. Hasil akhir implementasi halaman ini dapat dilihat pada Gambar 3.86.

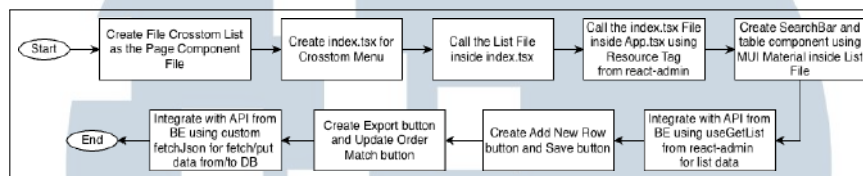
Gambar 3.86. Hasil implementasi halaman Crosstom Allocation

Pada halaman ini, *user* dapat melihat daftar lengkap transaksi yang telah diolah dari data perdagangan harian. Sistem juga memungkinkan *user* untuk menambah baris baru jika terdapat data koreksi atau penyesuaian dari tim terkait. Setelah melakukan perubahan, *user* cukup menekan tombol *Save Changes* untuk menyimpan pembaruan ke *database* melalui *API backend*. Fitur *search* turut disediakan agar proses pelacakan data menjadi lebih cepat dan efisien. Selain dapat melihat dan mengelola data secara langsung, sistem juga menyediakan fitur *Export Crosstom Allocation* yang membantu *user* mendapatkan salinan data untuk kebutuhan analisis lebih lanjut. Fitur ini berguna terutama saat tim membutuhkan rekap transaksi untuk audit internal, rekonsiliasi, maupun pelaporan kepada pihak terkait. Hasil *export* tersebut dapat dilihat pada Gambar 3.87.

| TradeDate | ClientID | BuySell | StockID | Quantity | Price | BoardID | MarketID | BrokerID |
|-----------|----------|---------|---------|----------|-------|---------|----------|----------|
| 09-Sep-25 | 1E | S | MY | 817 | 20 | T | C | Z |
| 09-Sep-25 | 1F | B | MY | 81 | 20 | T | C | Z |
| 09-Sep-25 | 1F | B | MY | 375 | 20 | T | C | Z |
| 09-Sep-25 | 1F | B | MY | 19 | 20 | T | C | Z |
| 09-Sep-25 | 1E | S | MY | 19 | 20 | T | C | Z |
| 09-Sep-25 | 1E | S | MY | 17 | 20 | T | C | Z |
| 09-Sep-25 | 1F | B | MY | 17 | 20 | T | C | Z |
| 09-Sep-25 | 1F | B | MY | 10 | 20 | T | C | Z |
| 09-Sep-25 | 1E | S | MY | 60 | 20 | T | C | Z |
| 09-Sep-25 | 1F | B | MY | 1 | 20 | T | C | Z |
| 09-Sep-25 | 1E | S | MY | 1 | 20 | T | C | Z |
| 09-Sep-25 | 1E | S | MY | 37 | 20 | T | C | Z |
| 09-Sep-25 | 1F | B | MY | 6 | 20 | T | C | Z |
| 09-Sep-25 | 1E | S | MY | 8 | 20 | T | C | Z |
| 09-Sep-25 | 1F | B | MY | 3 | 20 | T | C | Z |
| 09-Sep-25 | 1F | B | MY | 4 | 20 | T | C | Z |
| 09-Sep-25 | 1F | B | MY | 8 | 20 | T | C | Z |
| 09-Sep-25 | 1E | S | MY | 4 | 20 | T | C | Z |
| 09-Sep-25 | 1E | S | MY | 3 | 20 | T | C | Z |

Gambar 3.87. Hasil export Crosstom Allocation

File hasil *export* berisi data lengkap dari tabel *Crosstom Allocation*. Format ini memastikan bahwa seluruh informasi penting yang diperlukan untuk proses rekonsiliasi telah tersedia dalam format siap pakai. Data yang diekspor bersifat konsisten dengan data yang tampil di aplikasi, sehingga meminimalkan risiko ketidaksesuaian dan meningkatkan akurasi proses operasional. Berikutnya, alur pembuatan halaman ini ditampilkan pada Gambar 3.88.



