

## BAB 3

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Koordinasi

Kegiatan kerja magang dilaksanakan di PT Dover Chemical yang berlangsung selama satu semester, dengan penempatan sebagai *Website Developer (Full Stack)* pada departemen IT Dover Chemical. Dalam hal ini, Bapak Albert Oktovianus, Bapak Raffy Kamandita dan Ibu Lenny memiliki peran sebagai pengawas dan pembimbing pada kegiatan program kerja magang.

#### 3.2 Tugas yang Dilakukan

Tugas yang dilakukan selama pelaksanaan program kerja magang di Perseroan Terbatas Dover Chemical adalah membuat aplikasi manajemen data utama pengguna.

Aplikasi dikembangkan dengan arsitektur HMVC (*Hierarchical Model–View–Controller*), yaitu pola pengembangan yang memungkinkan setiap modul memiliki struktur tersendiri yang terdiri atas *Model*, *View*, dan *Controller*. Pendekatan ini digunakan dengan tujuan untuk memastikan agar setiap modul dapat dikembangkan secara independen, sehingga membuat *source code* lebih terorganisir dan dapat dikembangkan dengan lebih mudah.

Tugas yang dilakukan selama pelaksanaan program kerja magang, dapat dirinci menjadi beberapa poin sebagai berikut:

- Mempelajari *framework*, dan *library* apa saja yang digunakan dalam pembuatan tampilan *frontend* dan logika aplikasi di belakang layar.
- Membangun tampilan (*frontend*) dan logika aplikasi (*backend*) menggunakan *CodeIgniter* sebagai kerangka bahasa pemrograman, dengan pola arsitektur HMVC.
- Menyusun struktur *database* dan menghubungkannya dengan *Model* pada aplikasi *Customer Management System*.
- Mengembangkan *Controller* untuk mengatur alur logika aplikasi serta menghubungkan *Model* dan *View*.

- Melakukan pengujian fungsi seluruh modul pada aplikasi untuk memastikan setiap menu dan fitur berjalan dengan baik sesuai spesifikasi.
- Melaksanakan uji coba menyeluruh terhadap sistem untuk menjamin bahwa integrasi antara *backend* dan *frontend* bekerja dengan baik.

### 3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

Tabel 3.1. Uraian Tugas Harian Kerja Magang Setiap Minggu Selama Pelaksanaan Kerja Magang

Minggu Ke-	Uraian Pelaksanaan Kerja Magang
1	Memahami sistem <i>Customer Management Website</i> serta mulai mengembangkan metode login untuk pengguna internal, fitur CRUD untuk penambahan customer baru, dan layout sub-tab <i>Primary Information</i> (frontend dan backend).
2	Mengembangkan CRUD untuk sub-tab <i>Directors</i> dan <i>Shareholders</i> menggunakan <i>CodeIgniter</i> , PHP, JavaScript, dan HTML, yang melibatkan integrasi antara frontend dan backend.
3	Melakukan re-konfigurasi CRUD pada penambahan customer baru dengan menambahkan <i>mandatory fields</i> , pemetaan data jenis usaha dan kota, serta mengembangkan CRUD untuk tab <i>Contacts</i> pada sisi frontend dan backend.
4	Mengimplementasikan fitur tampilan <i>NetSuite Group Class View</i> yang terhubung antara frontend dan backend.
5	Membuat CRUD untuk tab <i>Attachments</i> , sehingga sistem dapat mendukung unggah dan manajemen dokumen melalui frontend dan backend.
6	Melakukan perbaikan bug pada sub-tab <i>Banks</i> dan <i>Assets</i> yang ditemukan di sisi backend.
Lanjut ke halaman berikutnya	

**Tabel 3.1: Uraian Pekerjaan Setiap Minggu (lanjutan)**

Minggu Ke-	Uraian Pelaksanaan Kerja Magang
7	Melakukan penyempurnaan fitur <i>Attachments</i> dengan menambahkan manajemen folder berdasarkan <i>customer_id</i> , melakukan peningkatan pada tampilan antarmuka berupa <i>filter</i> dan <i>pagination</i> , serta melakukan presentasi awal terkait usulan pengembangan fitur lanjutan.
8	Melanjutkan pengembangan <i>Customer Detail</i> dengan meningkatkan UI, memperbaiki <i>backend</i> pada sub-tab finansial, menambahkan informasi prospek, serta membuat CRUD untuk sub-tab <i>Major Supplier</i> , <i>Major Customer</i> , dan <i>Credit Limit Information</i> .
9	Membuat CRUD untuk sub-tab <i>General Address</i> dan <i>ERP Group Address</i> , menyempurnakan skalabilitas seluruh sub-tab untuk data berjumlah besar, memperbaiki kendala duplikasi dokumen, serta melakukan penyesuaian alur data <i>credit limit</i> .
10	Mengimplementasikan fitur <i>logging</i> aktivitas pengguna pada seluruh sub-tab, menambahkan pencatatan nilai sebelum dan sesudah perubahan data, serta meningkatkan <i>pagination</i> pada tab alamat dengan penanda alamat terbaru dan pengaturan ulang halaman.
11	Menambahkan <i>date range filter</i> dan <i>ERP group filter</i> pada halaman pelanggan, memperbaiki isu penamaan dokumen dan duplikasi berkas, serta memulai perancangan <i>Sales Activity Dashboard</i> melalui pembuatan indeks database, relasi tabel, dan kueri SQL untuk pengumpulan data.
12	Mengembangkan <i>Sales Activity Dashboard</i> pada sisi <i>backend</i> dan <i>frontend</i> , termasuk pembuatan API, parameter, <i>role-based access control</i> , fungsi validasi, grafik aktivitas (garis dan distribusi status), serta ringkasan aktivitas berdasarkan bulan, kuartal, dan minggu.
Lanjut ke halaman berikutnya	

**Tabel 3.1: Uraian Pekerjaan Setiap Minggu (lanjutan)**

Minggu Ke-	Uraian Pelaksanaan Kerja Magang
13	Menambahkan fitur jumlah aktivitas berdasarkan topik, grafik <i>pie chart</i> , serta tampilan riwayat kunjungan pelanggan. Menambahkan detail tabel riwayat kunjungan, membuat fitur <i>hover detail</i> , dan menyesuaikan kueri utama berdasarkan <i>customer-based activity</i> .
14	Melakukan migrasi layout aplikasi ke <i>AdminLTE</i> termasuk pagination AJAX, sidebar baru, perbaikan UI tabel pelanggan, serta penyesuaian struktur database untuk kompatibilitas penuh dengan aplikasi Dover lainnya.
15	Memperbaiki penanganan nilai <i>null</i> , memperbaiki <i>primary key</i> , menghubungkan aktivitas penjualan dengan tabel <i>m_topic</i> , menambahkan modul unggah dokumen timbangan, memperbaiki tampilan kolom tabel pelanggan, serta melakukan presentasi kedua untuk membahas rencana pengembangan lanjutan.
16	Meningkatkan backend untuk mencegah duplikasi data, memperbaiki akses kontrol <i>group class</i> , memperbarui data pada <i>Primary Information</i> , serta memperbaiki perilaku default sub-tab alamat.
17	Melakukan pembersihan dan pelabelan 6600 baris data pada tabel <i>m_customer_dc</i> , memperbaiki aturan label alamat, serta menyempurnakan modul unggah dokumen.
18	Menyelesaikan revisi akhir data pelanggan dengan menghapus dan menggabungkan data ganda, serta menambahkan data NIB, NPWP 16 digit, dan tanggal pendirian pada sub-tab <i>Primary Information</i> .
19	Penyempurnaan modul pendaftaran transaksi, penambahan kolom <i>alias name</i> , pengembangan sub-tab <i>Others</i> , penambahan kolom ekspor seperti <i>Port Destination</i> , dan dokumentasi peningkatan fitur.
Lanjut ke halaman berikutnya	

**Tabel 3.1: Uraian Pekerjaan Setiap Minggu (lanjutan)**

Minggu Ke-	Uraian Pelaksanaan Kerja Magang
20	Penerapan aturan <i>customer_status</i> baru, pembaruan format tanggal–waktu, dan penghapusan fitur CRUD alamat pada sub-tab <i>ERP Group Address</i> .
21	Implementasi logging penuh untuk seluruh sub-tab, termasuk penyimpanan nilai sebelum dan sesudah perubahan untuk kebutuhan audit trail.
22	Penyesuaian UI formulir tambah/ubah pelanggan pada Primary Information serta validasi alamat berdasarkan aturan bisnis terbaru.
23	Pembuatan struktur modul Approval pada backend (controller dan model) untuk mendukung proses persetujuan perubahan data.
24	Implementasi penuh modul Approval pada halaman Customer dan seluruh tab (pending, delete, active, rejected).
25	Implementasi database penerima email approval dan fitur pengiriman e-mail otomatis (admin dan accounting).

### 3.3.1 Software dan Hardware yang digunakan

Dalam proses pengembangan aplikasi *Customer Management System (CMS)*, diperlukan beberapa perangkat lunak (*software*) dan perangkat keras (*hardware*) untuk mendukung pembuatan aplikasi. Berikut adalah (*software*) dan (*hardware*) yang digunakan pada pengembangan aplikasi ini:

#### A Software

Software yang digunakan selama pengembangan aplikasi *Customer Management System (CMS)* meliputi:

1. Visual Studio Code merupakan editor kode yang bisa digunakan untuk mengelola *source-code* yang dilengkapi dengan dukungan bahasa pemrograman yang luas. Visual Studio Code juga dapat terhubung dengan Github, sehingga dapat membantu pengguna dalam melakukan sinkronisasi versi dengan cepat [9].

2. XAMPP, merupakan perangkat lunak yang digunakan untuk penyediaan server lokal basis data MySQL, sehingga dapat digunakan ketika pihak developer perlu melakukan simulasi komunikasi dengan basis data pada fase produksi [10].
3. GitHub, adalah platform berbasis web untuk pengelolaan kode sumber dengan sistem *version control* menggunakan Git. GitHub digunakan sebagai repositori utama, mendukung kolaborasi tim, *branching*, *pull request*, serta *issue tracking* yang memperlancar proses pengembangan perangkat lunak [11].
4. Seafile merupakan perangkat lunak yang menyediakan jasa penyimpanan berbasis infrastruktur komputasi awan (*cloud*) yang mendukung fitur penyesuaian, penyimpanan, dan pembaruan data secara *real-time* [12].

## **B Hardware**

Perangkat keras yang digunakan oleh pengembang selama proses pembuatan aplikasi *Customer Management System (CMS)* memiliki spesifikasi sebagai berikut:

- Prosesor: Intel I7 Generasi ke-14 dengan 13620H (10 core)
- RAM (*Random Access Memory*): 32 Giga-Byte
- Ukuran Tampilan: Panel monitor berukuran 16 inci
- Media Penyimpanan: SSD 1 Tera-Byte
- Platform Sistem: Windows 11 Pro

### **3.4 Perancangan Aplikasi Manajemen Data Utama Pengguna (*Customer Management System*)**

#### **3.4.1 Analisis Masalah dan Kebutuhan Pengguna**

Pengembangan aplikasi manajemen data utama pengguna menggunakan model *System Development Life Cycle (SDLC)* yang dapat membantu pengembang



untuk melakukan pembuatan rancangan aplikasi dengan siklus pengulangan secara cepat, yaitu RAD *Rapid Application Development*. Model ini dipilih untuk memungkinkan adanya evaluasi yang dilakukan secara cepat dan rutin, sehingga aplikasi yang dirancang dapat benar-benar sesuai dengan kebutuhan pengguna.

#### **A Proses Bisnis**

Proses pengelolaan data pelanggan dilakukan oleh tim *sales* (penjualan). Secara umum, proses pengelolaan data terdiri dari beberapa tahapan. Pertama, tim penjualan akan mendaftarkan pelanggan baru, lalu mengisi data-data pokok yang terkait dengan pelanggan, misalnya seperti alamat, tipe bisnis, lokasi, dan informasi kontak. Proses ini dilakukan secara digital dan menggunakan aplikasi berbasis web.

Setelah mendaftarkan pelanggan baru, tahapan selanjutnya adalah mengisi data-data lanjutan, seperti supplier, aset, informasi bank, jenis kredit yang digunakan, serta direksi dari perusahaan. Selain itu, tim penjualan juga diwajibkan untuk mengupload berbagai data klien yang diperlukan misalnya seperti foto dan berbagai macam dokumen yang penting. Data-data ini kemudian dapat digunakan oleh tim penjualan untuk mempermudah alur transaksi penjualan kepada klien yang datanya sudah terdaftar.

#### **B Analisis Kebutuhan Sistem**

Sistem pengelolaan data yang saat ini digunakan oleh PT Dover Chemical merupakan sistem berbasis *web* yang dikembangkan dengan tujuan untuk memungkinkan tim penjualan menambahkan data secara langsung ke *database*. Meskipun sistem ini berhasil mengatasi kebutuhan transaksi penjualan, terdapat beberapa keterbatasan yang dirasakan, yaitu tim penjualan yang tidak dapat melakukan perubahan data secara langsung, serta belum adanya sistem pemantauan yang terintegrasi langsung dengan data aktivitas tim penjualan. Situasi ini, membuat tim penjualan dan manajemen memiliki beberapa kebutuhan untuk mengatasi keterbatasan yang ada pada sistem lama.

Pertama, tim penjualan membutuhkan sebuah aplikasi yang memungkinkan mereka untuk melakukan pengelolaan dan pemantauan data pada satu aplikasi yang sama. Hal ini dapat membantu tim penjualan untuk memahami progress mereka dalam mendapatkan klien, memantau performa bulanan, serta memungkinkan adanya kolaborasi antar pengguna dalam pengelolaan alamat. Misalnya seperti

penambahan alamat *visit* dan alamat berdasarkan *grouping class* di Netsuite.

Kemudian, pihak manajemen juga memiliki kebutuhan untuk melihat data aktivitas tim penjualan. Hal ini diperlukan dengan tujuan untuk menentukan strategi penjualan yang objektif berdasarkan data, serta memantau performa tim penjualan pada periode waktu tertentu. Selain itu, data historis juga dapat digunakan untuk pendataan pelanggan yang memiliki potensi untuk melakukan transaksi berulang di masa depan.

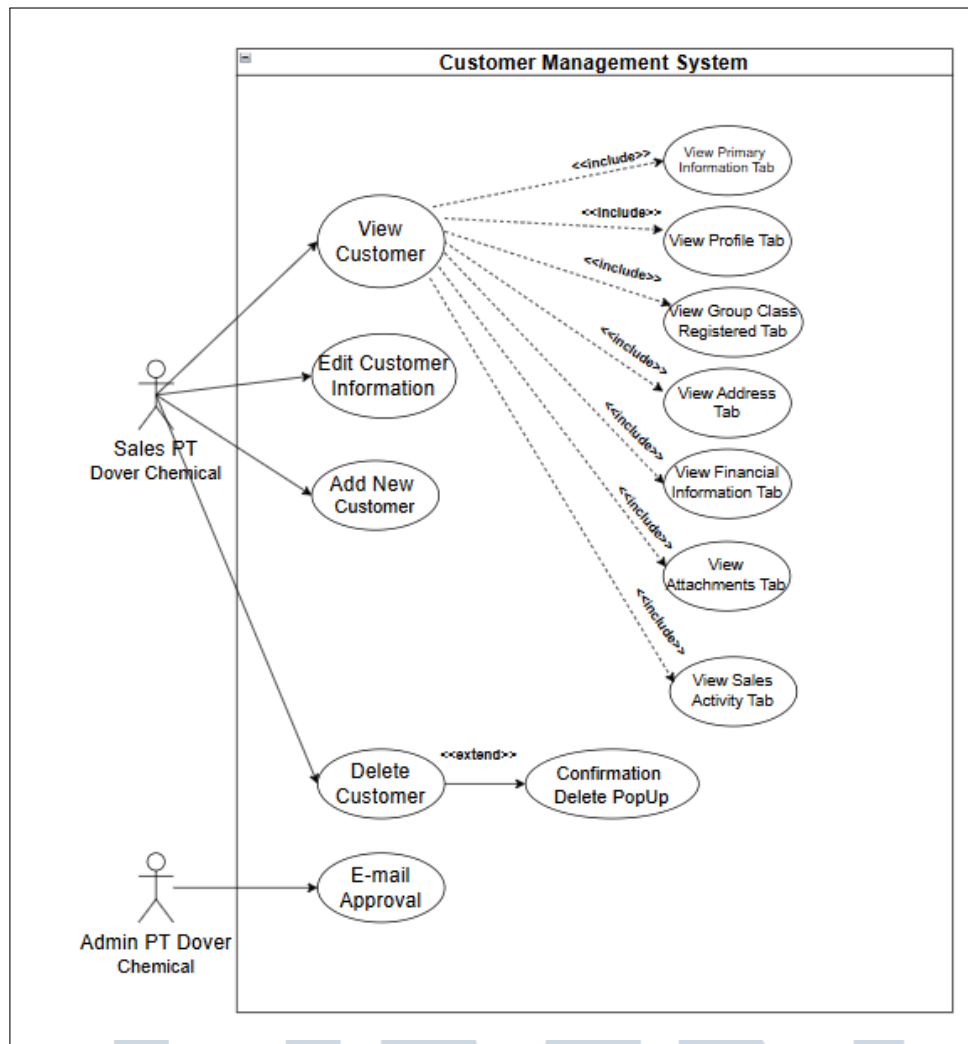
### **3.4.2 Perancangan Sistem**

#### **A Use Case Diagram**

Subbab ini membahas diagram *use case* yang digunakan untuk menggambarkan interaksi antara pengguna dengan aplikasi Customer Management System. Diagram ini berperan sebagai acuan dalam memahami kebutuhan fungsional sistem sebelum proses perancangan dan implementasi dilakukan.