Gambar 3.88. Flow pembuatan halaman Crosstom Allocation

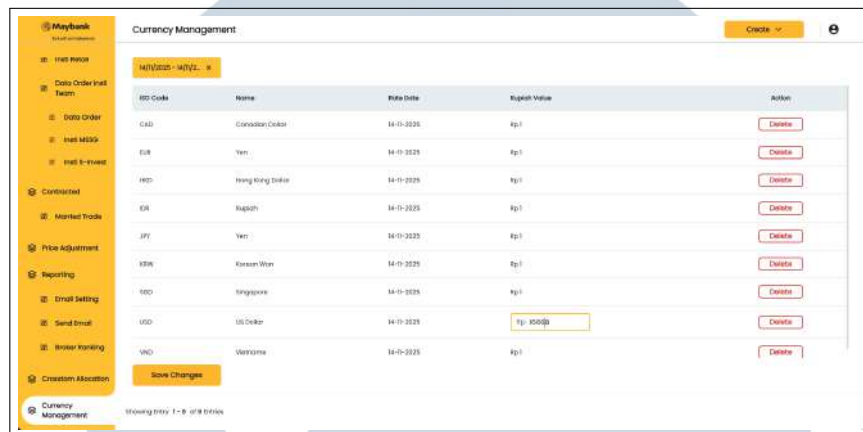
Proses pengembangan halaman *Crosstom Allocation* mengikuti alur kerja terstruktur mulai dari pembuatan *file CrosstomList.tsx* sebagai *page component* utama. Kemudian, *file index.tsx* dibuat untuk memanggil *file CrosstomList.tsx*. *File App.tsx* digunakan untuk memanggil *file index.tsx* supaya dapat ditampilkan pada halaman *website*. Tahapan berikutnya yaitu pembuatan komponen *table* untuk menu ini, untuk komponen *searchbar* menggunakan komponen yang sudah dibuat sebelumnya. Selanjutnya, integrasi dengan *API* dilakukan menggunakan *useGetList* supaya dapat menampilkan data dari respon *api*.

Tahapan berikutnya adalah pembuatan tombol *Add New Row* dan *Save All* dilakukan supaya *user* bisa melakukan *edit* terhadap data yang masih salah atau kurang. Selanjutnya, tombol *export report* dan *update order match* dibuat agar *user* bisa mendapatkan hasil *order match* terbaru dan mendapatkan *report* yang sesuai untuk keperluan operasional perusahaan. Pada tahap terakhir, seluruh interaksi dengan *backend* dilakukan melalui *custom fetchJson*, yang memastikan bahwa operasi *fetch* maupun *put* dapat berjalan sesuai kebutuhan sistem dan alur bisnis yang berlaku.

I.14 Menu Currency Management

Menu *Currency Management* disediakan untuk mengelola nilai kurs harian yang digunakan dalam proses perhitungan transaksi pada sistem *Trade Settlement*. Melalui menu ini, *user* dapat melihat daftar mata uang yang telah dibuat. Selain itu, fitur ini memungkinkan *user* untuk memperbarui nilai kurs secara langsung melalui tabel ataupun menambahkan data kurs baru melalui halaman khusus. Dengan

keberadaan fitur ini, sistem dapat menjaga konsistensi nilai konversi mata uang yang digunakan dalam berbagai laporan dan perhitungan internal. Tampilan utama fitur ini dapat dilihat pada Gambar 3.89.



| ISO Code | Name | Rate Date | Rupiah Value | Action |
|----------|------------------|------------|--------------|--------|
| CAD | Canadian Dollar | 14-11-2025 | Rp.0 | Delete |
| EUR | Yeni | 14-11-2025 | Rp.0 | Delete |
| HKD | Hong Kong Dollar | 14-11-2025 | Rp.0 | Delete |
| KRW | Rupiah | 14-11-2025 | Rp.0 | Delete |
| JPY | Yeni | 14-11-2025 | Rp.0 | Delete |
| KWD | Kuwait Dinar | 14-11-2025 | Rp.0 | Delete |
| SGD | Singapore | 14-11-2025 | Rp.0 | Delete |
| USD | US Dollar | 14-11-2025 | 15,000.00 | Delete |
| VND | Vietnam | 14-11-2025 | Rp.0 | Delete |