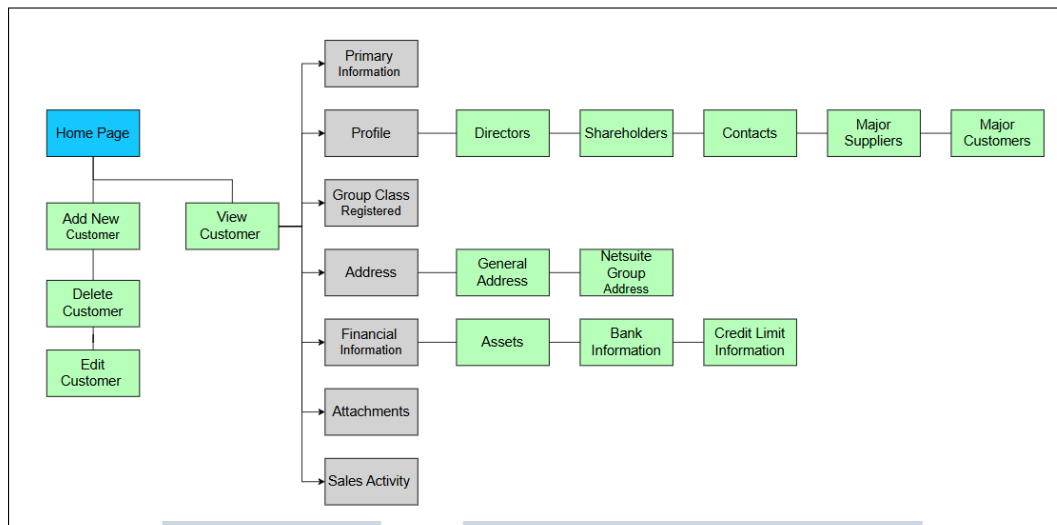


Gambar 3.1. Diagram *Use Case* Aplikasi *Customer Management System*

Dengan adanya diagram *use case*, alur interaksi antara pengguna dan sistem dapat dipetakan dengan terstruktur, sehingga setiap fungsi yang dikembangkan dapat disesuaikan dengan kebutuhan operasional tim sales di PT Dover Chemical.

## B Site Map

Sitemap digunakan untuk memperlihatkan struktur dan alur navigasi dalam aplikasi. Diagram ini menunjukkan bagaimana *user* dapat mengakses suatu *tab* atau *sub-tab* yang saling terhubung. Komponen utama pada sitemap ini mencakup halaman utama, subhalaman, dan hubungan antarhalaman. Sitemap untuk aplikasi *customer management system* dapat dilihat pada Gambar 3.15.



Gambar 3.2. Sitemap Aplikasi *Customer Management System*

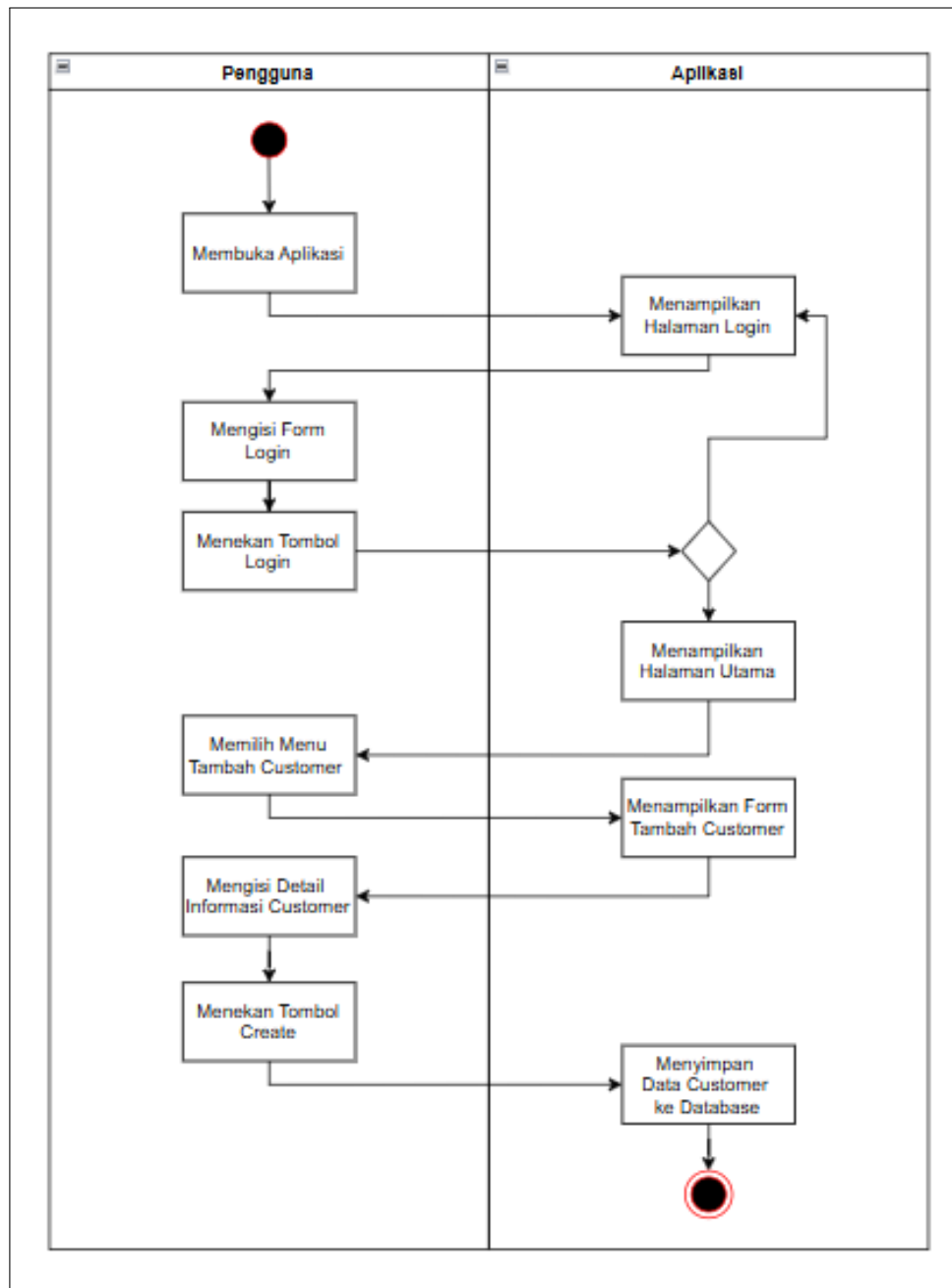
Sitemap yang disusun berfungsi untuk memetakan alur navigasi antarhalaman secara sistematis, sehingga struktur aplikasi dapat dipahami dan proses pengembangan dapat didefinisikan dengan jelas sebelum implementasi pembuatan aplikasi dilakukan.

### C Activity Diagram

Subbab ini membahas *activity diagram* yang digunakan untuk menggambarkan alur aktivitas pengguna dalam menggunakan fitur pada aplikasi *Customer Management System*. Diagram aktivitas berfungsi untuk menunjukkan urutan langkah yang dilakukan oleh *user* serta respon sistem terhadap setiap interaksi yang terjadi.

Gambar 3.3 menunjukkan alur proses ketika pengguna menambahkan data *customer* baru ke dalam aplikasi.

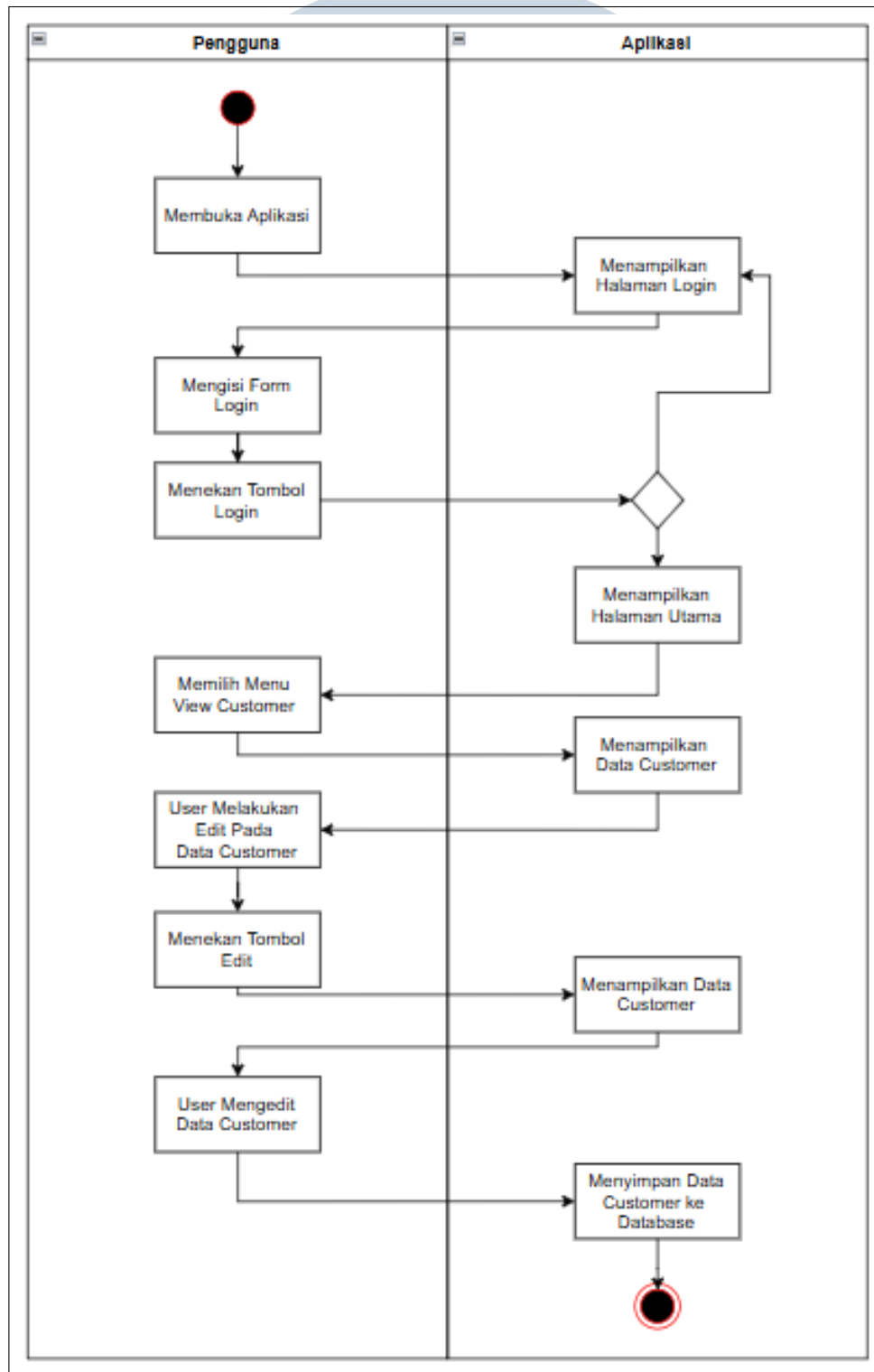
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.3. *Diagram Aktivitas tambah Customer*

Berdasarkan diagram aktivitas tersebut, proses dimulai ketika pengguna memilih fitur tambah *customer*, kemudian mengisi data yang diperlukan. Sistem akan memproses data yang dimasukkan dan menyimpannya ke dalam basis data apabila data valid. Diagram ini membantu dalam memahami alur kerja sistem serta memastikan bahwa proses penambahan data *customer* berjalan sesuai dengan kebutuhan aplikasi.

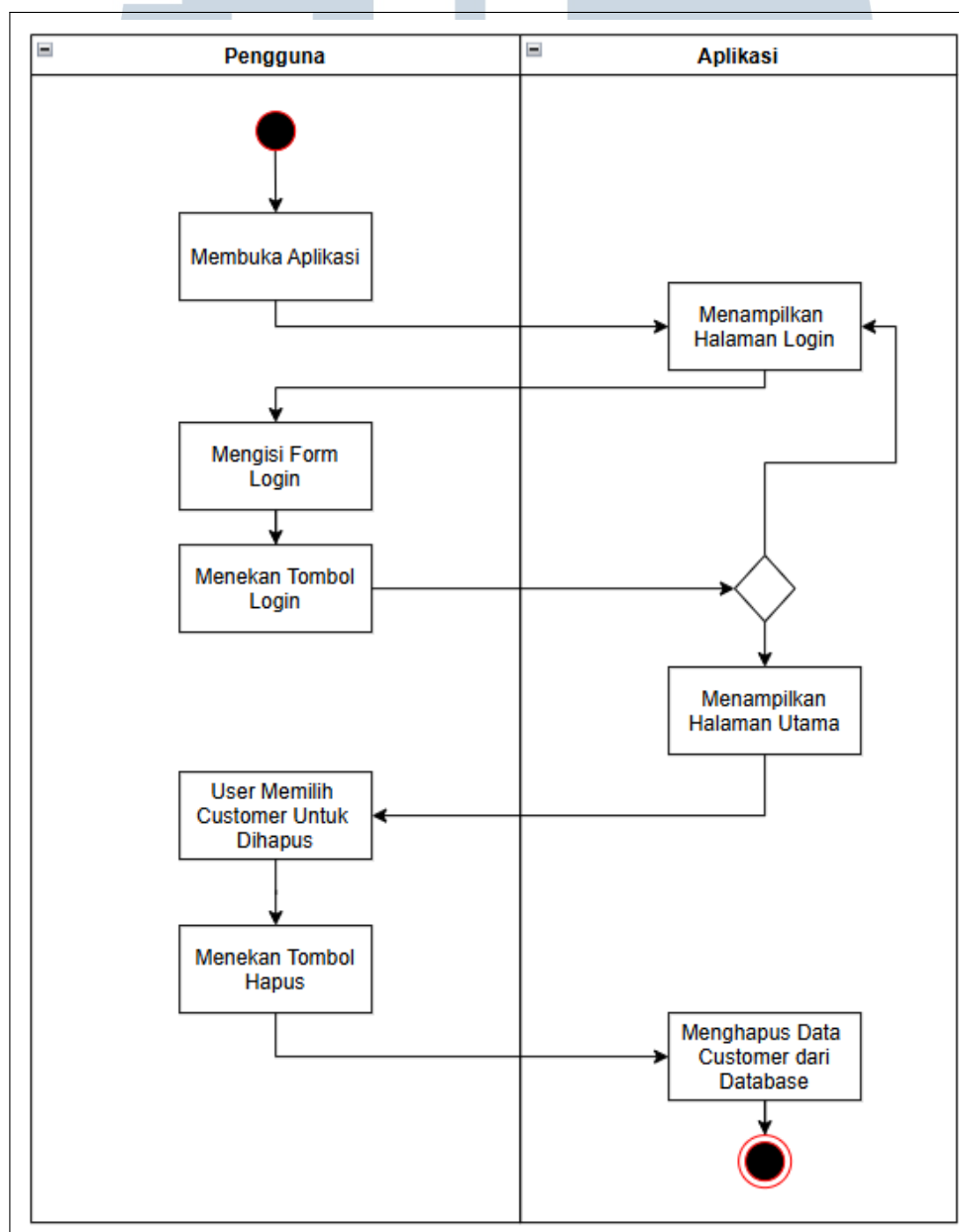
Selanjutnya, Gambar 3.4 menampilkan *activity diagram* proses pengeditan atau memperbarui data *customer* yang sudah ada.



Gambar 3.4. *Diagram Aktivitas Edit Customer*

Pada diagram aktivitas diatas, proses dimulai ketika pengguna memilih data *customer* yang akan diperbarui. Pengguna kemudian melakukan perubahan pada informasi yang diperlukan, dan sistem akan memvalidasi serta menyimpan perubahan data tersebut ke dalam basis data. Diagram ini membantu memastikan bahwa proses pembaruan data *customer* berjalan dengan terstruktur sesuai dengan alur aplikasi.

Terakhir, Gambar 3.5 memperlihatkan diagram aktivitas untuk proses penghapusan data *customer* dari aplikasi.



Gambar 3.5. Diagram Aktivitas Hapus Customer

Berdasarkan diagram aktivitas diatas, proses dimulai ketika pengguna memilih data *customer* yang akan dihapus. Sistem kemudian menampilkan konfirmasi penghapusan untuk memastikan tindakan tersebut dilakukan secara sadar. Setelah pengguna menyetujui, sistem akan menghapus data *customer* dari basis data dan memperbarui tampilan daftar data. Diagram ini berperan penting untuk menghindari kesalahan dalam penghapusan data.

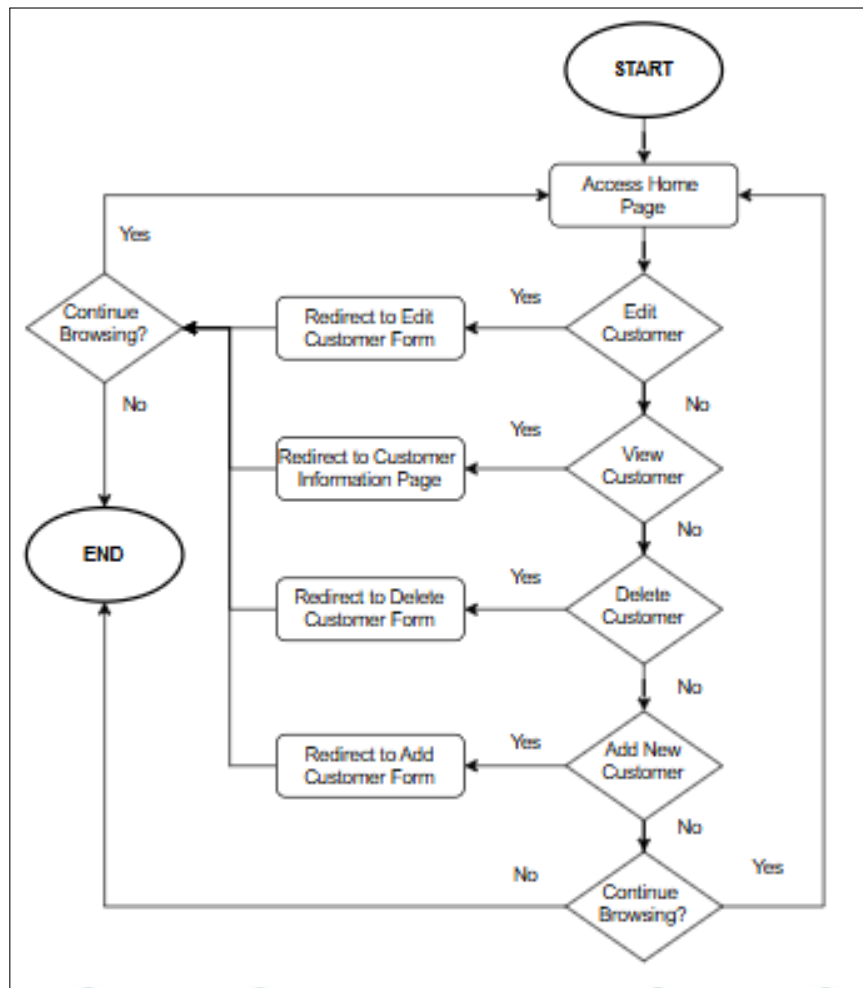
#### **D Flow Chart**

*Flow chart* digunakan untuk menggambarkan alur logika proses dan pengambilan keputusan dalam suatu sistem secara terstruktur. Diagram ini bertujuan untuk memperjelas urutan langkah yang dilakukan oleh *user* maupun sistem, mulai dari awal proses hingga proses tersebut selesai. Dengan adanya *flow chart*, alur kerja pada setiap halaman aplikasi dapat dipahami dengan lebih sistematis.

Pada Gambar 3.6, ditampilkan *flow chart* yang menggambarkan alur proses pada halaman *Home*. Pada halaman ini, *user* dapat melakukan beberapa aksi dalam pengelolaan data pelanggan, yaitu: *edit customer*, *view customer*, *delete customer*, dan *add new customer*.



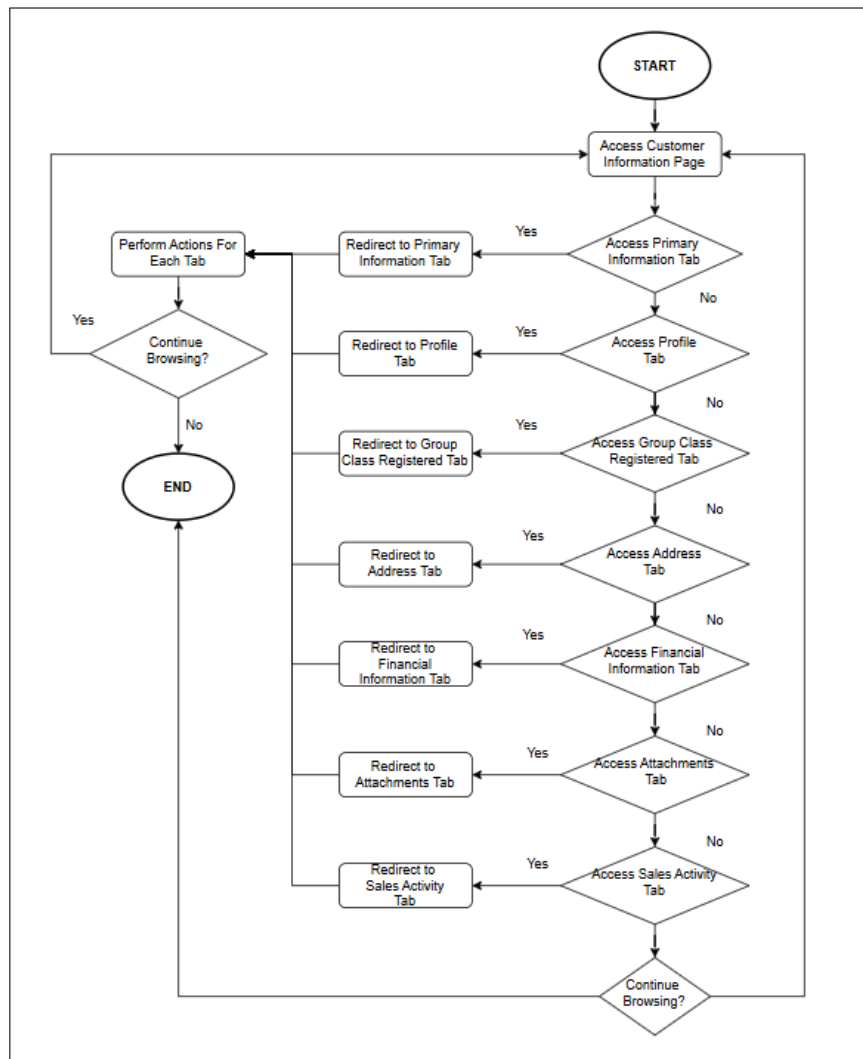




Gambar 3.6. Flow Chart Halaman Home Page

Flow chart 3.6 menunjukkan bahwa pengguna dapat melakukan empat aktivitas di halaman Home, yaitu *edit*, *view*, *delete*, dan *add new customer*. Pada flow chart tersebut, juga ditunjukkan bahwa masing-masing aktivitas akan diarahkan ke tampilan berupa *form* pada masing-masing aktivitas.

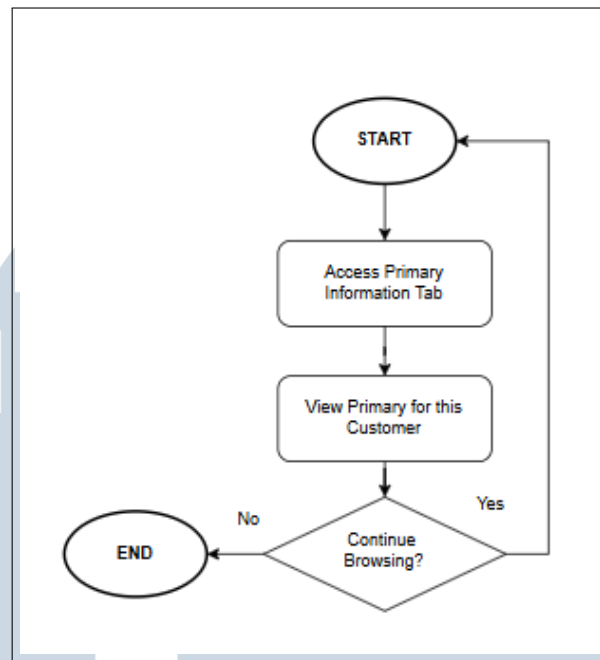
Pada Gambar 3.7, ditampilkan *flow chart* yang menggambarkan alur proses pada halaman *Customer Information* yang bisa diakses ketika *user* menekan tombol *view customer*.



Gambar 3.7. Flow Chart Halaman Customer Information Page

Flowchart diatas menunjukkan bahwa terdapat beberapa *tab* yang bisa diakses oleh user, yaitu : *Primary Information*, *Profile*, *Group Class Registered*, *Address*, *Financial Information*, *Attachments*, dan *Sales Activity*.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

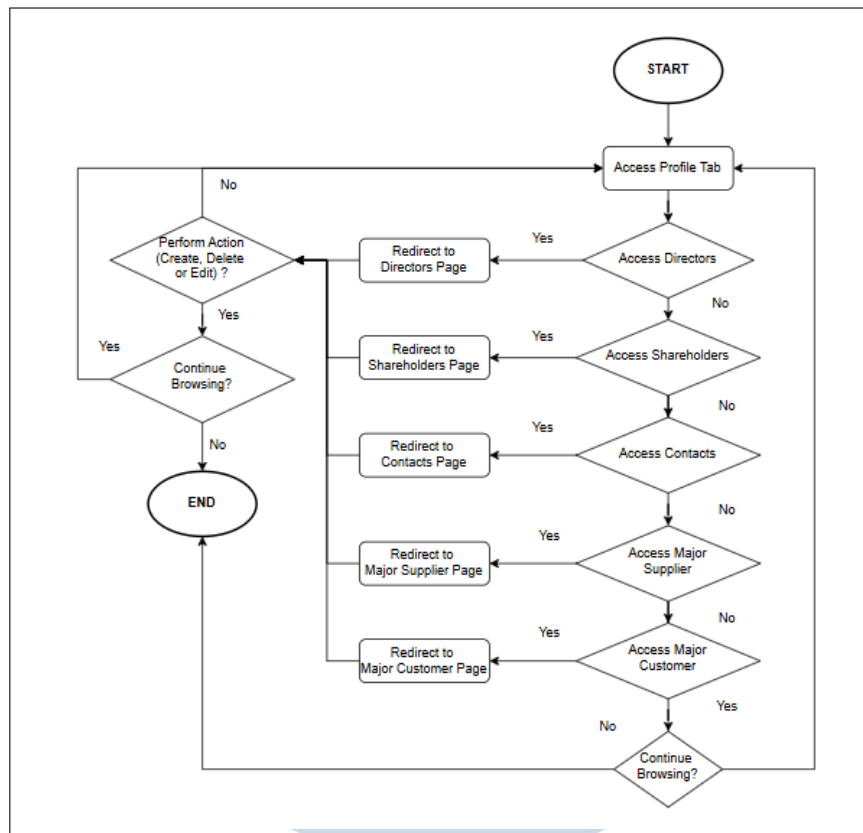


Gambar 3.8. *Flow Chart Primary Information Tab*

Melalui *Primary Information Tab*, user dapat melihat data penting seperti nama pelanggan, alamat, nomor pajak (NPWP), *website*, tanggal pembuatan data, serta informasi utama lainnya. Alur pada *flow chart* ini menunjukkan proses akses data utama pelanggan yang memiliki fitur untuk menampilkan data milik pelanggan, karena terhubung langsung dengan basis data utama pelanggan, alamat pelanggan, serta data terkait surat-surat resmi pelanggan.

Pada Gambar 3.9, ditampilkan *flow chart* yang menggambarkan alur proses pada *Profile Tab*.

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

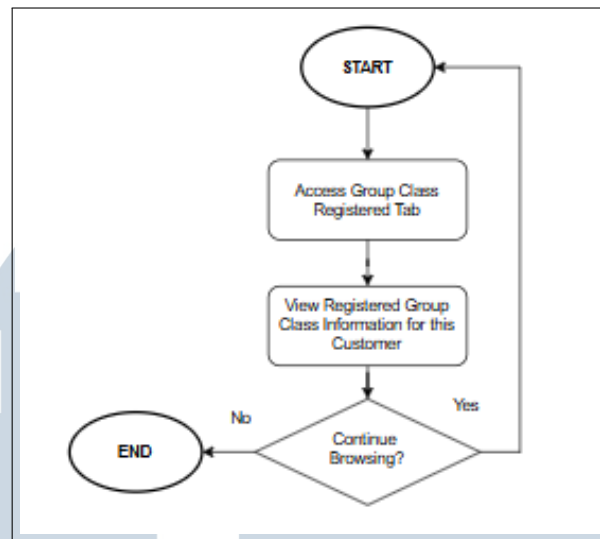


Gambar 3.9. Flow Chart Profile Tab

Pada *tab* ini, terdapat 5 *sub-tab* yang bisa diakses oleh *user*, yaitu: *Directors*, *Shareholders*, *Contacts*, *Major Supplier*, dan *Major Customer*. *User* dapat melakukan aktivitas (*create*, *edit*, dan *delete*) untuk mengisi informasi pada masing-masing *sub-tab*.

Pada Gambar 3.10, ditampilkan *flow chart* yang menggambarkan alur proses pada Group Class Registered Tab.

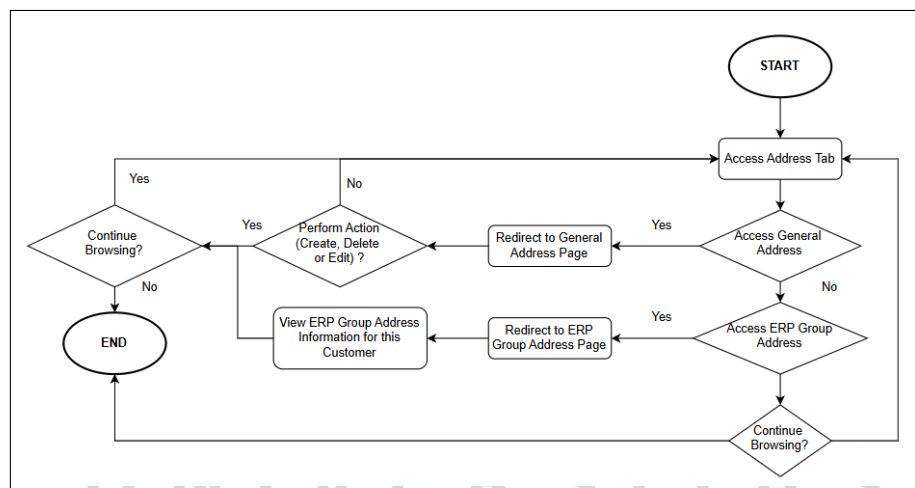
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.10. Flow Chart Group Class Registered Tab

Pada *tab Group Class Registered*, user dapat melihat apakah suatu *customer* sudah terdaftar di suatu *group* tertentu yang terhubung ke data ERP Netsuite.

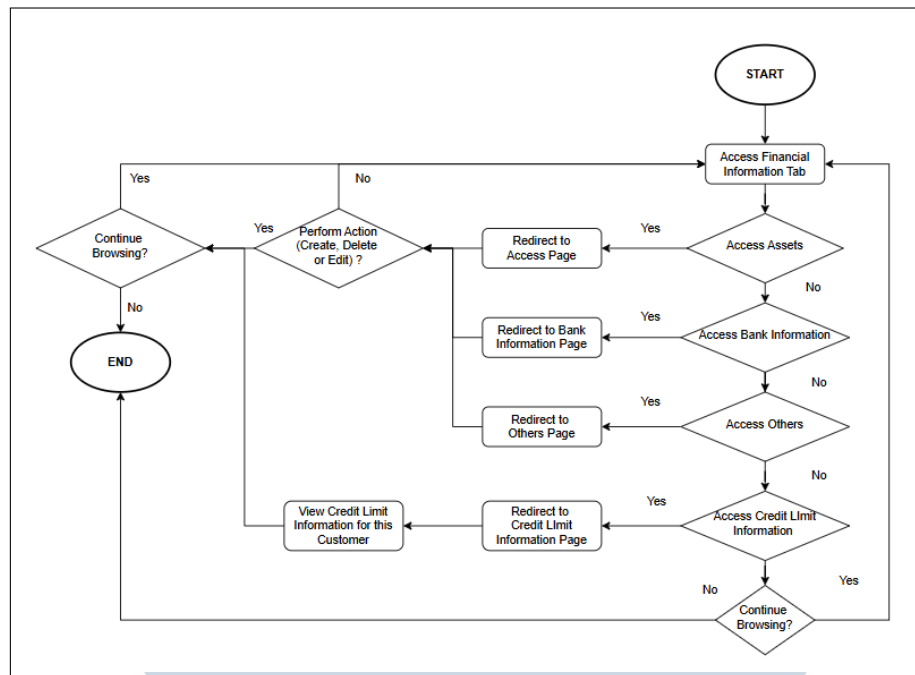
Pada *Address Tab*, user dapat mengakses dua *sub-tab*, yaitu *General Address* dan *ERP Group Address*.