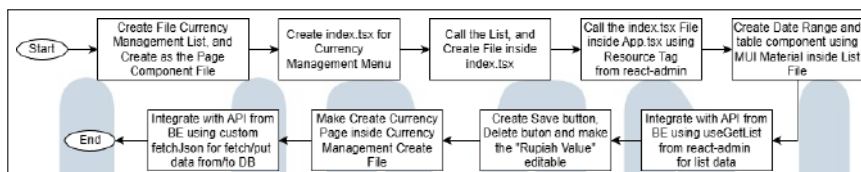
Gambar 3.89. Hasil implementasi halaman Currency Management

Pada halaman utama, *user* dapat menentukan rentang tanggal untuk menampilkan nilai kurs berdasarkan periode tertentu. Setiap hari, daftar mata uang yang telah disimpan akan ditampilkan, dan *user* harus mengisi kolom *rupiah value* yang sesuai dengan kurs tanggal hari itu. Kolom “*Rupiah Value*” dibuat dapat *diedit* secara langsung sehingga *user* dapat melakukan penyesuaian kurs tanpa perlu membuka halaman terpisah. Selain itu, tersedia tombol *Delete* untuk menghapus data kurs dan tombol *Save Changes* untuk menyimpan pembaruan nilai kurs, yang secara otomatis akan diteruskan ke sistem *backend*. Dengan mekanisme ini, proses pemeliharaan data kurs menjadi lebih fleksibel dan efisien. Selanjutnya, disediakan juga halaman *create* untuk menambahkan data kurs baru. Tampilan halaman pembuatan data kurs ditunjukkan pada Gambar 3.90.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Gambar 3.90. Hasil implementasi halaman create Currency Management

Pada halaman ini, *user* dapat mengisi data lengkap mata uang yang akan ditambahkan, seperti *ISO Code*, nama mata uang, nilai kurs rupiah, dan tanggal berlaku. Formulir ini dirancang sederhana dan mudah digunakan agar proses input dapat dilakukan dengan cepat tanpa risiko kesalahan. Setelah seluruh data diisi dan tombol *Submit* ditekan, sistem akan menyimpan informasi tersebut ke dalam *database* dan menampilkannya pada halaman utama. Alur pengembangan fitur *Currency Management* divisualisasikan pada Gambar 3.91.



Gambar 3.91. Flow pembuatan halaman Currency Management

Flow tersebut menggambarkan bahwa proses dimulai dari pembuatan *CurrencyManagementList.tsx* dan *CurrencyManagementCreate.tsx* sebagai komponen halaman. Setelah itu sama seperti menu lainnya, kedua komponen tadi dipanggil di dalam *file index.tsx* yang setelahnya akan digunakan dalam *file App.tsx*. Penggunaan *useGetList* juga digunakan pada halaman ini untuk menampilkan data yang diterima dari *api*. Selanjutnya dibuatlah halaman untuk menyimpan dan menghapus data kurs mata uang, serta dibuat halaman untuk membuat kurs mata uang pada komponen *CurrencyManagementCreate.tsx*. Tahap terakhir dilakukan integrasi *API* untuk mengambil data kurs yang sudah tersimpan serta mengirim perubahan nilai kurs atau data baru ke *backend*. Komponen tabel dirancang menggunakan *MUI Material*, sedangkan input tanggal menggunakan

Date Range Picker agar *user* dapat memfilter data secara tepat. Dengan mekanisme ini, seluruh proses pengelolaan kurs menjadi lebih terstruktur, mudah digunakan, dan terintegrasi penuh dengan sistem transaksi.