Gambar 3.11. Flow Chart Address Tab

Pada *sub-tab General Address*, user dapat melakukan proses *create*, *edit*, dan *delete* data alamat. Sementara itu, *sub-tab ERP Group Address* bersifat *view only*, sehingga user hanya dapat melihat data alamat yang berasal dari sistem ERP.

Pada Gambar 3.12, ditampilkan *flow chart* yang menggambarkan alur proses pada *Financial Information Tab*.



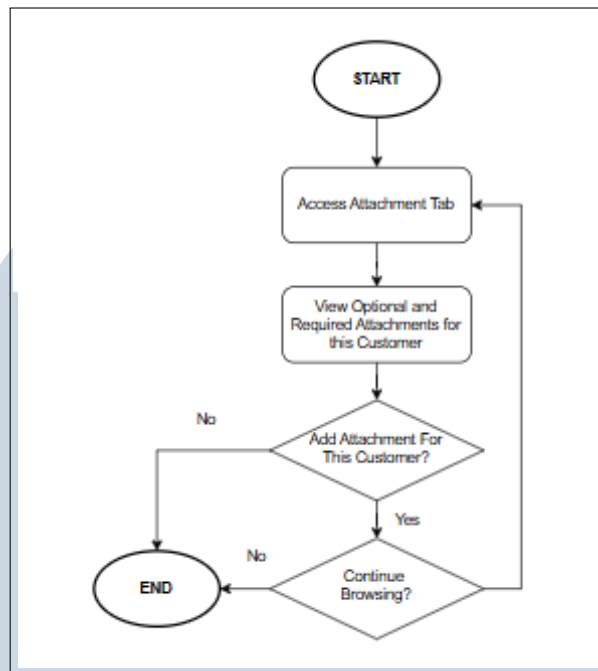
Gambar 3.12. *Flow Chart Financial Information Tab*

Flow chart diatas menggambarkan bahwa terdapat 4 *sub-tab* yang bisa diakses oleh *user* pada *tab* ini, yaitu *Assets*, *Bank Information*, *Credit Limit Information*, dan *Others*.

Pada Gambar 3.13, ditampilkan *flow chart* yang menggambarkan alur proses pada *Attachment Tab*. Terdapat 2 jenis dokumen yang dapat diupload oleh *user*, yaitu dokumen yang bersifat *optional* dan *required*.

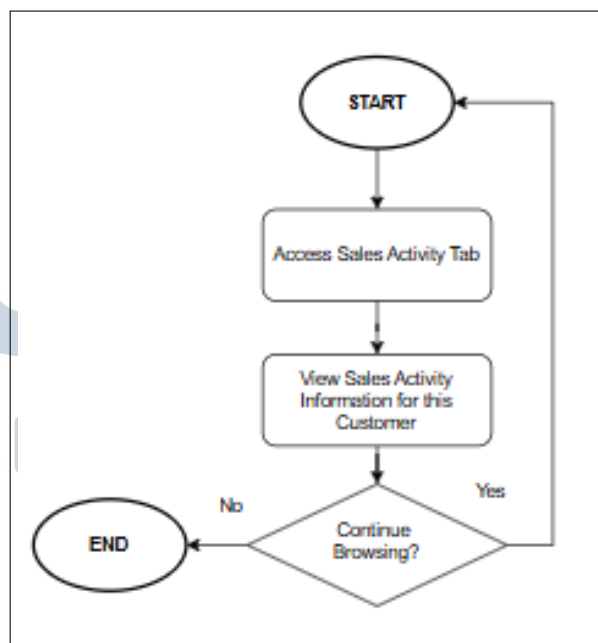
U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A





Gambar 3.13. *Flow Chart Attachment Tab*

Pada Gambar 3.14, ditampilkan *flow chart* yang menggambarkan alur proses pada *Sales Activity Tab*.



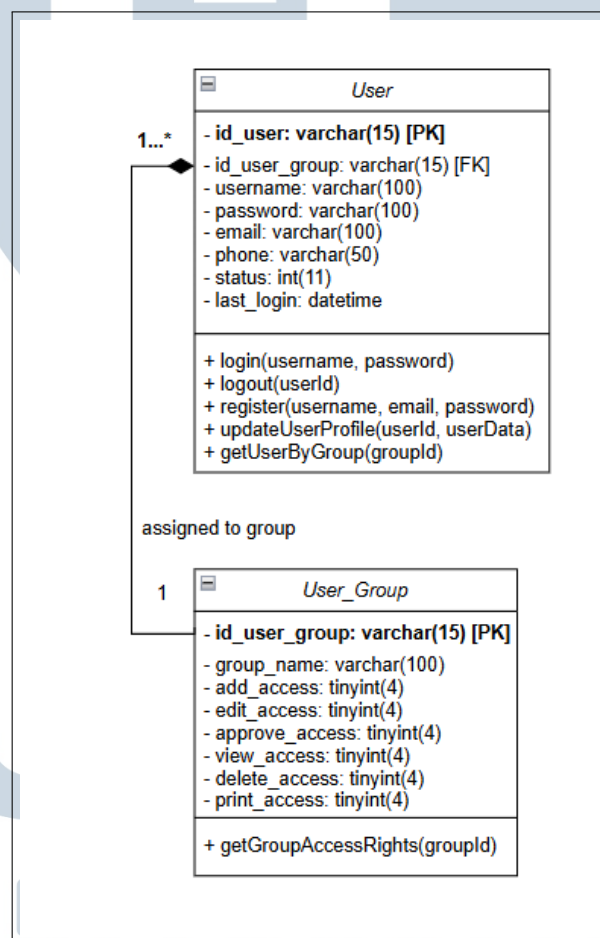
Gambar 3.14. *Flow Chart Sales Activity Tab*

Flow chart diatas menunjukkan *tab* aktivitas tim penjualan, dimana *user*

dapat melihat berbagai informasi histori transaksi penjualan yang disajikan dalam bentuk *dashboard* interaktif.

## E Class Diagram

Pada Gambar 3.15, ditampilkan *class diagram* yang menggambarkan hubungan antara *class User* dan *class User Group*. Diagram ini menunjukkan struktur data serta fungsi-fungsi yang digunakan untuk mengelola informasi pengguna dan pengelompokan hak akses di dalam sistem.



Gambar 3.15. *Class Diagram User dan User Group*

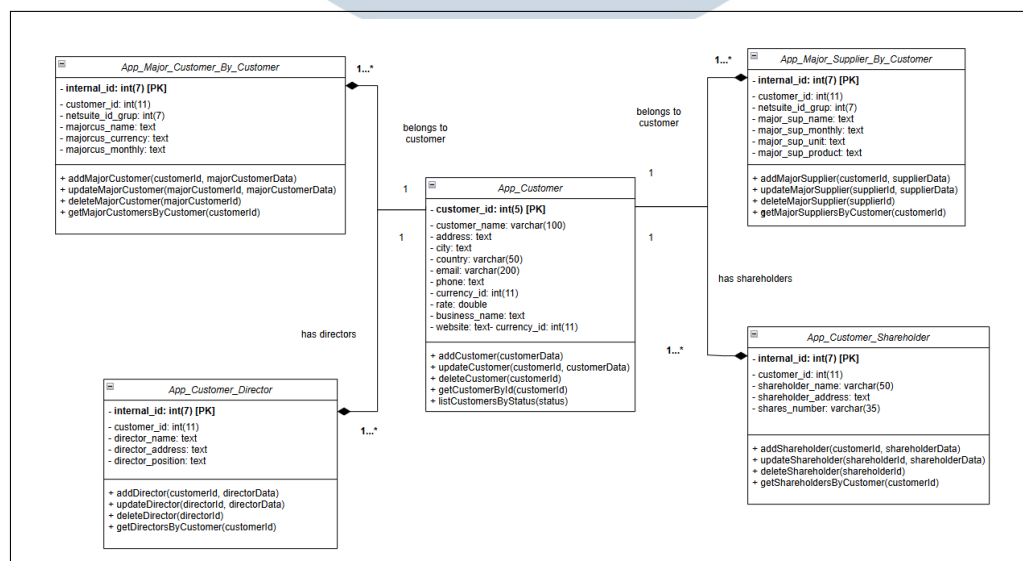
Berikut merupakan penjelasan fungsi dari masing-masing *class* pada diagram di atas:

1. **User**: *Class* ini menyimpan informasi tentang pengguna seperti *username*, *email*, *password*, serta waktu terakhir login. *Class* ini juga memiliki fungsi

untuk melakukan proses autentikasi, seperti *login*, *logout*, dan pembaruan profil pengguna. Setiap pengguna akan terhubung dengan satu *User Group* untuk menentukan hak aksesnya.

2. *User Group*: *Class* ini berfungsi untuk mengelompokkan pengguna berdasarkan peran atau hak akses yang dimilikinya di dalam sistem. Data yang disimpan mencakup nama grup, deskripsi, dan status aktif. Selain itu, terdapat fungsi untuk menambahkan, mengubah, atau menghapus hak akses, serta menampilkan daftar pengguna yang tergabung dalam suatu grup. Satu grup data terdiri dari banyak *customer*.

Pada Gambar 3.16, ditampilkan *class diagram* yang menggambarkan hubungan antara *class Customer* dengan beberapa *class* terkait, yaitu *Major Customer By Customer*, *Major Supplier By Customer*, *Customer Director*, dan *Customer Shareholder*. Diagram ini merepresentasikan struktur data yang digunakan untuk menyimpan serta mengelola informasi profil pelanggan beserta hubungan bisnis dan struktur kepemilikannya di dalam sistem.



Gambar 3.16. *Class Diagram Customer Profile dan Relasinya*

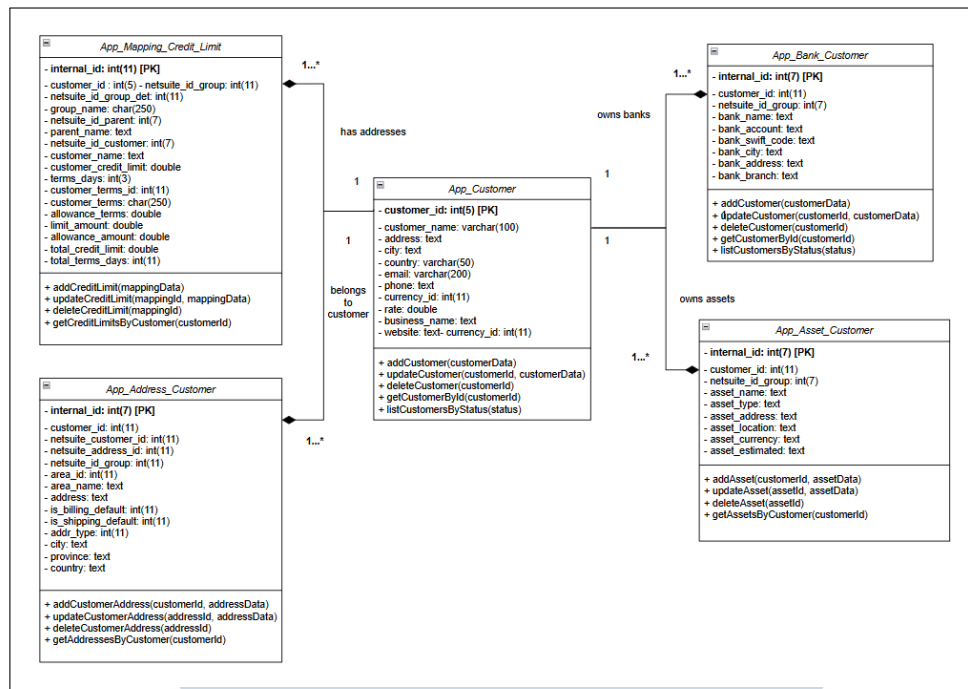
Penjelasan dari masing-masing tabel yang diwujudkan dalam bentuk *class* adalah sebagai berikut:

1. *Customer*: *Class* ini berfungsi sebagai *class* utama yang menyimpan informasi dasar mengenai pelanggan. Data yang dikelola mencakup *customer*

*name, address, city, country*, dan atribut identitas lainnya. *Class* ini juga berperan sebagai penghubung ke berbagai *class* lain yang berelasi, seperti daftar direktur, pemegang saham, pelanggan utama, dan pemasok utama.

2. Major Customer By Customer: *Class* ini digunakan untuk mencatat daftar pelanggan utama (*major customers*) yang berhubungan dengan pelanggan di *class Customer*. *Class* ini memiliki relasi *many-to-one* terhadap *Customer*, menandakan bahwa satu pelanggan dapat memiliki beberapa pelanggan utama.
3. Major Supplier By Customer: *Class* ini berfungsi untuk menyimpan data pemasok utama (*major suppliers*) yang terkait dengan pelanggan tertentu. Hubungan *class* ini dengan *class Customer* bersifat *many-to-one* terhadap *Customer*, yang berarti satu *customer* dapat memiliki beberapa pemasok utama.
4. Customer Director: *Class* ini mencatat informasi mengenai individu yang menjabat sebagai direktur dari suatu pelanggan. Setiap direktur memiliki atribut seperti *name, position*, dan informasi lainnya. Hubungan antara *Customer* dan *Customer Director* bersifat *one-to-many*, sehingga satu *customer* dapat memiliki lebih dari satu direktur yang terdaftar.
5. Customer Shareholder: *Class* ini menyimpan data mengenai pemegang saham dari pelanggan tertentu. Informasi yang disimpan mencakup *name, share percentage*, dan detail kepemilikan lainnya. Hubungan antara *Customer* dan *Customer Director* bersifat *one-to-many*, sehingga satu *customer* dapat memiliki banyak pemegang saham yang terdaftar di dalam sistem.

Pada Gambar 3.17, ditampilkan *class diagram* yang menggambarkan hubungan antara *class Customer* dengan beberapa *class* yang menyimpan data terkait aspek keuangan dan informasi alamat pelanggan. Diagram ini menunjukkan keterkaitan antara data utama pelanggan dengan alamat, aset, rekening bank, serta batas kredit yang dimiliki.



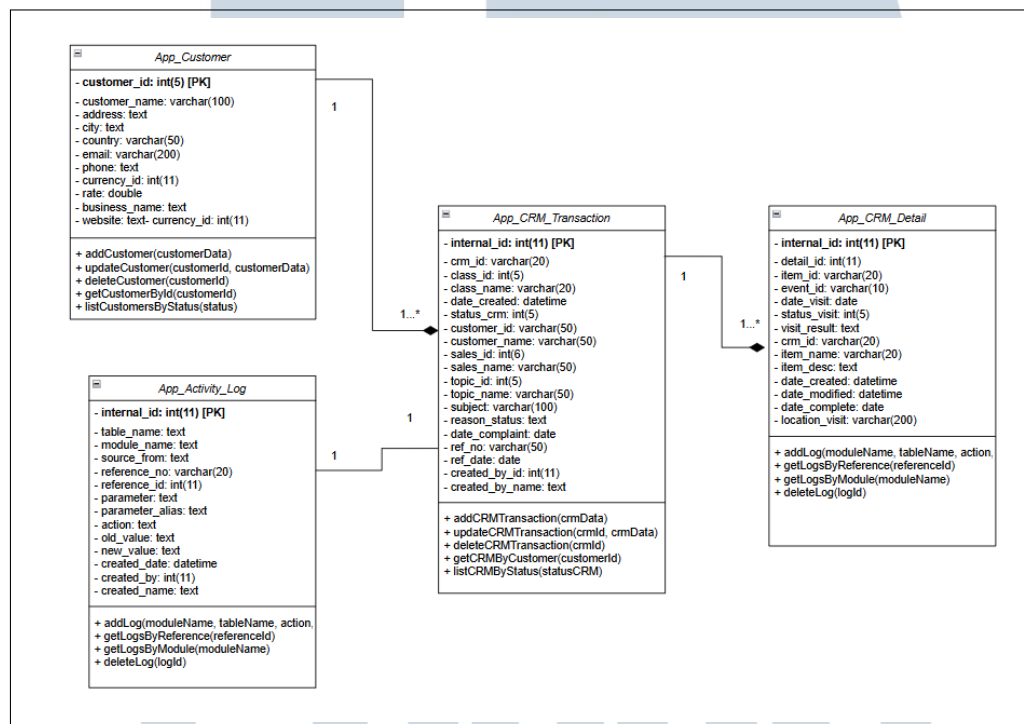
Gambar 3.17. Class Diagram Customer dan Relasi Data Keuangan

Penjelasan dari masing-masing tabel yang diwujudkan dalam bentuk *class* adalah sebagai berikut:

1. Address Customer: *Class* ini menyimpan data alamat pelanggan. Relasinya dengan *Customer* bersifat *many-to-one*, yang berarti satu pelanggan dapat memiliki lebih dari satu alamat, misalnya alamat *visit* dan *billing*.
2. Assets Customer: *Class* ini berfungsi untuk menyimpan seluruh aset yang dimiliki oleh pelanggan. Setiap aset terkait dengan satu pelanggan melalui relasi *many-to-one*. Informasi yang disimpan meliputi jenis aset, nilai aset, lokasi, serta status kepemilikan.
3. Bank Customer: *Class* ini menyimpan informasi mengenai rekening bank yang dimiliki oleh pelanggan. Data yang dikelola mencakup nama bank, nomor rekening, nama pemilik, serta cabang bank terkait. Hubungan antara *Customer* dan *Bank Customer* bersifat *one-to-many*, sehingga satu pelanggan dapat memiliki beberapa rekening di berbagai bank.
4. Netapp Credit Limit: *Class* ini digunakan untuk mencatat batas kredit (*credit limit*) yang diberikan kepada pelanggan. Data yang disimpan mencakup nilai batas kredit, jumlah pemakaian, saldo tersisa, serta informasi kredit lainnya.

Relasinya terhadap *Customer* bersifat *many-to-one*, menandakan bahwa satu pelanggan dapat memiliki lebih dari satu catatan kredit.

Pada Gambar 3.18, ditampilkan *class diagram* yang menggambarkan hubungan antara *class Customer*, *CRM Transaction*, *CRM Detail*, dan *Activity Log*. Diagram ini menjelaskan bagaimana sistem mengelola proses bisnis yang melibatkan pelanggan, mulai dari pencatatan aktivitas penjualan hingga pelacakan detail setiap transaksi yang terjadi.



Gambar 3.18. *Class Diagram CRM Transaction dan Activity Log*

Penjelasan dari masing-masing tabel yang diwujudkan dalam bentuk *class* adalah sebagai berikut:

1. CRM Transaction: Tabel basis data ini menyimpan data utama setiap transaksi CRM, seperti ID transaksi, nama pelanggan, sales, topik, status, serta tanggal pembuatan dan pembaruan. Hubungan antara *Customer* dan *CRM Transaction* bersifat *one-to-many*, yang berarti bahwa satu pelanggan dapat memiliki banyak transaksi CRM. Selain itu, *class* ini juga terhubung ke *CRM Detail* untuk mencatat detail dari aktivitas tim penjualan.
2. CRM Detail: *Class* ini digunakan untuk mencatat detail setiap transaksi CRM, termasuk aktivitas kunjungan pelanggan, hasil kunjungan, catatan



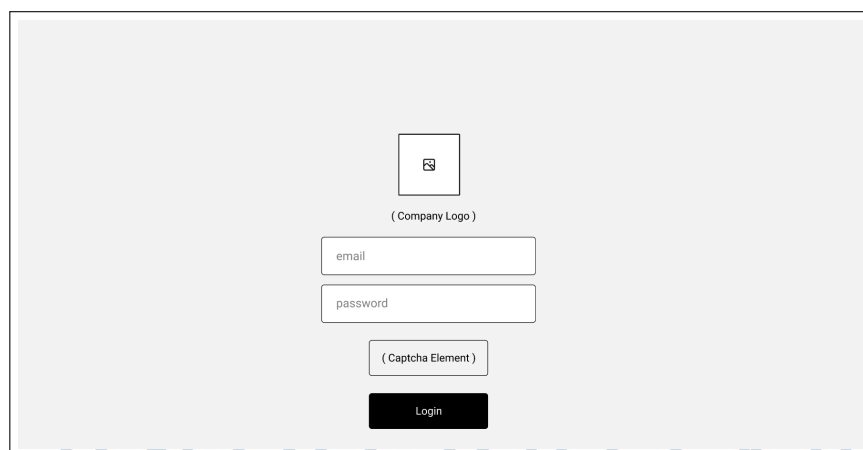
pertemuan, serta waktu penyelesaian. Hubungannya dengan *CRM Transaction* bersifat *one-to-many*, yang berarti bahwa satu transaksi dapat memiliki beberapa detail kegiatan.

3. Activity Log: *Class* ini berfungsi untuk mencatat seluruh aktivitas sistem yang terjadi, termasuk penambahan, pembaruan, atau penghapusan data di berbagai modul. Informasi yang disimpan meliputi nama modul, jenis tindakan, sumber referensi, serta parameter perubahan. Relasinya dengan *CRM Transaction* bersifat *many-to-one*.

## F Mockup Tampilan Aplikasi

Sebelum melakukan pembuatan tampilan dalam bentuk codingan, dilakukan pembuatan (*mockup*) tampilan aplikasi yang bertujuan untuk menentukan struktur utama dari halaman aplikasi yang dibuat. Berikut ini merupakan penjelasan dari contoh tampilan model aplikasi (*mockup*) manajemen data pelanggan.

Gambar 3.19 menunjukkan (*mockup*) dari tampilan halaman Login, yang memiliki elemen berupa logo perusahaan, input field (*e-mail*) dan (*password*) serta elemen autentikasi berupa *captcha*.

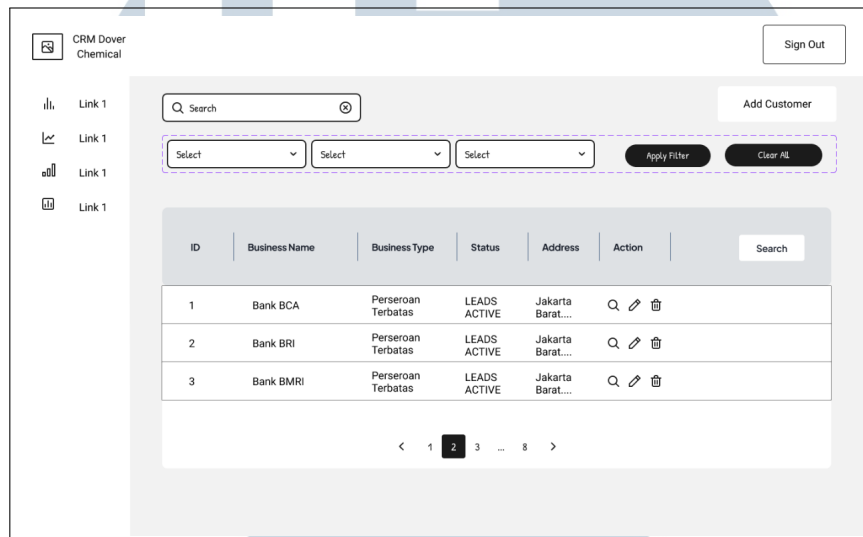


Gambar 3.19. *Mockup Halaman Login Aplikasi Customer Management System*

Halaman login menyediakan fitur autentikasi pengguna, yang dapat dilakukan sebelum sistem menampilkan data dan fungsi internal aplikasi. Elemen seperti *input field e-mail*, *password*, serta *captcha* digunakan untuk mencegah penyalahgunaan sistem dari pihak yang tidak berwenang.

Gambar 3.20 menampilkan *mockup* halaman utama dari aplikasi *Customer Management System* yang muncul setelah pengguna melakukan *login* pada aplikasi.

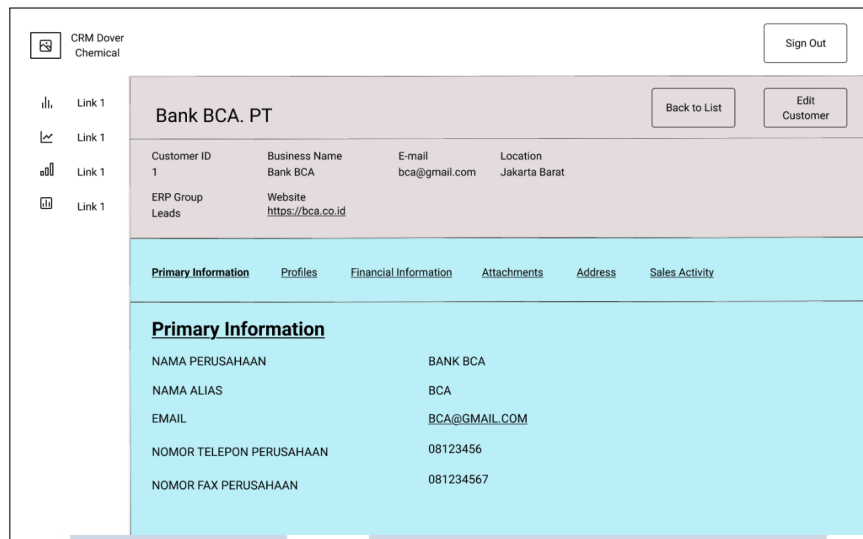
Pada halaman ini, terdapat tabel yang menampilkan data pelanggan yang disertai dengan beberapa informasi seperti tipe bisnis, status, alamat, dan aksi yang dapat pengguna lakukan untuk pengelolaan data suatu pelanggan. Selain itu, juga terdapat filter dan kolom pencarian (*search bar*) yang dapat digunakan untuk mempermudah pencarian pengguna.



Gambar 3.20. Mockup Halaman Utama Aplikasi Customer Management System

Melalui halaman ini, *user* dapat melihat data utama pelanggan secara keseluruhan. Tabel dilengkapi dengan filter, dan elemen *pagination*, serta aksi yang tersedia di kolom dengan pilihan untuk melihat, mengedit atau menghapus pelanggan. Selain itu, juga terdapat fitur tambah pelanggan, serta *side bar* yang dapat menavigasikan pengguna secara langsung, jika ingin berpindah menu.

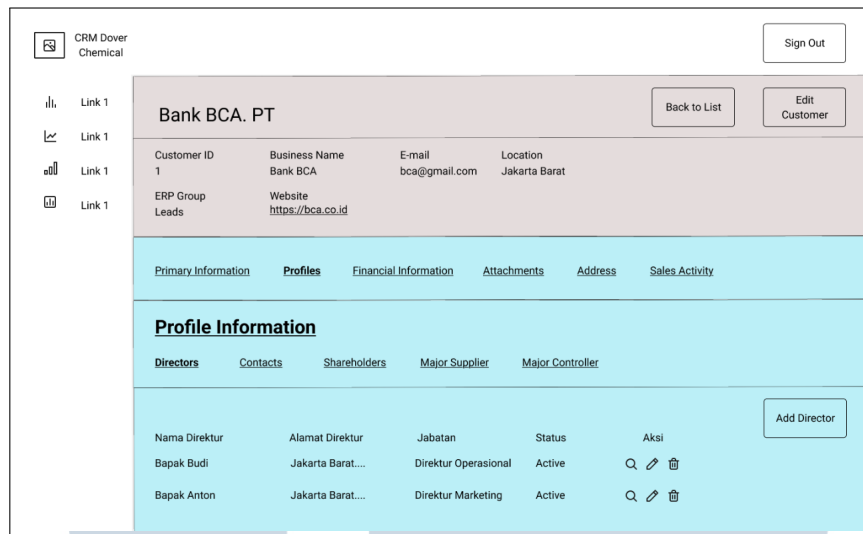
Gambar 3.21 menunjukkan tampilan *mockup* dari halaman Primary Information yang menampilkan berbagai informasi utama dari suatu pelanggan. Informasi yang terdapat pada halaman ini terdiri dari nama, kredensial utama, alamat, tanggal pendirian, kontak yang bisa dihubungi, dan berbagai data lainnya.



Gambar 3.21. *Mockup Halaman Tab Primary Information Aplikasi Customer Management System*

Melalui halaman ini, *user* dapat melihat informasi utama pelanggan secara terpusat dalam satu tampilan. Informasi yang disajikan diambil dari berbagai macam basis data yang memiliki relasi ke basis data utama pelanggan, misalnya seperti data alamat, surat-surat penting, serta kontak dan *credential* pelanggan. Pada halaman ini, pengguna juga dapat melakukan perubahan data langsung pada pelanggan, yang nantinya akan diimplementasikan dengan penambahan fitur *edit user*, sesuai dengan kebutuhan perusahaan.

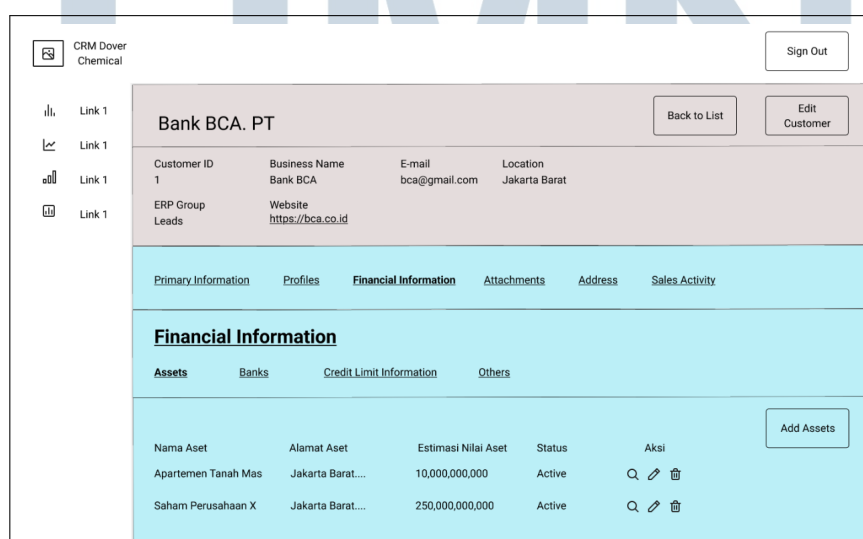
Gambar 3.22 merupakan tampilan dari halaman dari Tab Profile Information yang berisi semua informasi yang terkait dengan pihak atau kontak yang terhubung dengan suatu perusahaan atau pelanggan.



Gambar 3.22. *Mockup Halaman Tab Profile Information Aplikasi Customer Management System*

Melalui halaman ini, *user* dapat mengelola informasi pihak-pihak yang memiliki keterkaitan dengan pelanggan, seperti kontak (internal) yang berhubungan dengan perusahaan tersebut. Selain itu, pengguna juga dapat melakukan aktivitas penambahan, perubahan dan penghapusan pada data direktur, pemegang saham, pemasok utama dan pelanggan utama (pembeli).

Gambar 3.23 merupakan tampilan dari halaman Tab Financial Information yang berisi data-data finansial berupa aset, informasi batas kredit, bank, dan data lainnya.



Gambar 3.23. *Mockup Halaman Tab Financial Information Aplikasi Customer Management System*

Melalui *tab* Financial Information, *user* dapat melihat dan mengelola informasi keuangan pelanggan yang dibutuhkan untuk mendukung proses bisnis perusahaan. Data finansial yang ditampilkan membantu *user* dalam memahami kondisi dan batasan keuangan pelanggan. Pengguna dapat melakukan proses CRUD *create, update and delete* pada *sub-tab* aset, bank, dan lainnya (*others*). Namun untuk *sub-tab* informasi kredit, pengguna hanya dapat melihat data yang disajikan dan terhubung langsung ke basis data *netsuite*.

### 3.4.3 Tampilan Aplikasi

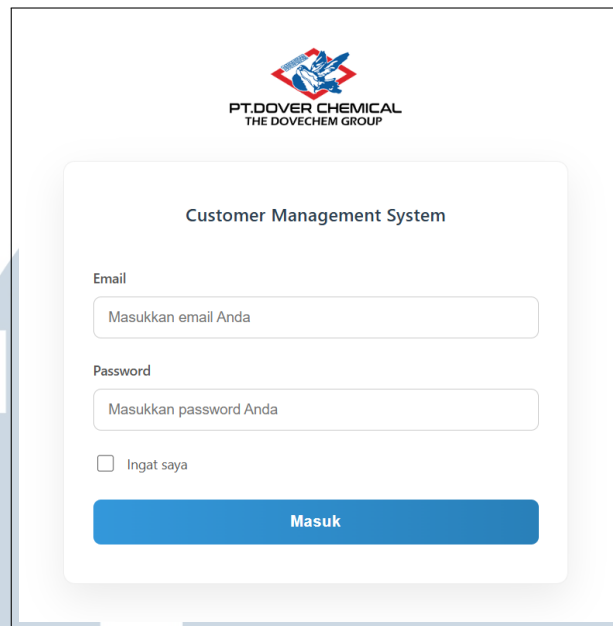
Pada bagian ini akan dijelaskan tampilan aplikasi *Customer Management System* yang telah dikembangkan. Penjelasan difokuskan pada fungsi setiap halaman utama serta perannya dalam mendukung alur penggunaan aplikasi dan pengelolaan data pelanggan oleh *user*.

Aplikasi manajemen data utama pelanggan ini dibuat menggunakan *framework AdminLTE* untuk bagian tampilan, sesuai dengan standar yang saat ini sedang digunakan oleh aplikasi berbasis *web* lainnya di PT Dover Chemical. Sementara itu, *back-end* pada aplikasi dibuat menggunakan *framework* CodeIgniter dengan arsitektur hirarki modul, sesuai dengan standar yang berlaku di perusahaan. Berikut ini adalah tampilan beserta dengan penjelasan dari setiap halaman dan *tab* dari aplikasi ini yang telah dibuat:

#### A Halaman Login

Gambar 3.24 merupakan tampilan dari halaman autentikasi pengguna (*Login*) pada aplikasi *Customer Management System*. Pengguna yang sudah terdaftar dalam basis data perusahaan dapat mengakses aplikasi ini.