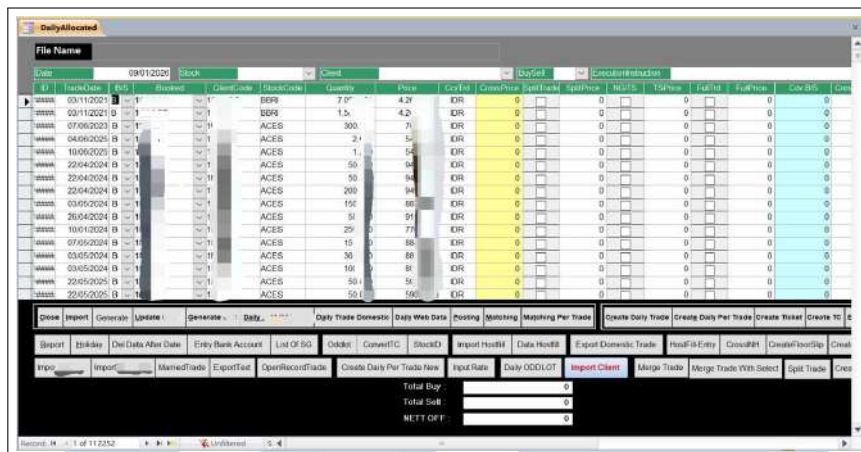
J Komparasi UI Sistem Lama dengan Sistem Baru

Pada bagian ini dijelaskan perbandingan *user interface* antara sistem lama dan sistem baru pada aplikasi *Trade Settlement*. Komparasi dilakukan untuk menunjukkan perubahan dan peningkatan desain tampilan yang berpengaruh terhadap kemudahan penggunaan, kejelasan informasi, serta efisiensi proses kerja *user*. Setiap menu utama akan ditampilkan dalam dua kondisi, yaitu tampilan pada sistem lama dan tampilan pada sistem baru, sehingga perbedaannya dapat diamati secara langsung melalui gambar yang disertakan. Pembahasan difokuskan pada aspek *frontend*, mencakup struktur tampilan, alur interaksi, serta penyederhanaan komponen yang mendukung kebutuhan operasional *user*.

J.1 Menu Data Order

Pada sistem lama, menu *Data Order* masih menggunakan antarmuka berbasis *desktop* dengan tampilan yang cenderung padat dan kompleks yang dapat dilihat pada Gambar 3.92. Seluruh elemen *input*, informasi transaksi, serta tombol aksi ditampilkan dalam satu layar tanpa pemisahan visual yang jelas, sehingga *user* harus memproses banyak informasi secara bersamaan. Struktur tabel pada sistem lama memuat berbagai kolom teknis seperti *Split Trade*, *Full Trade*, dan *Cross Price* yang tidak selalu digunakan dalam setiap aktivitas operasional. Kondisi tersebut berpotensi meningkatkan beban kognitif *user*, khususnya bagi *user* baru atau staf operasional yang hanya berfokus pada proses *input* dan pemantauan data order utama.

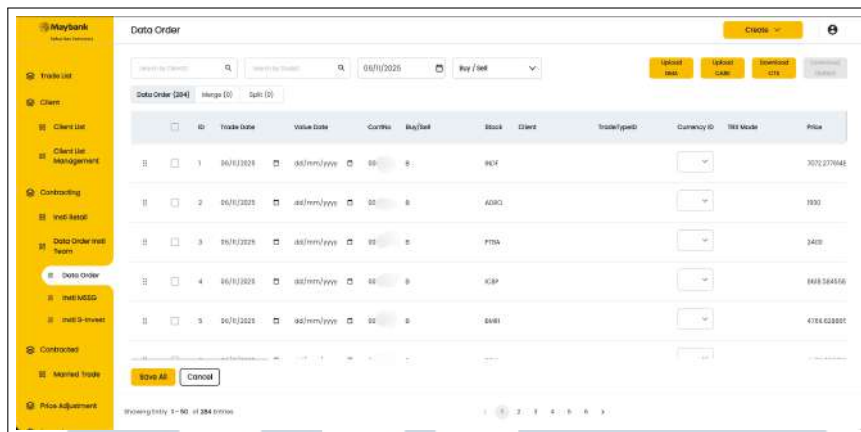
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.92. Tampilan menu Data Order sistem lama

Berdasarkan tampilan tersebut, sistem lama lebih menitikberatkan pada kelengkapan fungsi dibandingkan aspek kenyamanan dan efisiensi *user*. Banyaknya tombol aksi yang ditampilkan dalam satu baris, seperti *Import*, *Generate*, *Matching*, hingga *Create Trade*, berpotensi menimbulkan kebingungan dalam menentukan alur kerja yang sesuai. Selain itu, minimnya indikator visual seperti pengelompokan fitur dan penggunaan warna yang konsisten menyebabkan proses identifikasi fungsi menjadi kurang intuitif. Akibatnya, *user* memerlukan pengalaman serta pemahaman sistem yang cukup mendalam agar dapat mengoperasikan menu *Data Order* secara optimal tanpa meningkatkan risiko kesalahan operasional.