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



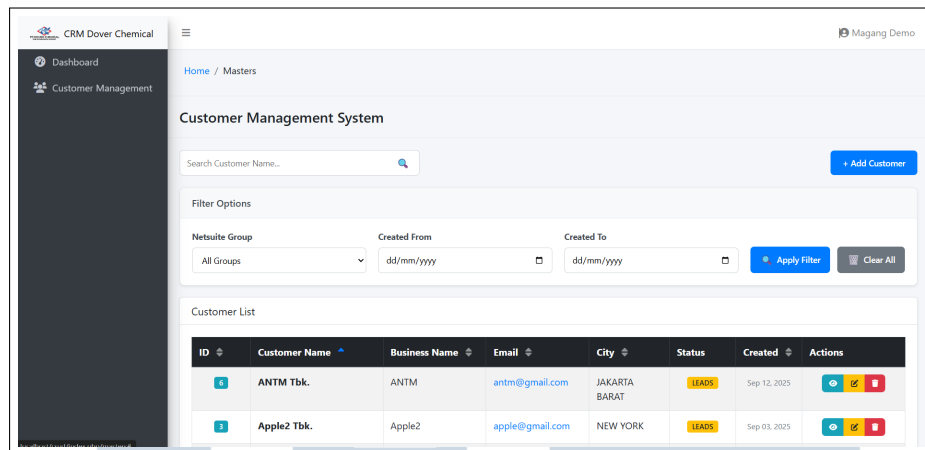
Gambar 3.24. *Tampilan Halaman Login Aplikasi Customer Management System*

Melalui halaman Login ini, sistem melakukan proses autentikasi terhadap pengguna sebelum memberikan akses ke fitur dan data yang tersedia di dalam aplikasi. Halaman ini berperan sebagai mekanisme pengaman awal agar hanya pengguna yang memiliki akun resmi yang dapat mengakses dan mengelola data pelanggan sesuai dengan hak akses yang dimiliki.

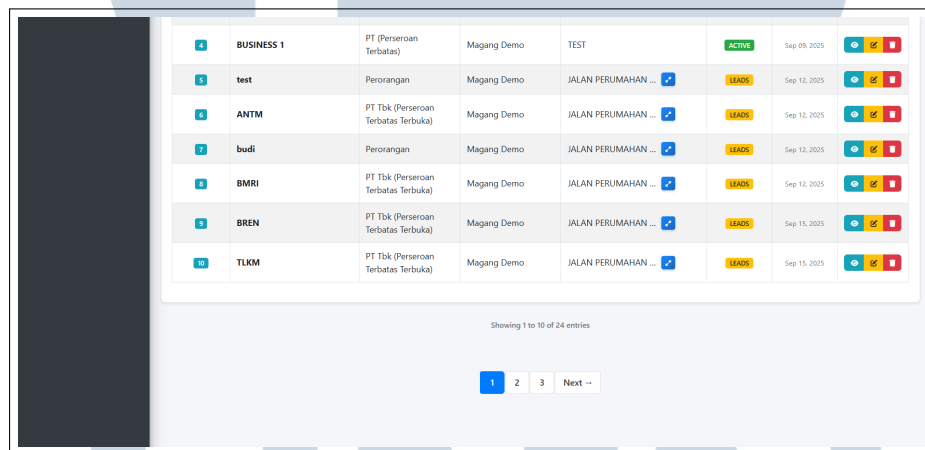
## B Home Page

Gambar 3.25 memperlihatkan antarmuka halaman *Home* yang berisikan daftar *customer* yang sudah didaftarkan ke *database*. Halaman ini menyediakan fitur untuk menambah pelanggan baru, pengeditan pelanggan, serta penghapusan pelanggan. Selain itu, juga ada kolom pencarian dan filter berdasarkan grup dan tanggal pembuatan.





Gambar 3.25. Tampilan Halaman Utama Customer Management Page (Bagian Atas)



Gambar 3.26. Tampilan Halaman Utama Customer Management Page (Bagian Bawah)

Melalui halaman *Home* ini, *user* dapat mengelola seluruh data pelanggan yang tersimpan di dalam sistem secara terpusat. Halaman ini berperan sebagai titik utama pengelolaan data, di mana *user* dapat melakukan pemantauan, pencarian, serta pembaruan data pelanggan sesuai kebutuhan operasional.

Gambar 3.27 merupakan gambar dari tampilan form penambahan pelanggan baru (*add customer*). Pada form ini, pengguna wajib mengisi beberapa informasi yang bersifat *required* misalnya seperti nama, tipe bisnis, alamat (terdiri dari negara, provinsi dan kota), serta tipe kawasan berikat. Selain itu, juga terdapat beberapa bagian yang bersifat opsional dan dapat diisi oleh pengguna, yaitu informasi kontak PIC dan informasi terkait dengan legalitas bisnis.

CRM Dover Chemical | Home / Masters / Create | Magang Demo

**Add New Customer**  
Tambah Pelanggan

[-- Back to List](#)

**Customer Identity**  
Identitas Pelanggan

**Real Customer Name\***  
Nama Asli Pelanggan \*

Enter customer name without title / Masukkan nama pelanggan tanpa gelar

**Business Type\***  
Jenis Bisnis \*

Select Business Type / Pilih Jenis Bisnis

**Title\***  
Gelar \*

Select Title / Pilih Gelar

**Customer Name (Auto Generated)**  
Nama Pelanggan Otomatis Dibuat

This will be automatically generated from Real Customer Name + Title / Ini akan otomatis dibuat dari Nama Asli Pelanggan + Gelar

**Parent Customer**  
Pelanggan Induk

Select Parent Customer / Pilih Pelanggan Induk

Parent customer can only be set when editing / Pelanggan induk hanya dapat diatur saat mengedit

Gambar 3.27. Tampilan Form Add Customer (1)

**Location Information**  
Informasi Lokasi

**Area Customer\***  
Area Pelanggan \*

☒ Lokal (Rp.) ☐ Export (\$)

**Country\***  
Negara \*

Select Country First / Pilih Negara Terlebih Dahulu

**City\***  
Kota \*

Select Country First / Pilih Negara Terlebih Dahulu

**Province**  
Provinsi

Province will be automatically filled when you select a city / Provinsi akan otomatis diisi saat Anda memilih kota

**Address\***  
Alamat \*

This will be set as default visit address / Ini akan diatur sebagai alamat kunjungan default

**Zip Code**  
Kode Pos

12345

Enter postal code / Masukkan kode pos

Gambar 3.28. Tampilan Form Add Customer (2)

**Kawasan Berikat**  
Kawasan Berikat

**No / Tidak**

Select whether the customer is located in a bonded zone / Pilih apakah pelanggan berada di kawasan berikat

**Contact Information**  
Informasi Kontak

**Email Address**  
Alamat Email

**Phone Number**  
Nomor Telepon

+62 812 3456 7890

Enter phone number with country code (e.g., +62) / Masukkan nomor telepon dengan kode negara (contoh: +62)

**Fax Number**  
Nomor Faks

+62 21 1234 5678

Enter fax number with country code (e.g., +62) / Masukkan nomor faks dengan kode negara (contoh: +62)

**Website**  
Situs Web

**Business Description**  
Deskripsi Bisnis

**Legal Information**  
Informasi Legal (Optional)

Gambar 3.29. Tampilan Form Add Customer (3)

The screenshot shows a web form titled 'Add Customer'. It is divided into several sections:

- Business Description:** A single-line text input field.
- Legal Information:** A section containing several fields:
  - NIB (Nomor Induk Berusaha):** A 15-digit text input field.
  - NPWP (15):** A 15-digit text input field for NPWP (Nomor Pokok Wajib Pajak).
  - ID Card Number:** A 16-digit text input field for ID Card Number.
  - Tanggal Pendirian:** A date input field with a dropdown arrow, labeled 'dd/mm/yyyy'.
  - Tanggal Akta Pendirian:** A date input field with a dropdown arrow, labeled 'dd/mm/yyyy'.
- Buttons:** At the bottom, there are three buttons: 'Create Customer' (blue), 'Batal' (grey), and 'Cancel' (grey).

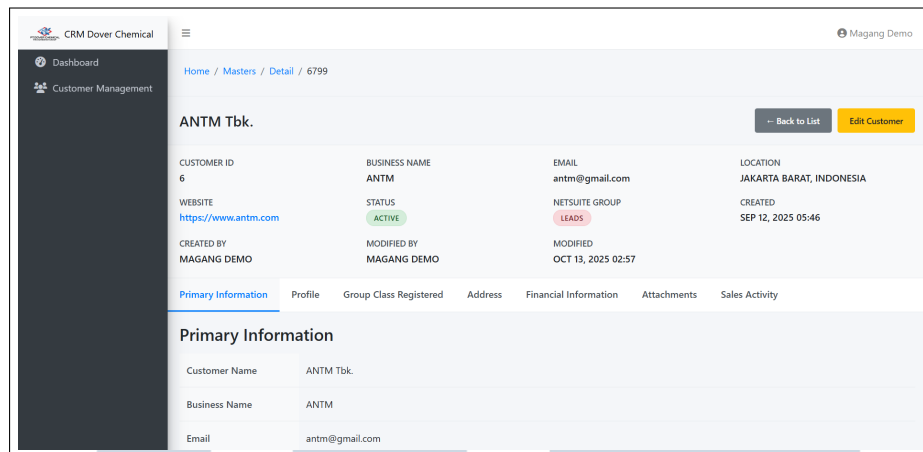
Gambar 3.30. Tampilan Form Add Customer (4)

Melalui rangkaian tampilan form ini, *user* diarahkan untuk mengisi data pelanggan secara bertahap dan terstruktur. Pembagian tampilan form ke dalam beberapa bagian bertujuan untuk menjaga keterbacaan dan memudahkan *user* dalam melengkapi informasi tanpa harus mengisi seluruh data dalam satu tampilan yang padat.

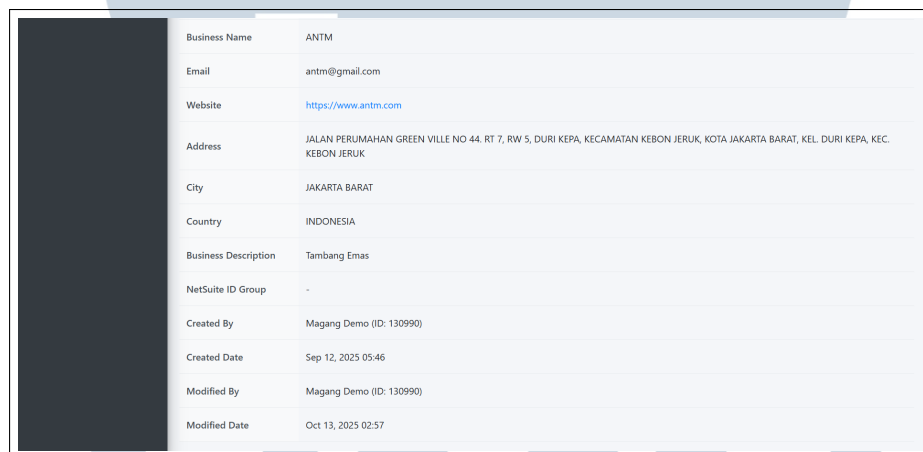
Dengan adanya pemisahan antara data yang bersifat wajib dan opsional, sistem dapat memastikan bahwa informasi utama pelanggan tetap terpenuhi, sementara data pendukung dapat ditambahkan sesuai kebutuhan. Pendekatan ini membantu meningkatkan konsistensi dan kelengkapan data pelanggan yang tersimpan di dalam sistem.

### C Primary Information

Gambar 3.31 dan Gambar 3.32 memperlihatkan tampilan *tab Primary Information*. *Tab* ini menampilkan data-data utama dari *customer*, misalnya seperti nama bisnis, alamat, deskripsi, nomor pajak (NPWP), tanggal pendirian, dan berbagai informasi lainnya.



Gambar 3.31. Tampilan Halaman Primary Information (Bagian Atas)



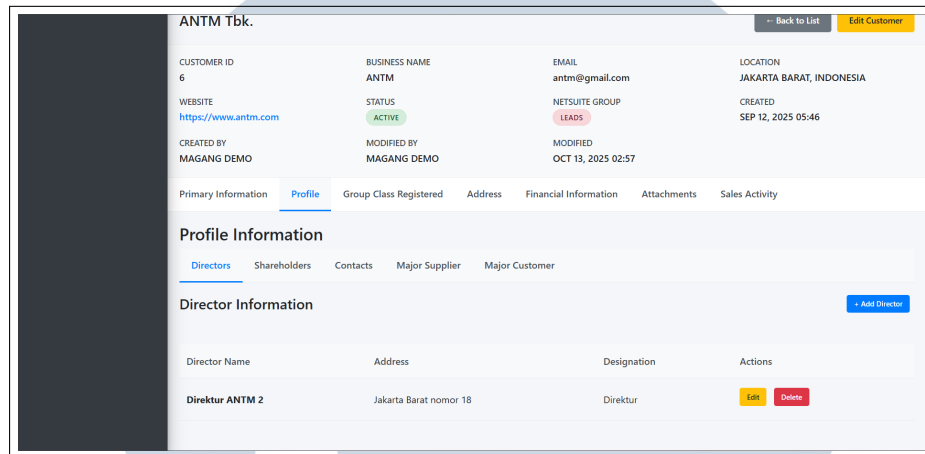
Gambar 3.32. Tampilan Halaman Primary Information (Bagian Bawah)

Pada bagian atas halaman (Gambar 3.31) ditampilkan informasi umum dan identitas utama *customer* seperti nama, nama bisnis, dan e-mail. Sementara itu, bagian bawah halaman (Gambar 3.32) menampilkan informasi tambahan yang melengkapi data utama seperti website, alamat (kota, provinsi, negara), deskripsi bisnis, id grup pada Netsuite, nama pengguna yang membuat data pelanggan, tanggal pembuatan data, serta informasi terkait pengeditan data.

## D Profile

*Tab Profile* merupakan halaman yang berisi data terkait dari pihak yang terafiliasi dengan *customer*. Terdapat 5 *sub-tab* pada *tab* ini, yaitu *Directors*, *Shareholders*, *Contacts*, *Major Supplier*, dan *Major Customer*. Pengguna dapat melakukan aktivitas *create*, *edit*, dan *delete* pada kelima *sub-tab* tersebut.

Gambar 3.33 menunjukkan tampilan dan *sub-tab Directors* yang berisi data seperti nama direktur, alamat, jabatan, dan aksi yang dapat pengguna lakukan. Pengguna dapat melakukan *create*, *delete* dan *update* pada data direktur.



Gambar 3.33. Tampilan Halaman Profile Information Sub-Tab Directors pada Modul Customer Management

Pada halaman ini, pengguna dapat melakukan pengelolaan data direktur melalui aksi *create*, *update*, dan *delete*. Fitur ini memungkinkan pengguna untuk menjaga agar data direktur tetap sesuai dan terbaru dengan kondisi aktual perusahaan.

Gambar 3.34 merupakan gambar dari form penambahan data direktur. Pengguna dapat mengisi informasi yang terkait dengan direktur dan semua *field* bersifat wajib diisi oleh pengguna.

Director Name / Nama Direktur	Address / Alamat	Designation / Jabatan	Status	Actions / Aksi
Direktur BCA Nomor 53			Active	<a href="#">Edit</a> <a href="#">Delete</a>
Direktur BCA 4			Active	<a href="#">Edit</a> <a href="#">Delete</a>
Test 2	test	test	Active	<a href="#">Edit</a> <a href="#">Delete</a>

Gambar 3.34. *Form Tambah Directors pada Halaman Profile Information Sub-Tab Directors Modul Customer Management*

Seluruh *field* pada form ini bersifat wajib diisi oleh pengguna sebelum data dapat disimpan ke dalam sistem. Data yang harus diisi adalah nama direktur, alamat dan jabatan spesifik direktur tersebut.

Gambar 3.35 merupakan tampilan dari halaman *sub-tab* Shareholders yang berisikan data-data yang terkait dengan pemegang saham, yang terdiri dari nama pemegang saham, alamat, dan jumlah saham. Pada *sub-tab* ini, pengguna juga dapat melakukan aktivitas pembuatan, pengeditan dan penghapusan data pemegang saham.

Shareholder Name / Nama Pemegang Saham	Address / Alamat	Share Number / Jumlah Saham	Status	Actions / Aksi
test	test	test	Active	<a href="#">Edit</a> <a href="#">Delete</a>
test	testwerwer	test	Active	<a href="#">Edit</a> <a href="#">Delete</a>

Gambar 3.35. *Tampilan Halaman Profile Information Sub-Tab Shareholders pada Modul Customer Management*

Melalui *sub-tab* ini, pengguna dapat melakukan pengelolaan data pemegang

saham, seperti menambahkan data baru, mengubah data yang sudah ada, serta menghapus data sesuai kebutuhan.

Gambar 3.36 merupakan gambar dari tampilan form penambahan pemegang saham, dimana pengguna dapat mengisi informasi seperti nama, alamat dan jumlah saham, dan semua *field* bersifat wajib diisi.

Shareholder Name / Nama Pemegang Saham	Share Number / Nomor Saham	Status	Actions / Aksi
test		Active	<a href="#">Edit</a> <a href="#">Delete</a>
test		Active	<a href="#">Edit</a> <a href="#">Delete</a>

Gambar 3.36. Form Tambah Shareholders pada Halaman Profile Information Sub-Tab Shareholders Modul Customer Management

Gambar 3.37 menampilkan gambar dari halaman *sub-tab* Contacts yang menampilkan data dari kontak pelanggan yang terdiri dari nama, nomor telepon, email, posisi dan departemen.

Contact Name / Nama Kontak	Phone / Telepon	Email / Surat Elektronik	Position / Posisi	Department / Departemen	Status	Actions / Aksi
tester2	081348520623	test@gmail.com	test	Owner	Pending	<a href="#">Edit</a> <a href="#">Delete</a>
budi	081348520623	magang.demo@dovechem.co.id	test	Production	Active	<a href="#">Edit</a> <a href="#">Delete</a>
william lo	081348520623	magang.demo@dovechem.co.id	test	Owner	Active	<a href="#">Edit</a> <a href="#">Delete</a>

Gambar 3.37. Tampilan Halaman Profile Information Sub-Tab Contacts pada Modul Customer Management

Melalui halaman ini, pengguna dapat mengelola data kontak pelanggan



sebagai pihak yang dapat dihubungi oleh tim penjualan PT Dover Chemical.

Gambar 3.38 merupakan gambar dari form untuk penambahan kontak. Pada form ini, pengguna wajib mengisi informasi seperti nama, nomor telepon email, posisi dan departemen.

Contact Name / Nama Kontak	Phone / Telepon	Email / Surat Elektronik	Position / Posisi	Department / Departemen	Status	Actions / Aksi
tester2	0813				Pending	Edit, Delete
budi	0813				Active	Edit, Delete
william lo	0813				Active	Edit, Delete

Gambar 3.38. Form Tambah Contacts pada Halaman Profile Information Sub-Tab Contacts Modul Customer Management

## E Group Class Registered

Gambar 3.39 memperlihatkan tampilan tab *Group Class Registered*. Pengguna dapat mengakses tab ini untuk melihat apakah suatu *customer* sudah terdaftar di suatu *group class* di ERP (*Enterprise Resource Planning*) Netsuite atau belum.

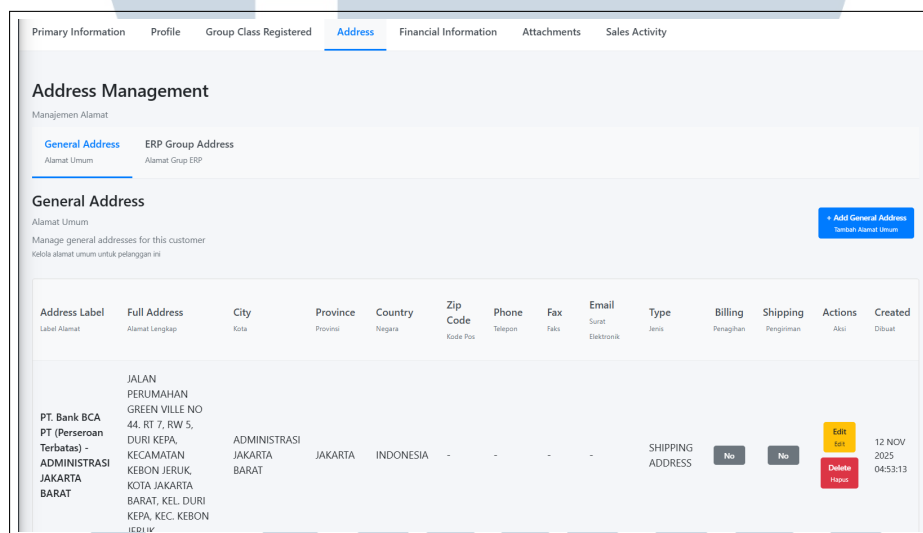
Customer ID	Customer Code	Customer Name	Type Customer	Class ID	Class Name	Sales Rep
1	110-10	Customer name 1	Prospect	1	SR2	Magang Sales
3	112-10	Customer Name 3	Child	1	SR2	Magang Sales

Gambar 3.39. Tampilan Sub-tab Group Class Registered pada Modul Customer Management

Melalui *tab* ini, pengguna dapat mengetahui apakah data *customer* sudah terintegrasi dan terdaftar dalam *group class* tertentu di ERP, sehingga memudahkan proses monitoring dan validasi data pelanggan.

## F Address

Gambar 3.40 menunjukkan tampilan *tab Address* pada modul *Customer Management*. Terdapat 2 *sub-tab* yang dapat diakses pada *tab* ini, yaitu *General Address* dan *ERP Group Address*. Pengguna dapat menambahkan alamat *General* yang bertipe alamat kirim, pembayaran, atau pengiriman, pada *sub-tab General Address*. Pada *sub-tab ERP Group Address*, pengguna dapat melihat alamat *customer* yang sudah terdaftar di sistem ERP (*Enterprise Resource Planning*).



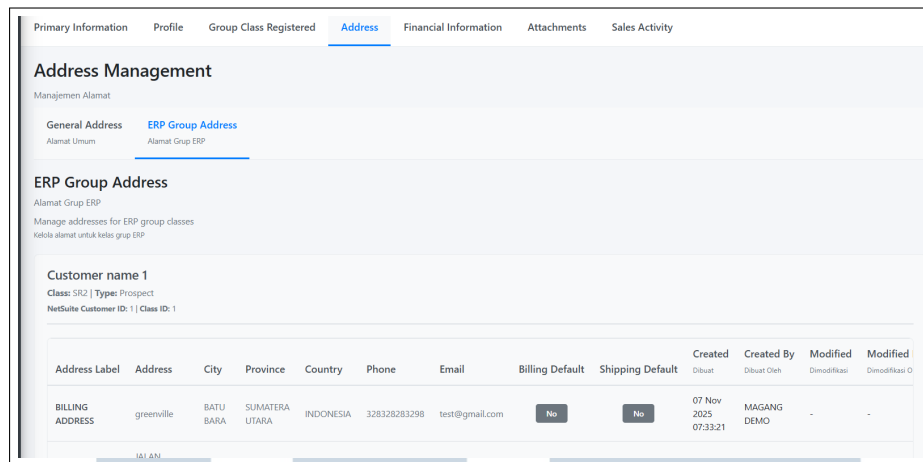
Gambar 3.40. Tampilan Sub-tab General Address pada Modul Customer Management

Melalui *sub-tab General Address*, pengguna dapat melihat daftar alamat yang dimiliki oleh *customer* serta menambahkan alamat baru dengan kategori tertentu, seperti alamat pengiriman, penagihan, maupun alamat pembayaran. Proses penambahan alamat dilakukan melalui form yang ditampilkan pada Gambar 3.41 dan Gambar 3.42, di mana pengguna wajib mengisi informasi alamat secara lengkap.

Gambar 3.41. Form Tambah General Address (Bagian Atas)

Gambar 3.42. Form Tambah General Address (Bagian Bawah)

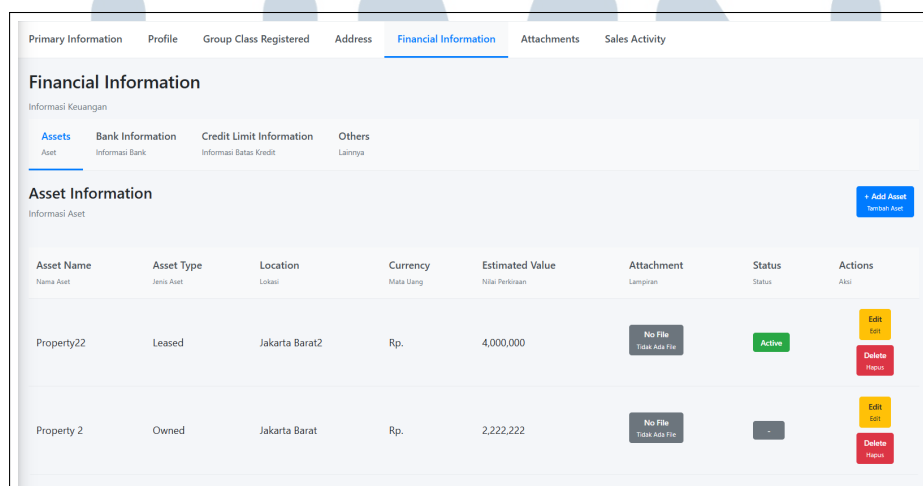
Sementara itu, *sub-tab ERP Group Address* yang ditunjukkan pada Gambar 3.43 digunakan untuk menampilkan data alamat *customer* yang telah terdaftar dan tersinkronisasi dengan sistem ERP (*Enterprise Resource Planning*). Informasi pada *sub-tab* ini bersifat *read-only* dan berfungsi sebagai referensi keterpaduan data antara aplikasi *Customer Management System* dan sistem ERP.



Gambar 3.43. Tampilan Halaman Address Sub-Tab ERP Group Address pada Modul Customer Management

## G Financial Information

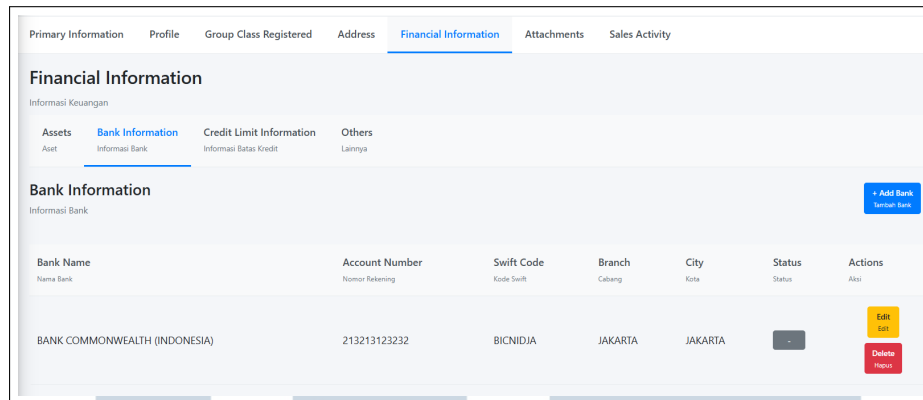
Gambar 3.44 memperlihatkan tampilan *tab Financial Information*. Terdapat 4 sub-tab yang bisa diakses oleh user pada tab ini, yaitu *Assets*, *Bank Information*, *Credit Limit Information*, dan *Others*. Pengguna dapat melakukan kegiatan *add*, *edit*, dan *delete* pada sub-tab *Assets*, *Bank Information*, dan *Others*. Pada sub-tab *Credit Limit Information*, user hanya bisa melakukan *view* untuk melihat data batas kredit dari customer tersebut.



Gambar 3.44. Tampilan Sub-Tab Financial Information pada Modul Customer Management

Gambar 3.45 merupakan gambar dari tampilan sub-tab Banks yang menampilkan data-data yang terkait dengan informasi bank pelanggan, seperti yang

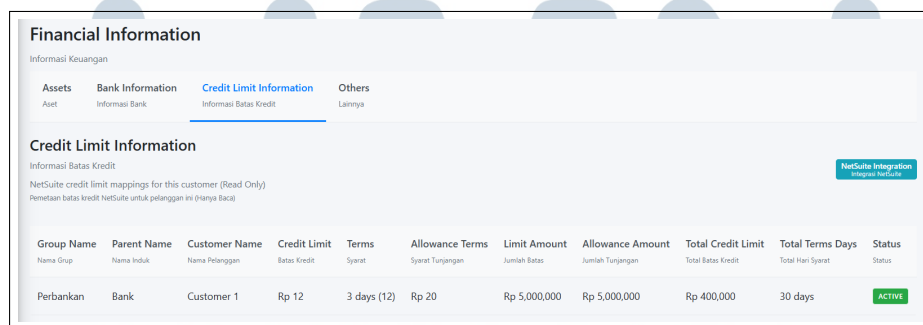
tertera pada gambar dibawah ini.



Gambar 3.45. Tampilan Halaman Financial Information Sub-Tab Banks pada Modul Customer Management

Melalui *sub-tab* ini, pengguna dapat melakukan pengelolaan data informasi bank pelanggan untuk mendukung proses administrasi dan transaksi keuangan. Data yang ditampilkan berupa nama bank, nomor akun, kode bank, kota, status, dan aksi yang dapat dilakukan pengguna.

Gambar 3.46 merupakan tampilan dari *sub-tab* Credit Limit Information yang berisi data-data yang terkait dengan batas kredit yang dimiliki oleh suatu pelanggan.



Gambar 3.46. Tampilan Halaman Financial Information Sub-Tab Credit Limit Information pada Modul Customer Management

Pada halaman ini, ditampilkan data seperti grup bisnis yang terdaftar, nama induk perusahaan atau entitas terkait, nama pelanggan, serta nilai batas kredit yang ditetapkan. Data tersebut diambil dari basis data yang terhubung ke ERP Netsuite, sehingga hanya bersifat *view only* dan pengguna tidak dapat melakukan perubahan data.

Gambar 3.47 dan Gambar 3.48 merupakan gambar dari tampilan bagian *sub-tab* Others, yang berisi beberapa informasi pencatatan informasi keuangan pelanggan yang dilakukan berdasarkan perspektif internal perusahaan.

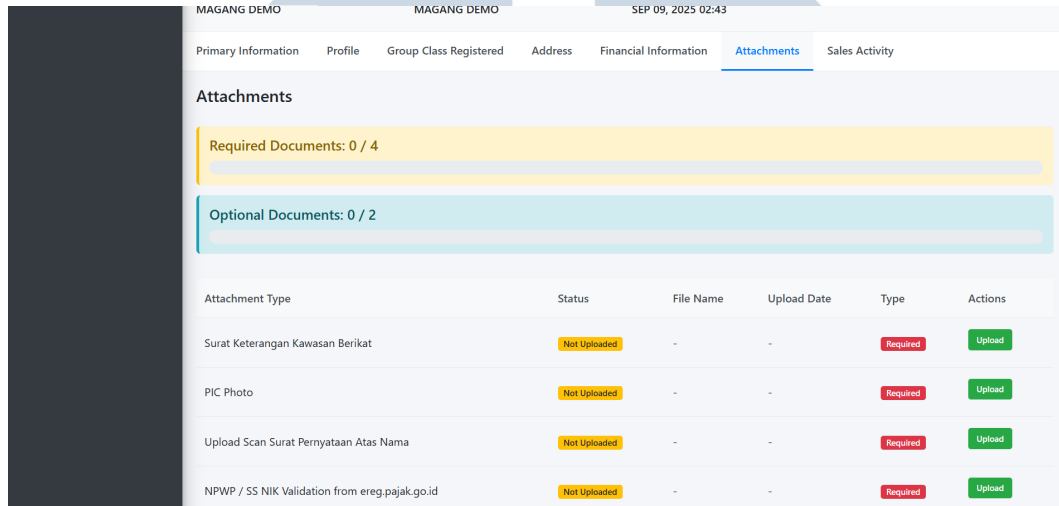
Gambar 3.47. Tampilan Halaman Financial Information Sub-Tab Others (Bagian Atas) pada Modul Customer Management

Gambar 3.48. Tampilan Halaman Financial Information Sub-Tab Others (Bagian Bawah) pada Modul Customer Management

Pada *sub-tab* ini, data yang ditampilkan dan dapat dimasukkan oleh pengguna pada halaman ini terdiri dari, modal awal yang diberikan, jumlah yang telah disetor, tipe bisnis, serta beberapa informasi yang terkait dengan kinerja penjualan.

## H Attachments

Gambar 3.49 memperlihatkan tampilan *tab Attachments*. Pengguna dapat mengupload berbagai dokumen penting yang terkait dengan *customer*.



Gambar 3.49. Tampilan halaman *Attachments* pada modul Customer Management

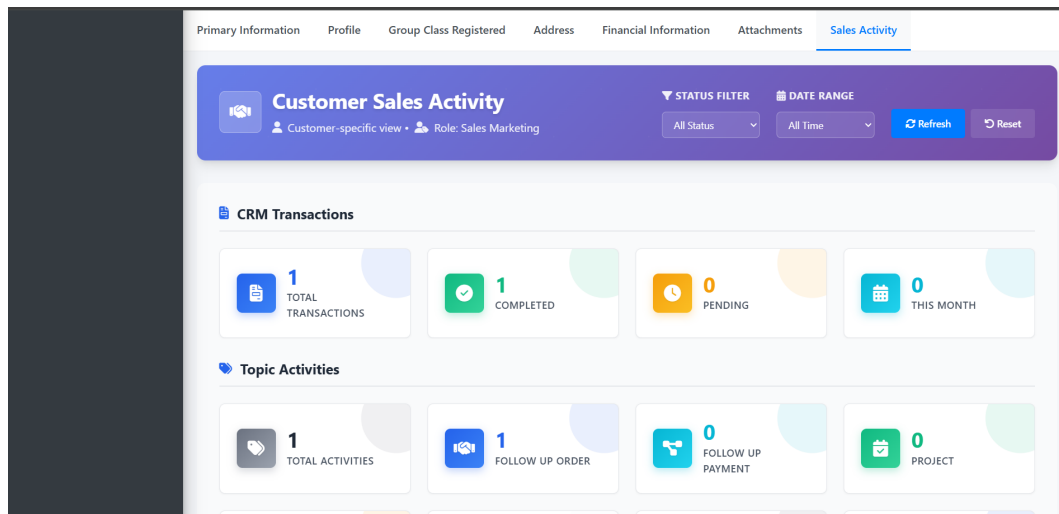
Terdapat 2 tipe dokumen yang bisa diupload oleh *user* pada *tab* ini, yaitu dokumen yang bersifat *required* atau *optional*. Pada *tab* ini juga terdapat *progress bar* yang menunjukkan berapa banyak dokumen yang sudah diupload oleh *user*.