Gambar 3.93 menampilkan menu *Data Order* pada sistem baru yang dirancang dengan pendekatan *interface* berbasis *web* yang lebih modern dan berorientasi pada kemudahan *user*. Informasi dan fitur utama ditata secara terstruktur dengan pemisahan yang jelas antara area pencarian, tabel data, dan tombol aksi. Selain itu, pada sistem terbaru *user* diberikan fleksibilitas untuk mengatur tampilan tabel, di mana setiap kolom dapat diatur posisinya sesuai dengan kebutuhan kerja masing-masing *user*. Penggunaan elemen visual seperti *search bar*, *dropdown filter*, serta fitur *pagination* membantu *user* dalam memfokuskan pencarian data sesuai kebutuhan tertentu. Pendekatan ini menunjukkan adanya peningkatan signifikan terhadap aspek *usability* dan *user experience* dibandingkan dengan sistem sebelumnya.



Gambar 3.93. Tampilan menu Data Order sistem baru

Selain itu, sistem baru menyajikan data transaksi dengan kolom yang lebih relevan dan mudah dipahami, sehingga *user* dapat membaca informasi secara lebih cepat dan akurat. Fitur aksi seperti *Upload DMA*, *Upload CARE*, dan *Download CTX* ditampilkan secara lebih ringkas dan terorganisasi, sehingga mengurangi potensi kesalahan dalam proses operasional. Keberadaan tombol *Save All* dan *Cancel* pada bagian bawah tabel juga memberikan kejelasan alur kerja sebelum data disimpan ke dalam sistem. Secara keseluruhan, sistem baru tidak hanya meningkatkan efisiensi kerja, tetapi juga memberikan pengalaman penggunaan yang lebih intuitif, konsisten, dan selaras dengan kebutuhan operasional yang berjalan saat ini.

J.2 Menu Married Trade

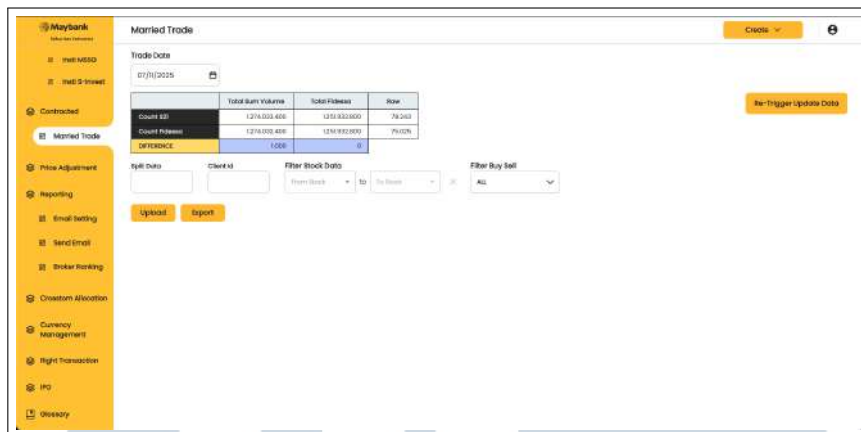
Tampilan menu *Married Trade* pada sistem lama dapat dilihat pada Gambar 3.94. Antarmuka berbasis *desktop* dengan tampilan yang relatif padat dan kurang terstruktur. Seluruh proses pemilihan data, pengecekan kuantitas, serta pembuatan *married trade* dilakukan dalam satu layar dengan banyak tombol aksi yang ditampilkan secara bersamaan. Informasi kuantitas dan perbedaan data ditampilkan secara teknis tanpa penekanan visual yang jelas, sehingga *user* harus melakukan pengecekan secara manual untuk memastikan kesesuaian data. Kondisi ini berpotensi menurunkan efisiensi kerja, terutama ketika *user* harus menangani volume data transaksi yang besar dalam waktu terbatas.



Gambar 3.94. Tampilan menu Married Trade sistem lama

Berdasarkan tampilan tersebut, sistem lama lebih menitikberatkan pada kelengkapan fungsi dibandingkan kemudahan navigasi dan kejelasan alur proses. Banyaknya tombol aksi seperti *Create BO*, *Import*, *Check Different QTY*, hingga *Transfer to Excel* yang ditampilkan dalam satu area berpotensi membingungkan *user* dalam menentukan langkah kerja yang tepat. Selain itu, minimnya pemisahan antara area *input*, proses validasi, dan hasil perhitungan menyebabkan proses pemantauan data menjadi kurang intuitif. Akibatnya, *user* memerlukan tingkat ketelitian yang lebih tinggi untuk menghindari kesalahan dalam proses *married trade*.

Berbeda dengan sistem sebelumnya, sistem baru menghadirkan tampilan menu *Married Trade* yang lebih modern sebagaimana ditunjukkan pada Gambar 3.95. Menu ini dirancang dengan pendekatan antarmuka berbasis *web* yang lebih sederhana, bersih, dan terfokus pada kebutuhan utama *user*. Informasi utama seperti total volume, total data *Fidessa*, serta selisih data ditampilkan dalam bentuk ringkasan tabel yang mudah dibaca. Selain itu, sistem baru menyediakan fitur penyaringan data seperti *filter stock*, *filter buy sell*, dan *client id* yang membantu *user* dalam mempersempit data sesuai dengan kebutuhan analisis. Pendekatan ini menunjukkan adanya peningkatan signifikan pada aspek *usability* dan efisiensi proses kerja.



Gambar 3.95. Tampilan menu Married Trade sistem baru

Selain itu, sistem baru menampilkan perbedaan data transaksi secara lebih terstruktur, sehingga *user* dapat dengan cepat mengidentifikasi adanya selisih antara data internal dan data *Fidessa*. Tombol aksi utama seperti *Upload*, *Export*, dan *Re-Trigger Update Data* ditampilkan secara lebih ringkas dan terpisah, sehingga mengurangi potensi kesalahan dalam pengambilan tindakan. Tata letak antarmuka yang lebih lapang juga membantu *user* dalam memahami alur proses *married trade* secara bertahap. Secara keseluruhan, sistem baru memberikan peningkatan yang signifikan dalam hal kejelasan informasi, efisiensi operasional, serta kenyamanan *user* dibandingkan dengan sistem sebelumnya.

J.3 Menu Client List

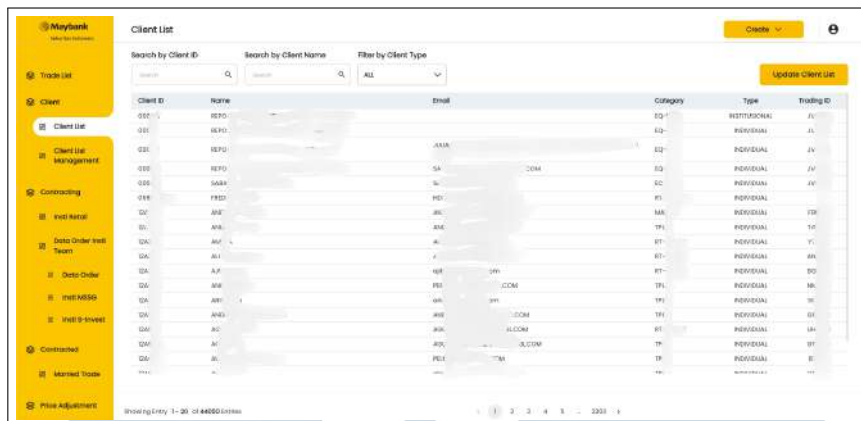
Jika dibandingkan dari sisi pengelolaan data klien, sistem lama pada menu *Client List* masih mengandalkan antarmuka berbasis *desktop*, seperti terlihat pada Gambar 3.96. Proses pencarian dan pengelolaan data klien dilakukan melalui beberapa kolom *input* yang ditampilkan secara terpisah, tanpa adanya fasilitas penyaringan data yang fleksibel. Berbagai tombol aksi seperti *Import Client*, *Update Client*, *Posting*, dan *Export* ditampilkan secara bersamaan dalam satu area, sehingga antarmuka terlihat padat. Kondisi ini menyebabkan *user* harus memahami terlebih dahulu fungsi masing-masing tombol sebelum dapat menjalankan proses pengelolaan data klien secara efektif.



Gambar 3.96. Tampilan menu Client List sistem lama

Berdasarkan tampilan tersebut, sistem lama lebih menekankan pada kelengkapan fungsi dibandingkan dengan kemudahan navigasi dan kenyamanan *user*. Tidak adanya tampilan tabel data klien yang terstruktur membuat *user* harus melakukan pencarian dan verifikasi data secara manual. Selain itu, minimnya pemisahan antara area pencarian, pengolahan data, dan hasil proses berpotensi memperlambat alur kerja, terutama ketika menangani jumlah data klien yang besar. Akibatnya, proses monitoring dan pembaruan data klien memerlukan tingkat ketelitian yang tinggi untuk menghindari kesalahan operasional.

Pada sistem baru, tampilan menu *Client List* mengalami perbaikan yang cukup signifikan, sebagaimana diperlihatkan pada Gambar 3.97. Menu ini dirancang dengan antarmuka berbasis *web* yang lebih modern, bersih, dan terstruktur sehingga memudahkan pengguna dalam mengakses informasi klien. Sistem menyediakan fitur pencarian berdasarkan *Client ID* dan *Client Name*, serta fasilitas *filter by client type* yang memudahkan *user* dalam menyaring data sesuai dengan kriteria tertentu. Data klien ditampilkan dalam bentuk tabel yang rapi dengan kolom-kolom informatif seperti *Client ID*, *Name*, *Email*, *Category*, dan *Type*. Pendekatan ini menunjukkan adanya peningkatan signifikan pada aspek keterbacaan data dan efisiensi proses pencarian informasi.



Gambar 3.97. Tampilan menu Client List sistem baru

Selain itu, sistem baru mendukung pengelolaan data dalam jumlah besar melalui fitur *pagination*, sehingga *user* dapat menavigasi data klien secara bertahap tanpa membebani tampilan layar. Tombol aksi seperti *Create* dan *Update Client List* ditampilkan secara lebih ringkas dan terpisah dari area tabel, sehingga alur kerja menjadi lebih jelas. Tata letak antarmuka yang lebih lapang juga membantu *user* dalam memahami hubungan antara fitur pencarian dan data yang ditampilkan. Secara keseluruhan, sistem baru memberikan peningkatan yang signifikan dalam hal kejelasan informasi, efisiensi pengelolaan data, serta kenyamanan *user* dibandingkan dengan sistem sebelumnya.

3.4 Kendala dan Solusi yang Ditemukan

Pada bagian ini akan diuraikan berbagai kendala yang ditemui, beserta solusi yang diupayakan untuk mengatasinya selama proses kerja magang berlangsung. Pemaparan ini bertujuan untuk memberikan gambaran nyata terhadap tantangan di lapangan serta bagaimana pendekatan yang dilakukan dalam menyelesaikan permasalahan tersebut.

3.4.1 Kendala yang Ditemukan

Berikut ini merupakan beberapa kendala yang dihadapi selama pelaksanaan kerja magang, yaitu:

1. *Transfer knowledge* mengenai logika bisnis dari aplikasi terdahulu cukup sulit karena kurangnya dokumentasi.

2. Komunikasi antar departemen yang terkait cukup sulit karena terdapat proyek lain yang dikerjakan secara paralel dan kesibukkan operasional dari departemen terkait.

3.4.2 Solusi yang Ditemukan

Solusi dari kendala yang dihadapi selama pelaksanaan kerja magang adalah sebagai berikut:

1. Mencoba untuk terus mendalami logika bisnis serta cara kerja dari aplikasi terdahulu, serta selalu bertanya jika terdapat hal yang membingungkan atau kurang jelas pada logika bisnis tertentu pada departemen terkait.
2. Mengadakan pertemuan rutin setiap hari jumat jam 3 sore agar komunikasi antar departemen tetap terjaga dan proyek terus berjalan dengan baik.