## I Sales Activity

Gambar 3.50 memperlihatkan tampilan *tab Sales Dashboard* yang berfungsi untuk menampilkan data histori transaksi penjualan yang disajikan dalam bentuk *dashboard* yang bersifat interaktif. Terdapat beberapa data yang dapat digunakan oleh pengguna untuk memantau performa dan hasil dari transaksi penjualan.

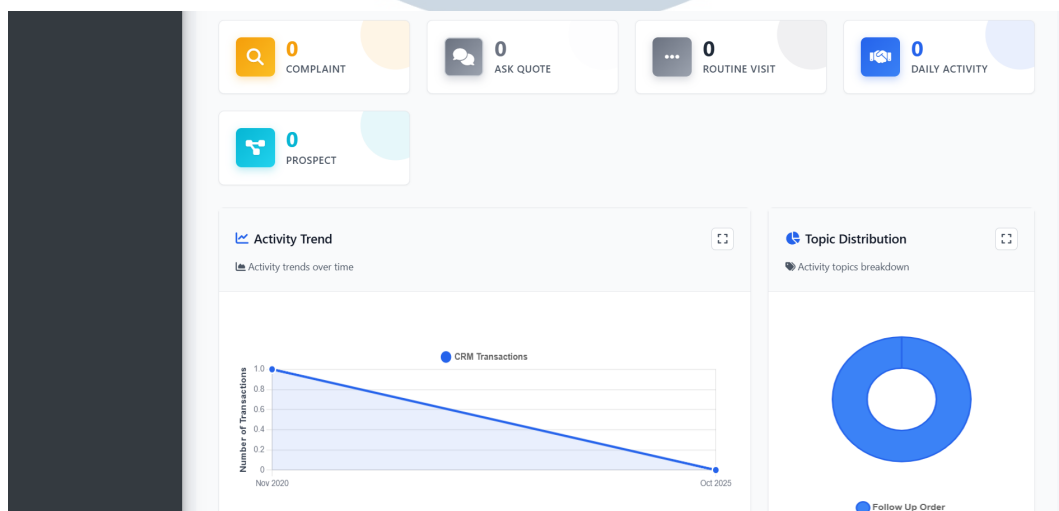
U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A





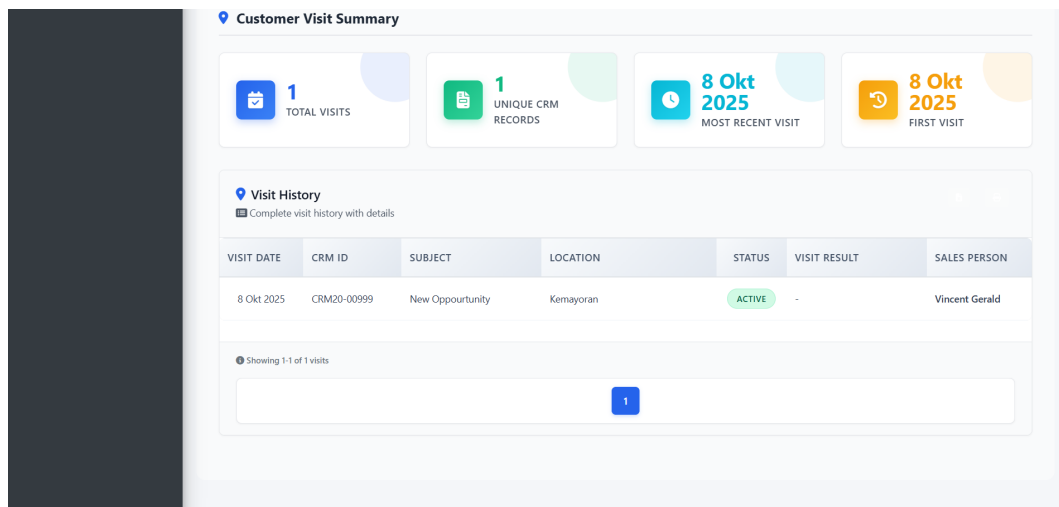
Gambar 3.50. Tampilan awal halaman *Sales Dashboard* pada modul Customer Management

Gambar 3.50 diatas, adalah data total transaksi CRM (*Customer Relationship Management*), yaitu data yang merepresentasikan seluruh aktivitas transaksi dan interaksi dengan pelanggan.



Gambar 3.51. Bagian *CRM Transactions* dan *Topic Activities* pada halaman *Sales Dashboard*

Kemudian, pada gambar 3.51 terdapat diagram yang menunjukkan distribusi, *trend*, dan total dari kegiatan yang dikelompokkan berdasarkan topik.



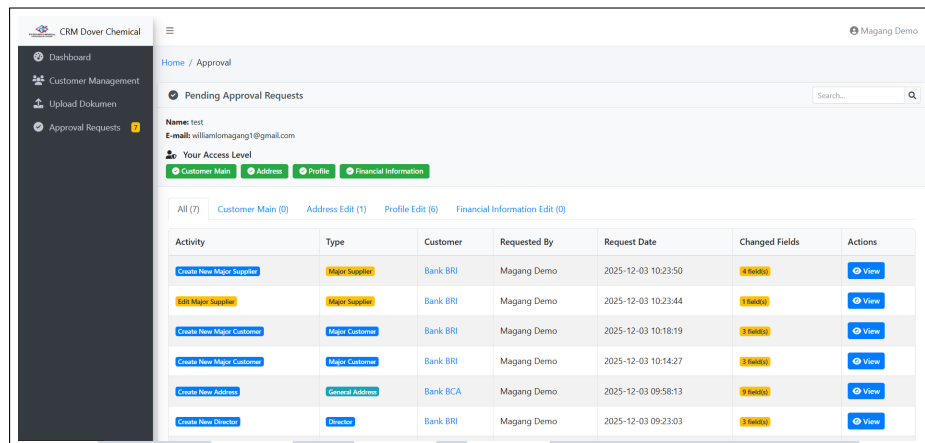
Gambar 3.52. Bagian *Activity Trend*, *Topic Distribution*, dan *Customer Visit Summary* pada halaman *Sales Dashboard*

Selain itu, juga terdapat tabel *visit history* yang berfungsi untuk menunjukkan hasil dari kunjungan *sales* sebelumnya.

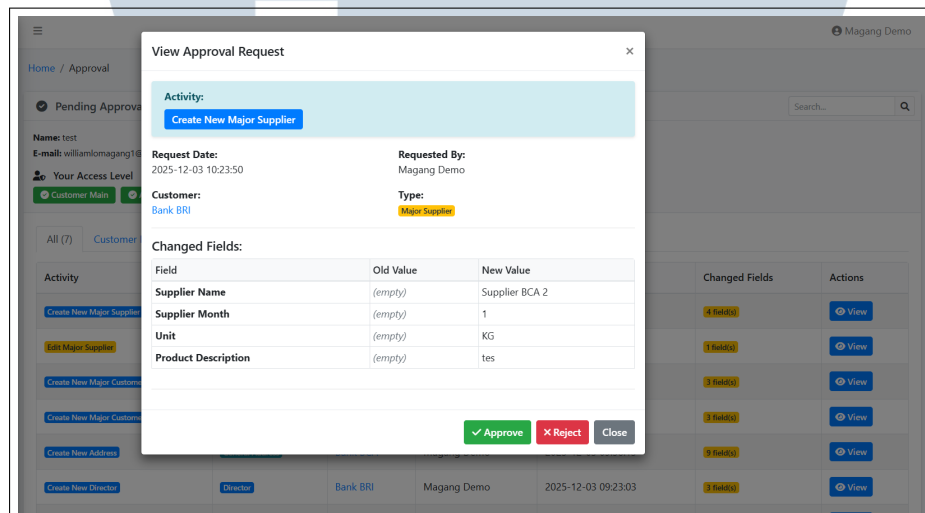
## J Approval

Halaman *Approval* memiliki fungsi untuk menampilkan semua permintaan (*request*) *approval* yang berlaku untuk data pelanggan yang bersifat pending. *Admin* yang memiliki akses pada halaman ini, dapat melakukan *approve* atau *reject* sesuai dengan skenario perubahan data yang ada. Contoh dari skenario yang memerlukan *approval* dari *admin* adalah perubahan dan penambahan data (berlaku untuk semua data di semua *tab* dan *sub-tab*). Gambar dari tampilan halaman *Approval* dapat dilihat pada Gambar 3.53 dan 3.54.

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 3.53. Tampilan Halaman Approval Modul Customer Management



Gambar 3.54. Tampilan View Approval Request pada Modul Customer Management

### 3.4.4 Pengujian Sistem

Pengujian aplikasi manajemen data utama pelanggan dilakukan dengan metode pengujian *black box* yang dilakukan pada tahap awal pengujian sistem. Pengujian dilakukan oleh *supervisor* magang, serta berorientasi pada fitur-fitur yang sudah dibuat pada setiap modul aplikasi yang ada.

Tabel 3.2. Hasil Black Box Testing

No	Fitur	Kasus	Output	Hasil	Kesimpulan
1	Login	Data benar	Masuk ke sistem	Sesuai	Lulus
2	Login	Email salah	Error email muncul	Sesuai	Lulus
3	Login	Pass salah	Error login muncul	Sesuai	Lulus
4	Login	Field kosong	Validasi berhasil muncul	Sesuai	Lulus
5	Home Page	Klik add	Form berhasil tampil	Sesuai	Lulus
6	Home Page	Klik view	Data berhasil tampil	Sesuai	Lulus
7	Home Page	Klik edit	Data berhasil diedit	Sesuai	Lulus
8	Home Page	Klik delete	Data berhasil dihapus	Sesuai	Lulus
9	Home Page	Cari data	Hasil berhasil muncul	Sesuai	Lulus
10	Home Page	Filter date	Data berhasil terfilter	Sesuai	Lulus
11	Home Page	Filter group	Data berhasil terfilter	Sesuai	Lulus
Lanjut ke halaman berikutnya					

Tabel 3.2: Hasil Black Box Testing (lanjutan)

No	Fitur	Kasus	Output	Hasil	Kesimpulan
12	Primary Information	Buka tab	Data berhasil tampil	Sesuai	Lulus
13	Primary Information	Edit data	Data berhasil disimpan	Sesuai	Lulus
14	Primary Information	Wajib kosong	Validasi berhasil muncul	Sesuai	Lulus
15	Directors	Tambah data	Data berhasil disimpan	Sesuai	Lulus
16	Directors	Edit data	Data berhasil disimpan	Sesuai	Lulus
17	Directors	Hapus data	Data berhasil dihapus	Sesuai	Lulus
18	Shareholders	Tambah data	Data berhasil disimpan	Sesuai	Lulus
19	Shareholders	Edit data	Data berhasil disimpan	Sesuai	Lulus
20	Shareholders	Hapus data	Data berhasil dihapus	Sesuai	Lulus
21	Contacts	Tambah data	Data berhasil disimpan	Sesuai	Lulus
22	Contacts	Edit data	Data berhasil disimpan	Sesuai	Lulus
Lanjut ke halaman berikutnya					

Tabel 3.2: Hasil Black Box Testing (lanjutan)

No	Fitur	Kasus	Output	Hasil	Kesimpulan
23	Contacts	Hapus data	Data berhasil dihapus	Sesuai	Lulus
24	Supplier	Tambah data	Data berhasil disimpan	Sesuai	Lulus
25	Supplier	Hapus data	Data berhasil dihapus	Sesuai	Lulus
26	Customer	Tambah data	Data berhasil disimpan	Sesuai	Lulus
27	Customer	Hapus data	Data berhasil dihapus	Sesuai	Lulus
28	Group Class	Buka tab	Data berhasil tampil	Sesuai	Lulus
29	Group Class	Tidak ada	Status berhasil tampil	Sesuai	Lulus
30	Address General	Tambah data	Data berhasil disimpan	Sesuai	Lulus
31	Address General	Edit data	Data berhasil disimpan	Sesuai	Lulus
32	Address General	Hapus data	Data berhasil dihapus	Sesuai	Lulus
33	Address ERP	Lihat data	Data berhasil tampil	Sesuai	Lulus
Lanjut ke halaman berikutnya					

Tabel 3.2: Hasil Black Box Testing (lanjutan)

No	Fitur	Kasus	Output	Hasil	Kesimpulan
34	Assets	Tambah data	Data berhasil disimpan	Sesuai	Lulus
35	Assets	Edit data	Data berhasil disimpan	Sesuai	Lulus
36	Assets	Hapus data	Data berhasil dihapus	Sesuai	Lulus
37	Bank Information	Tambah data	Data berhasil disimpan	Sesuai	Lulus
38	Bank Information	Edit data	Data berhasil disimpan	Sesuai	Lulus
39	Bank Information	Hapus data	Data berhasil dihapus	Sesuai	Lulus
40	Credit Limit	Lihat data	Data berhasil tampil	Sesuai	Lulus
41	Other Information	Tambah data	Data berhasil disimpan	Sesuai	Lulus
42	Other Information	Edit data	Data berhasil disimpan	Sesuai	Lulus
43	Other Information	Hapus data	Data berhasil dihapus	Sesuai	Lulus
44	Attachment	Upload file	File berhasil diupload	Sesuai	Lulus
Lanjut ke halaman berikutnya					

Tabel 3.2: Hasil Black Box Testing (lanjutan)

No	Fitur	Kasus	Output	Hasil	Kesimpulan
46	Attachment	Hapus file	File berhasil dihapus	Sesuai	Lulus
47	Sales Dashboard	Lihat dashboard	Grafik berhasil tampil	Sesuai	Lulus
48	Sales Dashboard	Lihat histori	Tabel berhasil tampil	Sesuai	Lulus
49	Sales Dashboard	Filter data	Grafik berhasil berubah	Sesuai	Lulus

### 3.5 Kendala dan Solusi yang Ditemukan

#### A Kendala

Kendala yang ditemui selama proses pengujian sistem adalah sebagai berikut.

1. Pada awal pengembangan, fitur belum terdefinisi dengan jelas, sehingga beberapa fungsi mengalami penambahan atau penyesuaian mendadak akibat belum adanya *product requirement* yang baku.
2. Terjadi kesalahan interpretasi terhadap kolom pada database yang digunakan untuk field tertentu, sehingga mengakibatkan ketidaksesuaian antara data yang diinput dan data yang seharusnya digunakan oleh sistem.

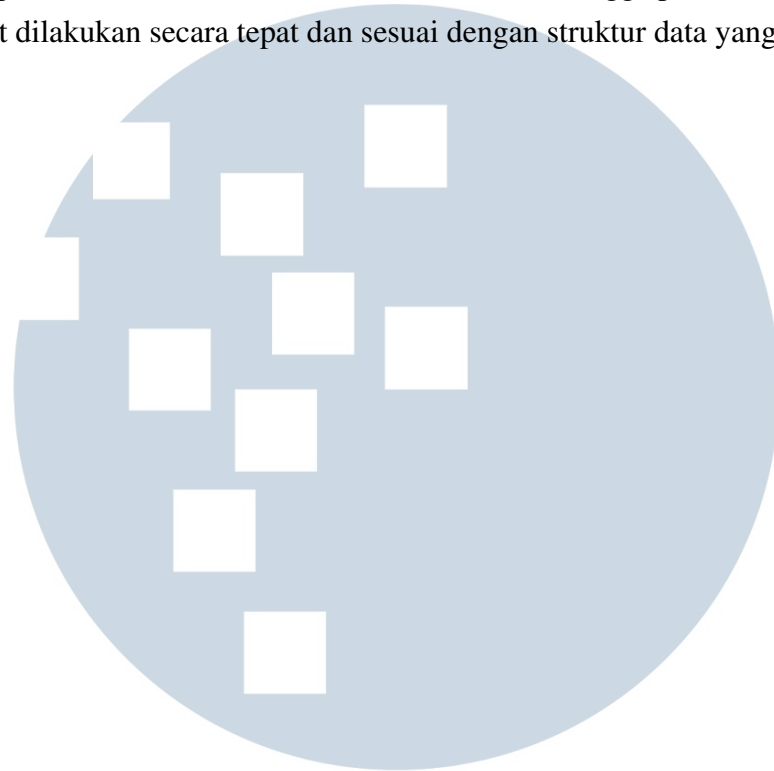
#### B Solusi

Solusi yang diterapkan untuk mengatasi kendala tersebut antara lain sebagai berikut.

1. Berdiskusi dengan *supervisor* untuk menentukan *Product Requirement Document (PRD)*, serta mengadakan meeting secara rutin untuk memastikan setiap perubahan fitur dapat dipantau dan disesuaikan dengan kebutuhan sistem.



2. Melakukan diskusi lebih lanjut dengan *supervisor* untuk mempelajari fungsi setiap kolom *database* secara lebih mendalam, sehingga proses implementasi dapat dilakukan secara tepat dan sesuai dengan struktur data yang benar.



UMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA