

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama kegiatan magang di PT Surya Digital Indonesia, pelaksanaan kerja magang berada langsung pada posisi *Front-End Developer* di proyek pengembangan Aplikasi Mobile ERP untuk mendukung manajemen operasional perusahaan. Posisi pelaksanaan ini ditempatkan di bawah divisi pengembangan digital yang bertanggung jawab dalam pengembangan sistem dan implementasi fitur pada Aplikasi Mobile ERP tersebut.

Dalam pelaksanaannya, seluruh tugas dan aktivitas kerja dikoordinasikan serta diarahkan secara intensif oleh Bapak Ade Surya Wibawa selaku pembimbing lapangan yang ditunjuk oleh perusahaan. Beliau berperan sebagai mentor yang memberikan arahan teknis, bimbingan pengembangan, serta evaluasi berkala terhadap hasil kerja yang diselesaikan.

Proses koordinasi meliputi diskusi awal untuk memahami kebutuhan proyek dan klien, bimbingan rutin mengenai teknis implementasi, pelaporan progres kerja, serta penerimaan umpan balik langsung dari klien yang digunakan untuk melakukan revisi dan penyempurnaan fitur. Dengan mekanisme koordinasi tersebut, proses pengerjaan proyek dapat berjalan lebih terarah dan sesuai standar yang diterapkan perusahaan.

3.2 Tugas yang Dilakukan

Selama melaksanakan kegiatan magang, pelaksana kerja magang bertanggung jawab untuk mendukung proses pengembangan Aplikasi Mobile ERP yang sedang dikembangkan oleh perusahaan. Tugas utamanya meliputi pengembangan struktur dan tampilan halaman aplikasi menggunakan Visual Studio Code dengan menggunakan framework React Native serta bahasa pemrograman JavaScript. Dalam proses pengerjaannya harus menyesuaikan desain dengan kebutuhan klien, mulai dari penyusunan tata letak *button* hingga penempatan elemen visual, guna memastikan alur penggunaan aplikasi berjalan secara terstruktur dan mudah dipahami.

Selain pengembangan tampilan, terdapat juga implementasi berbagai form input yang berkaitan dengan fitur *Create, Read, Update, Delete* (CRUD) pada

beberapa modul yang ada di aplikasi. Selain itu, pengerjaan sejumlah revisi berdasarkan permintaan klien langsung, seperti dengan adanya penambahan dan penyempurnaan komponen form, pengelolaan tombol aksi CRUD, serta perbaikan bug yang beberapa kali ditemukan pada fitur tersebut. Di samping itu, dilakukan juga penambahan fitur penyortiran data serta membantu implementasi fitur kalender data untuk meningkatkan kemudahan dan efisiensi penggunaan aplikasi.

3.3 Uraian Pelaksanaan Magang

Pelaksanaan magang ini difokuskan pada pengembangan Aplikasi Mobile ERP untuk mendukung operasional perusahaan. Proses ini mengikuti tahapan-tahapan yang telah di uraikan dalam Subbab 1.3 yang menjelaskan tentang Waktu dan Prosedur Pelaksanaan Kerja Magang, dengan penekanan pada aspek teknis pengembangan. Tabel 3.1 mendokumentasikan pekerjaan selama periode magang, dari minggu ke-1 hingga ke-17, yaitu sebagai berikut.

Tabel 3.1. Rekapitulasi tugas mingguan selama magang

Minggu Ke-	Pekerjaan yang Dilakukan
1	Pengenalan lingkungan kerja, alur pengembangan proyek, serta struktur aplikasi Mobile ERP. mahasiswa mulai mempelajari teknologi <i>React Native</i> dan <i>TypeScript</i> yang digunakan dalam pengembangan aplikasi.
2	Pendalaman <i>React Native</i> dan <i>TypeScript</i> , meliputi struktur folder, komponen tampilan, serta sistem navigasi yang digunakan pada aplikasi Mobile ERP.
3	Melanjutkan proses pembelajaran serta melakukan penyesuaian awal pada desain tampilan aplikasi dan memahami alur perpindahan antarhalaman.
4	Mengimplementasikan desain aplikasi sesuai kebutuhan klien pada halaman home dan modul utama sistem ERP.
5	Mulai mengerjakan tampilan pada modul Master Data, meliputi halaman Customer, Supplier, dan Product, termasuk penyusunan form input untuk kebutuhan Create dan Update data.

Lanjutan pada halaman berikutnya

Tabel 3.1 – Rekapitulasi tugas mingguan selama magang (Lanjutan)

Minggu Ke-	Pekerjaan yang Dilakukan
6	Melanjutkan mengerjakan dan memperbaiki bug dalam penyusunan form input untuk kebutuhan Create dan Update data pada Master Data
7	Melanjutkan mengerjakan dan memperbaiki bug dalam penyusunan form input untuk kebutuhan Create dan Update data pada Master Data. Menambahkan keyboard awareness juga.
8	Menyesuaikan tampilan form aplikasi di PO dan SO berdasarkan hasil uji coba sementara sesuai permintaan dari client
9	Melakukan revisi desain tampilan aplikasi berdasarkan masukan klien, termasuk penyesuaian form, dan tombol pada Master Data.
10	Mengimplementasikan dan memperbaiki fitur sorting data pada beberapa halaman Master Data seperti Customer, Product, dan Site, agar data dapat diurutkan berdasarkan nama atau kategori.
11	Melakukan perbaikan bug pada button shorting pada modul Master Data, Serta revisi beberapa form berdasarkan klien, termasuk penyesuaian tombol.
12	Memperbaiki bug button delete pada Master Data dan juga bug saat membuka modul Master data yang ada di di page Dashboard
13	Memperbaiki button bug pada saat melakukan edit data di page Master Data tidak bisa terupdate secara langsung
14	Melakukan perbaikan minor pada tampilan form model yang tidak full aplikasi pada Master Data.
15	Mengimplementasikan fitur kalender pada form di Master Data
16	Melakukan revisi rumus total yang belum bisa terjumlah pada page PO

Lanjutan pada halaman berikutnya

Tabel 3.1 – *Rekapitulasi tugas mingguan selama magang (Lanjutan)*

Minggu Ke-	Pekerjaan yang Dilakukan
17	Melakukan pengecekan akhir tampilan aplikasi dan berdiskusi dengan pembimbing lapangan sebagai persiapan penyusunan laporan magang.

3.4 Perangkat Penunjang Pelaksanaan Magang

Selama proses kegiatan magang, pemanfaatan berbagai perangkat lunak (*software*), perangkat keras (*hardware*), serta kerangka kerja dan pustaka menjadi komponen penting dalam pembuatan Aplikasi Mobile ERP. Adapun penjelasan mengenai masing-masing komponen tersebut adalah sebagai berikut.

3.4.1 Perangkat Lunak yang Digunakan

Tabel 3.2. Daftar perangkat lunak, spesifikasinya (versi), dan kegunaannya

Nama Perangkat	Spesifikasi (Versi)	Kegunaan
Visual Studio Code (VS Code)	1.90.1	<i>Code Editor</i>
Git	2.45.2.windows.1	<i>Version Control System</i>
React Native		Pembuat aplikasi mobile
TypeScript		Bahasa pemrograman utama

3.4.2 Perangkat Keras yang Digunakan

Tabel 3.3. Daftar perangkat keras dan spesifikasinya

Nama Perangkat	Spesifikasi
Laptop Pribadi	Huawei Matebook D 14
Processor	13th Gen Intel(R) Core(TM) i5-13420H (2.10 GHz))
RAM (Random Access Memory)	16 GB

Lanjutan pada halaman berikutnya

Tabel 3.3 – Daftar perangkat keras dan spesifikasinya (Lanjutan)

Nama Perangkat	Spesifikasi
Storage	477GB
Operating System	Windows 11 Home 64-bit
Display	1920×1200 piksel (Full HD+)
GPU (Graphics Processing Unit)	Intel UHD Graphics

3.4.3 Kerangka Kerja dan Pustaka yang Digunakan

Tabel 3.4. Daftar kerangka kerja & pustaka beserta versinya

Nama	Versi	Nama	Versi
@eva-design/eva	2.2.0	axios	1.13.2
@expo-google-fonts/montserrat	0.4.1	babel-plugin-module-resolver	5.0.2
@expo-google-fonts/open-sans	0.4.1	date-fns	4.1.0
@expo/react-native-action-sheet	4.1.1	expo	54.0.2
@expo/vector-icons	15.0.2	expo-blur	15.0.7
@lucide/lab	0.1.2	expo-camera	17.0.7
@react-native-async-storage/async-storage	2.2.0	expo-constants	18.0.8
@react-native-community/datetimepicker	8.4.5	expo-dev-client	6.0.12

Lanjutan pada halaman berikutnya

Nama	Versi	Nama	Versi
@react-native-picker/picker	2.11.1	expo-document-picker	14.0.7
@react-navigation/bottom-tabs	7.2.0	expo-file-system	19.0.20
@react-navigation/native	7.0.14	expo-font	14.0.8
@types/react-native	0.72.8	expo-haptics	15.0.7
@ui-kitten/components	5.3.1	expo-linear-gradient	15.0.7
firebase	12.1.0	expo-linking	8.0.8
firebase-admin	13.6.0	expo-print	15.0.7
lucide-react-native	0.475.0	expo-router	6.0.1
react	19.1.0	expo-secure-store	15.0.7
react-dom	19.1.0	expo-sharing	14.0.7
react-native	0.81.4	expo-splash-screen	31.0.9
react-native-chart-kit	6.12.0	expo-status-bar	3.0.8
react-native-collapsible	1.6.2	expo-symbols	1.0.7
react-native-date-picker	5.0.13	expo-system-ui	6.0.7
react-native-elements	3.4.3	expo-web-browser	15.0.7
react-native-eval-icons	1.3.1	react-native-fs	2.20.0
react-native-gesture-handler	2.28.0	react-native-gifted-charts	1.4.63
react-native-keyboard-aware-scroll-view	0.9.5	react-native-modal	14.0.0
react-native-modal-datetime-picker	18.0.0	react-native-paper	5.14.5

Lanjutan pada halaman berikutnya

Nama	Versi	Nama	Versi
react-native-reanimated	4.1.0	react-native-safe-area-context	5.6.0
react-native-screens	4.16.0	react-native-svg	15.12.1
react-native-url-polyfill	2.0.0	react-native-vector-icons	10.3.0
react-native-web	0.21.0	react-native-webview	13.15.0
react-native-worklets	0.5.1	xlsx	0.18.5

3.5 Proses Pelaksanaan Magang

Dalam pelaksanaan kegiatan kerja magang dilakukan ketika proses pengembangan aplikasi telah berjalan, sehingga proses ini melanjutkan pengembangan sistem yang telah dirancang sebelumnya oleh tim lainnya. Aplikasi dikembangkan menggunakan React Native sebagai framework utama karena mendukung pengembangan aplikasi mobile lintas platform secara efisien, serta TypeScript untuk meningkatkan keteraturan kode dan meminimalkan kesalahan melalui penerapan static typing. Pemilihan teknologi tersebut telah ditetapkan sejak awal pengembangan oleh tim, sehingga proses pelaksanaan magang mengikuti arsitektur dan teknologi yang telah digunakan. Pada aspek antarmuka dan pengalaman pengguna (UI/UX), dilakukan implementasi desain berdasarkan arahan supervisor dan kebutuhan klien, dengan tujuan memastikan tampilan dan alur penggunaan aplikasi sesuai dengan preferensi serta kenyamanan pengguna.

3.5.1 Flowchart ERP (Enterprise Resource Planning)

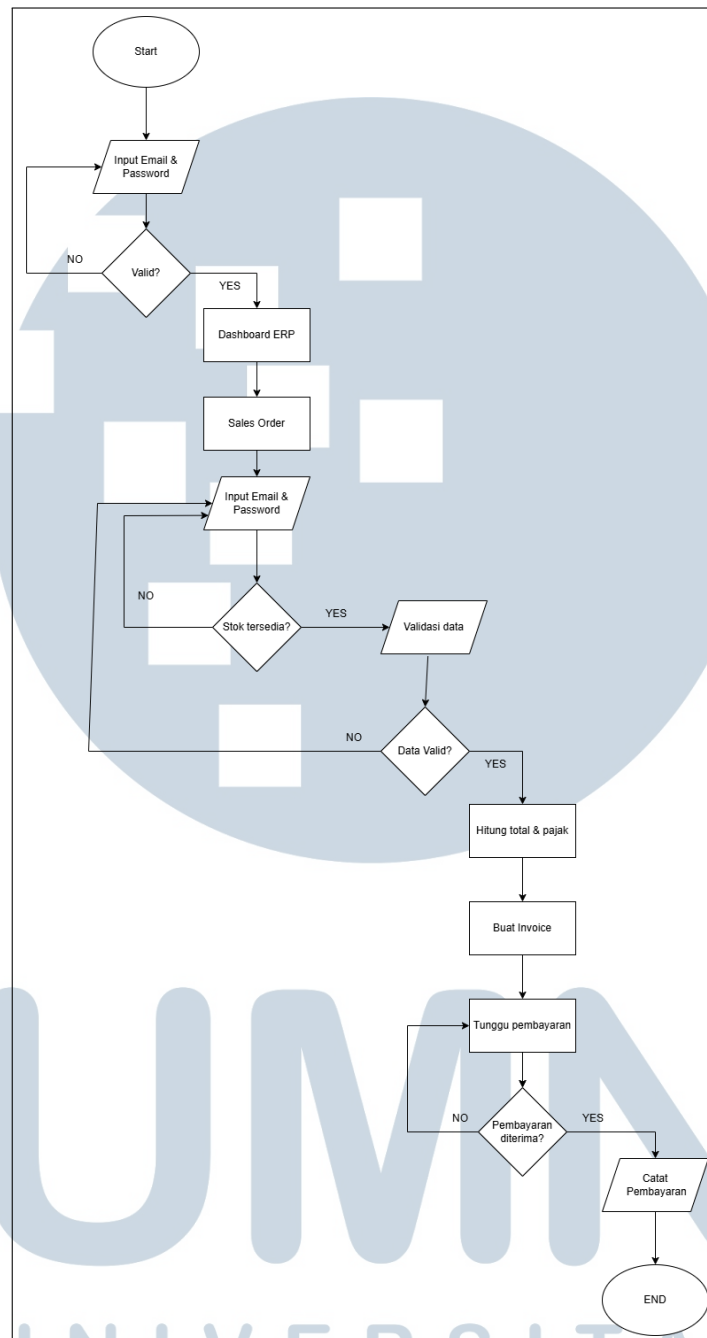
Flowchart sistem ERP (Enterprise Resource Planning) mobile ini telah menggambarkan alur kerja penggunaan dalam melakukan proses penjualan dengan terstruktur. Proses diawali dengan tahap *start*, kemudian pengguna harus lakukan input email dan password memasuki sistem. Kemudian melakukan proses validasi data login. Apabila data yang dimasukkan tidak valid, maka pengguna tidak dapat melanjutkan proses. Namun, jika data valid, pengguna diarahkan langsung ke halaman Dashboard ERP.

Pada Dashboard ERP, pengguna memilih menu Sales Order untuk melakukan transaksi penjualan. Setelah itu pengguna menginput data customer dan produk yang akan dipesan. Kemudian sistem akan melakukan pengecekan adakah ketersediaan untuk stok ada atau tidak tersedia. Apabila stok tidak tersedia, maka proses transaksi tidak dapat dilanjutkan. Jika stok tersedia, sistem akan mengarahkan pengguna untuk melanjutkan ke tahap validasi data transaksi.

Setelah pengecekan data transaksi dinyatakan valid, sistem akan segera memulai proses perhitungan total pembayaran dan pajak. Setelah itu sistem akan membuatkan invoice untuk pengguna sebagai bukti transaksi dan akan masuk ke tahap menunggu pembayaran dari pelanggan. Setelah itu Sistem kemudian melakukan pengecekan apakah pembayaran telah diterima atau tidak.

Jika pembayaran telah diterima, maka sistem akan melanjutkan ke proses pencatatan pembayaran. Setelah seluruh proses selesai, alur kerja sistem diakhiri pada tahap *end*, yang berarti bahwa semua proses transaksi penjualan pada sistem ERP telah selesai. Gambar 3.1 merupakan sebuah flowchartnya yaitu sebagai berikut.





Gambar 3.1. Flowchart sistem ERP berbasis aplikasi mobile.

A Implementasi Komponen Desain Aplikasi

Berdasarkan Gambar 3.1 Flowchart ERP Berbasis Aplikasi Mobile, setelah pengguna berhasil melakukan proses autentikasi, sistem akan menampilkan halaman dashboard utama. Dashboard ini berfungsi sebagai pusat navigasi yang

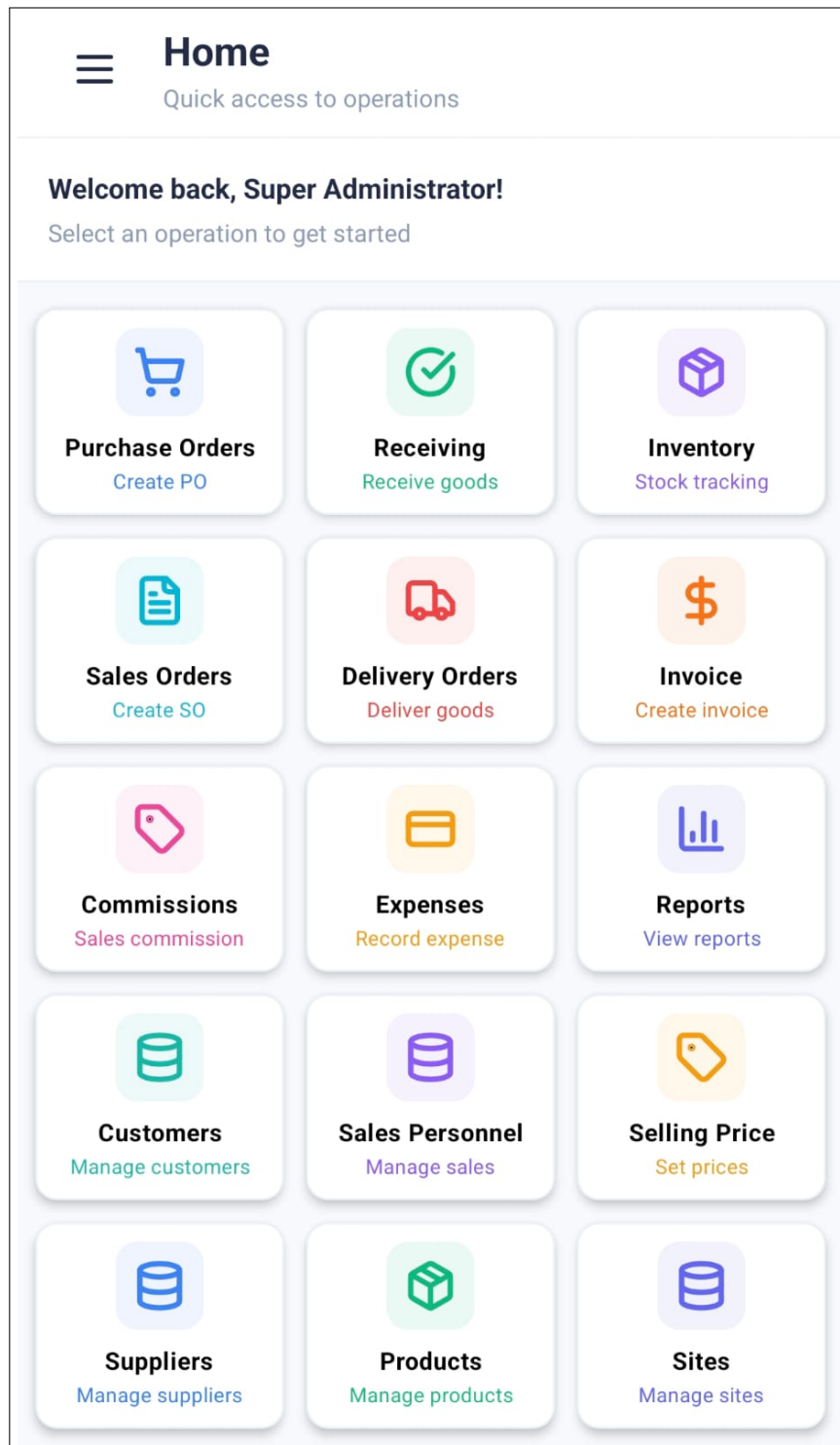
mengarahkan pengguna ke berbagai modul utama yang tersedia dalam aplikasi ERP.

Serta pada tampilan dashboard, terdapat komponen modul Sales Order disajikan dalam bentuk menu atau tombol yang dapat dipilih oleh pengguna sesuai dengan kebutuhan operasional. Sales Order yang dapat digunakan oleh pengguna untuk melakukan proses pembelian.

Implementasi komponen desain dashboard dilakukan menggunakan React Native. Hal ini memungkinkan konsistensi pada setiap bagian desain akan dikembangkan secara terpisah sehingga lebih mudah dalam proses pengelolaan, pemeliharaan, dan pengembangan lanjutan. Desain desain juga disesuaikan dengan alur sistem yang telah dirancang pada flowchart untuk memastikan kemudahan penggunaan dan konsistensi navigasi.

Untuk Gambar 3.2 merupakan sebuah implementasi desain aplikasi, berikut ditampilkan contoh tampilan dashboard utama aplikasi Mobile ERP.





Gambar 3.2. Tampilan *dashboard* aplikasi mobile ERP.

B Implementasi Modul ERP

Implementasi modul ERP berbasis mobile telah dilakukan setelah pengguna berhasil melakukan proses autentikasi dan masuk ke halaman dashboard utama. Dashboard berfungsi sebagai pusat navigasi yang mengarahkan pengguna ke berbagai modul utama dalam sistem ERP, yaitu modul Master Data, Transaksi, Keuangan, dan Sistem.

Setiap modul dirancang untuk mendukung kebutuhan operasional perusahaan dengan tampilan desain sangat terstruktur dan mudah digunakan. Modul digunakan untuk dapat menampilkan informasi keuangan, laporan, dan komisi berdasarkan data transaksi yang telah diproses sebelumnya. Sementara itu, modul Sistem menyediakan fitur pendukung seperti pengelolaan sistem, ekspor dan impor data, serta fungsi pemeliharaan aplikasi.

Sebagai Front-End Developer, dilakukan implementasi desain untuk pengguna pada setiap modul ERP agar sesuai dengan kebutuhan pengguna dan permintaan klien. Proses implementasi ini telah difokuskan pada tampilan halaman, pengaturan navigasi antar modul, serta peningkatan pengalaman pengguna melalui penyesuaian tata letak, pengembangan form pada aplikasi mobile yang dibangun menggunakan React Native dan TypeScript.

B.1 Implementasi Modul Master Data

Modul *Master Data* merupakan salah satu modul penting dalam ERP berbasis mobile ini yang digunakan untuk mengelola data yang menjadi acuan bagi seluruh proses bisnis. Serta Modul ini berfungsi sebagai tempat pengolahan data penting yang akan digunakan pada modul lainnya.

Modul Master Data menyediakan beberapa submenu, antara lain *Suppliers*, *Customers*, *Sites*, *Products*, *Sales Personnel*, *Shipping Agents*, *Selling Price*, *Expense Categories*, *Creditors*, dan *Users*. Setiap submenu memiliki kegunaannya masing-masing serta untuk memudahkan pengguna dalam mengelola data sesuai dengan kategorinya masing-masing.

Pada modul ini, mahasiswa berperan untuk mengimplementasikan desain pengolahan data (*Create*, *Read*, *Update*, *Delete*) yang juga membuat pengguna dapat melakukan proses penambahan, penampilan, perubahan, serta penghapusan data melalui aplikasi mobile. Proses tersebut dibutuhkan dalam bentuk formulir input, daftar data, serta tombol aksi yang dibuat sesuai dengan kebutuhan pengguna.

Implementasi Modul Master Data difokuskan pada penyusunan tampilan desain yang konsisten, mudah dipahami, serta nyaman digunakan pada perangkat mobile. Mahasiswa juga melakukan penyesuaian tata letak, komponen visual, dan alur interaksi pengguna agar proses pengolahan data dapat berjalan dengan lebih efektif.

B.1.1 Implementasi Fitur CRUD pada Master Data

Fitur *Create, Read, Update, Delete* (CRUD) merupakan sebuah fungsi dasar dalam pengelolaan data suatu bisnis yang membantu pengguna untuk menambahkan data baru, menampilkan data yang telah tersimpan, serta dapat memperbarui data yang ada, juga menghapus data yang tidak lagi diperlukan. Pada aplikasi Mobile ERP, fitur CRUD diterapkan pada modul *Master Data* untuk mengelola data utama seperti *Customer, Supplier, Product*, dan *Site*.

Fitur *Read* yang terdapat di modul Master Data Supplier digunakan untuk menampilkan data supplier yang telah tersimpan pada basis data di aplikasi Mobile ERP. Proses ini dilakukan dengan mengambil data dari layanan *Firebase* serta menampilkannya dalam bentuk daftar yang dapat diperbarui, difilter, dan diurutkan sesuai kebutuhan pengguna. Gambar 3.3 tampilan dari fitur read yaitu sebagai berikut. Kode 3.1 merupakan implementasi logika pemrograman yang digunakan untuk menampilkan data supplier dari basis data ke dalam antarmuka aplikasi pada modul *master data*, sehingga pengguna dapat melihat informasi supplier secara terstruktur.

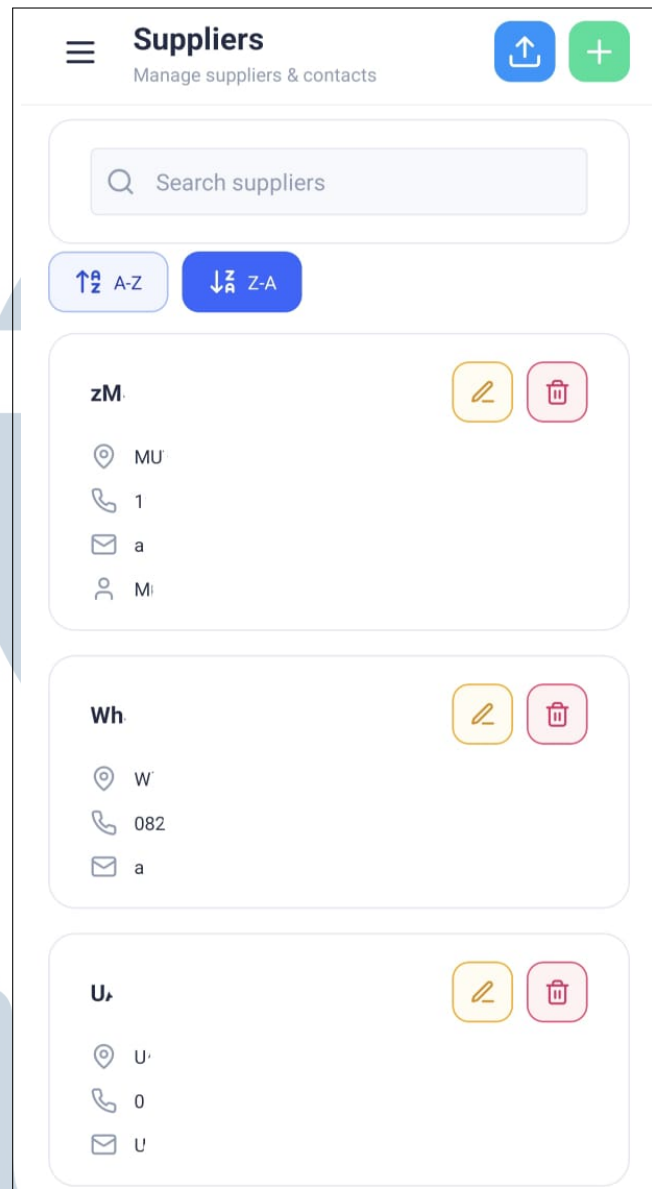
```
1  const loadSuppliers = async () => {  
2    try {  
3      const data = await firebaseDataService.getSuppliers();  
4      setSuppliers(data);  
5    } catch {  
6      Alert.alert('Error', 'Failed to load suppliers');  
7    } finally {  
8      setLoading(false);  
9    }  
10 };  
11  
12 // Refresh Data  
13 const onRefresh = async () => {  
14   setRefreshing(true);
```

```

15   await loadSuppliers();
16   setRefreshing(false);
17 };
18
19 // Sorting Data Supplier
20 const sortSuppliers = (list: Supplier[]) => {
21   return [...list].sort((a, b) => {
22     if (sortAsc)
23       return a.nama_supplier.localeCompare(b.nama_supplier);
24     else
25       return b.nama_supplier.localeCompare(a.nama_supplier);
26   });
27 };
28
29 // Filter Data Supplier
30 const filterSuppliers = () => {
31   if (!searchQuery.trim()) {
32     setFilteredSuppliers(sortSuppliers(suppliers));
33     return;
34   }
35
36   const q = searchQuery.toLowerCase();
37   setFilteredSuppliers(
38     sortSuppliers(
39       suppliers.filter(s =>
40         s.nama_supplier.toLowerCase().includes(q) ||
41         s.alamat.toLowerCase().includes(q) ||
42         s.telepon.includes(searchQuery) ||
43         (s.email ? s.email.toLowerCase().includes(q) : false
44       )
45     )
46   );

```

Kode 3.1: Implementasi fitur *read* data supplier.



Gambar 3.3. Tampilan *read* supplier pada modul *master data*.

Fitur *Create* juga merupakan bagian dari konsep CRUD yang memiliki fungsi untuk menambahkan data baru ke dalam sistem. Pada modul Master Data Supplier, fitur ini memungkinkan pengguna untuk melakukan input informasi supplier baru lewat sebuah form input yang akan ada dalam bentuk modal. Gambar 3.4 tampilan dari fitur *create*. Kode 3.2 merupakan implementasi logika pemrograman yang digunakan untuk menambahkan dan memperbarui data supplier pada modul *master data* melalui mekanisme validasi input, pengelolaan formulir, serta penyimpanan data ke dalam basis data.

```

1  const openModal = (supplier) => {
2    if (supplier) {
3      // Mode: EDIT
4      setEditingSupplier(supplier);
5      setFormData({
6        nama_supplier: supplier.nama_supplier || '',
7        alamat: supplier.alamat || '',
8        telepon: supplier.telepon || '',
9        email: supplier.email || '',
10       npwp: supplier.npwp || '',
11       nama_cp1: supplier.nama_cp1 || '',
12       telepon_cp1: supplier.telepon_cp1 || '',
13       email_cp1: supplier.email_cp1 || '',
14       nama_cp2: supplier.nama_cp2 || '',
15       telepon_cp2: supplier.telepon_cp2 || '',
16       email_cp2: supplier.email_cp2 || '',
17       keterangan: supplier.keterangan || ''
18     });
19   } else {
20     // Mode: CREATE
21     setEditingSupplier(null);
22     setFormData({
23       nama_supplier: '',
24       alamat: '',
25       telepon: '',
26       email: '',
27       npwp: '',
28       nama_cp1: '',
29       telepon_cp1: '',
30       email_cp1: '',
31       nama_cp2: '',
32       telepon_cp2: '',
33       email_cp2: '',
34       keterangan: ''
35     });
36   }
37   clearNameValidation();

```

```

38   setModalVisible(true);
39   requestAnimationFrame(animateIn);
40 };
41
42 const closeModal = () => {
43   animateOut(() => {
44     setModalVisible(false);
45     setEditingSupplier(null);
46   });
47 };
48
49 // Handle perubahan input nama supplier
50 const handleNameChange = (text) => {
51   setFormData({ ...formData, nama_supplier: text });
52   validateSupplierName(text);
53 };
54
55 // Simpan Data (Create atau Update)
56 const handleSave = async () => {
57   if (!formData.nama_supplier || !formData.alamat || !
58     formData.telepon || !formData.email) {
59     Alert.alert('Validation', 'Please fill all required
60     fields');
61     return;
62   }
63
64   if (hasNameError) {
65     Alert.alert('Error', nameValidationError);
66     return;
67   }
68
69   const validation = await ValidationService.
70     validateUniqueness(
71     'supplier',
72     formData.nama_supplier,
73     editingSupplier?.id_supplier
74   );

```

```

73     if (!validation.isValid) {
74         Alert.alert('Error', validation.message);
75         return;
76     }
77
78     setSaving(true);
79     try {
80         if (editingSupplier) {
81             await firebaseDataService.updateSupplier(
82                 editingSupplier.id_supplier,
83                 formData
84             );
85         } else {
86             await firebaseDataService.addSupplier({
87                 ...formData,
88                 input_by: 'system',
89                 input_date: new Date().toISOString()
90             });
91         }
92         await loadSuppliers();
93         closeModal();
94     } catch {
95         Alert.alert('Error', 'Failed to save supplier data');
96     } finally {
97         setSaving(false);
98     }
99 };

```

Kode 3.2: Implementasi fitur *create* data supplier

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Add Supplier X

Supplier Name *

Enter supplier name

Address *

Enter address

Phone *

Enter phone number

Email *

Enter email

NPWP

Enter NPWP (optional)

Contact Person 1

Enter contact person name

Phone CP1

Enter contact person phone

Email CP1

Enter contact person email

Gambar 3.4. Tampilan *form create data supplier* pada modul *master data*.

Fitur *Delete* merupakan bagian dari konsep CRUD yang memiliki fungsi untuk menghapus data yang sudah tidak diperlukan dari dalam sistem. Pada modul Master Data Supplier, fitur ini membantu pengguna untuk menghapus data supplier yang dipilih setelah melalui proses konfirmasi terlebih dahulu, hal ini dapat mencegah terjadinya penghapusan data secara tidak sengaja. Gambar 3.5 tampilan dari fitur delete. Kode 3.3 merupakan implementasi logika pemrograman yang berfungsi untuk menghapus data supplier dari basis data melalui modul *master data* berdasarkan aksi yang dilakukan oleh pengguna.

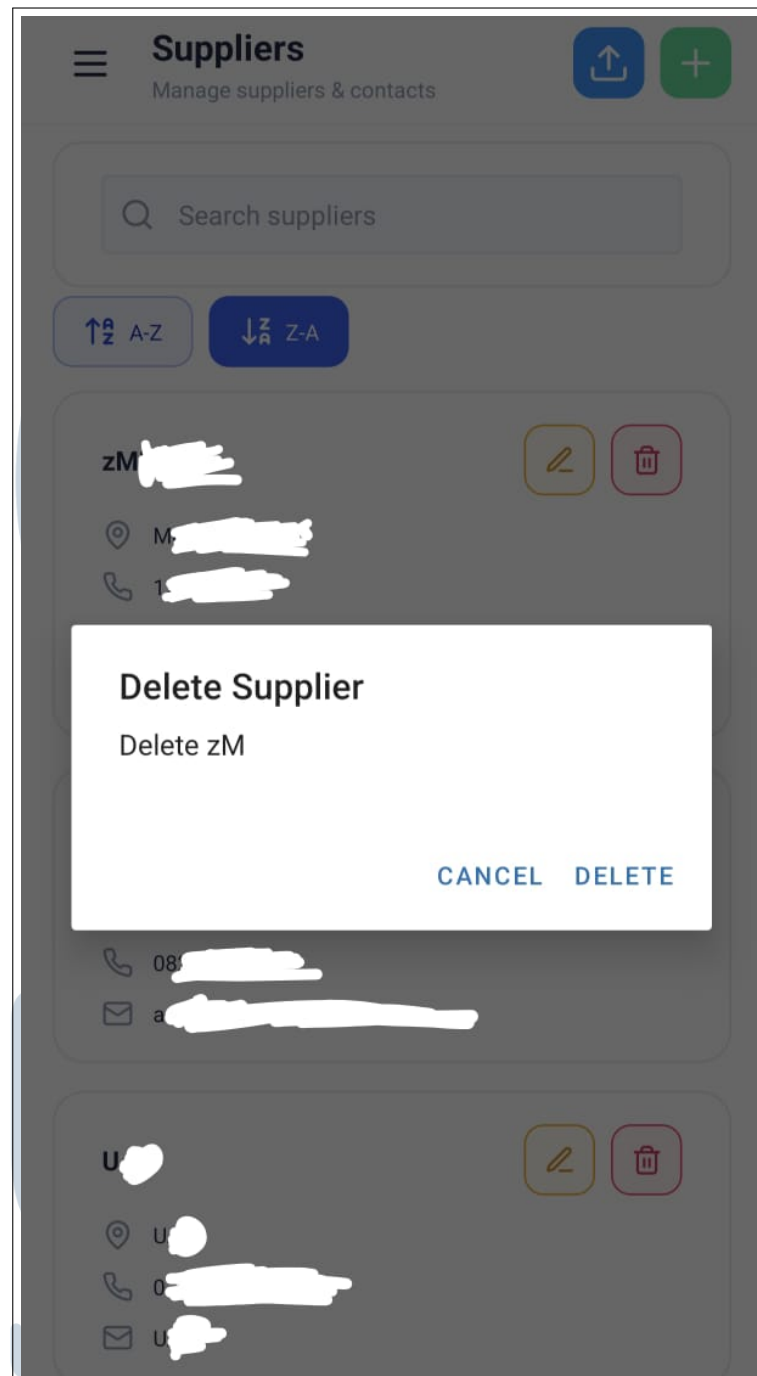
```

1  const handleDelete = (supplier: Supplier) => {
2    Alert.alert(
3      'Delete Supplier',
4      `Delete ${supplier.nama_supplier}?`,
5      [
6        { text: 'Cancel', style: 'cancel' },
7        {
8          text: 'Delete',
9          style: 'destructive',
10         onPress: async () => {
11           try {
12             await firebaseDataService.deleteSupplier(
13               supplier.id_supplier
14             );
15             await loadSuppliers(); // Reload data setelah
hapus
16           } catch {
17             Alert.alert(
18               'Error',
19               'Failed to delete supplier'
20             );
21           }
22         }
23       ]
24     );
25   };
26 };

```

Kode 3.3: Implementasi fitur *delete* data supplier.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.5. Tampilan *delete* supplier pada modul *master data*.

B.1.2 Implementasi Fitur Shorting pada Master Data

Fitur *Sorting* bisa digunakan untuk mengurutkan data supplier berdasarkan kriteria-kriteria tertentu dapat membantu pengguna dalam mencari dan membaca informasi. Pada modul Master Data Supplier, fitur ini juga membantu

pengguna untuk mengurutkan data berdasarkan nama supplier secara alfabet, baik secara menaik (ascending) maupun menurun (descending), sehingga data dapat ditampilkan dengan lebih terstruktur. Gambar 3.6 dan 3.7 menampilkan fitur *sorting* data supplier. Kode 3.4 merupakan implementasi logika pemrograman yang berfungsi untuk mengatur pengurutan data supplier berdasarkan nama secara alfabetis dengan mendukung mode *ascending* (A–Z) dan *descending* (Z–A).

```
1 // STATE - Menyimpan status urutan sorting
2 const [sortAsc, setSortAsc] = useState(true);
3 // true = A-Z (Ascending), false = Z-A (Descending)
4
5 // LOGIC FUNCTION - Fungsi untuk mengurutkan data supplier
6 const sortSuppliers = (list: Supplier[]) => {
7   return [...list].sort((a, b) => {
8     if (sortAsc) {
9       // Mode A-Z (Ascending)
10      return a.nama_supplier.localeCompare(b.nama_supplier);
11    } else {
12      // Mode Z-A (Descending)
13      return b.nama_supplier.localeCompare(a.nama_supplier);
14    }
15  });
16 };
17
18 // FILTER + SORT INTEGRATION
19 // Menggabungkan proses pencarian dan pengurutan data
20 const filterSuppliers = () => {
21   // Jika tidak terdapat kata kunci pencarian
22   if (!searchQuery.trim()) {
23     setFilteredSuppliers(sortSuppliers(suppliers));
24     return;
25   }
26
27   const q = searchQuery.toLowerCase();
28   setFilteredSuppliers(
29     sortSuppliers(
30       suppliers.filter(s =>
31         s.nama_supplier.toLowerCase().includes(q) ||
32         s.alamat.toLowerCase().includes(q) ||
```

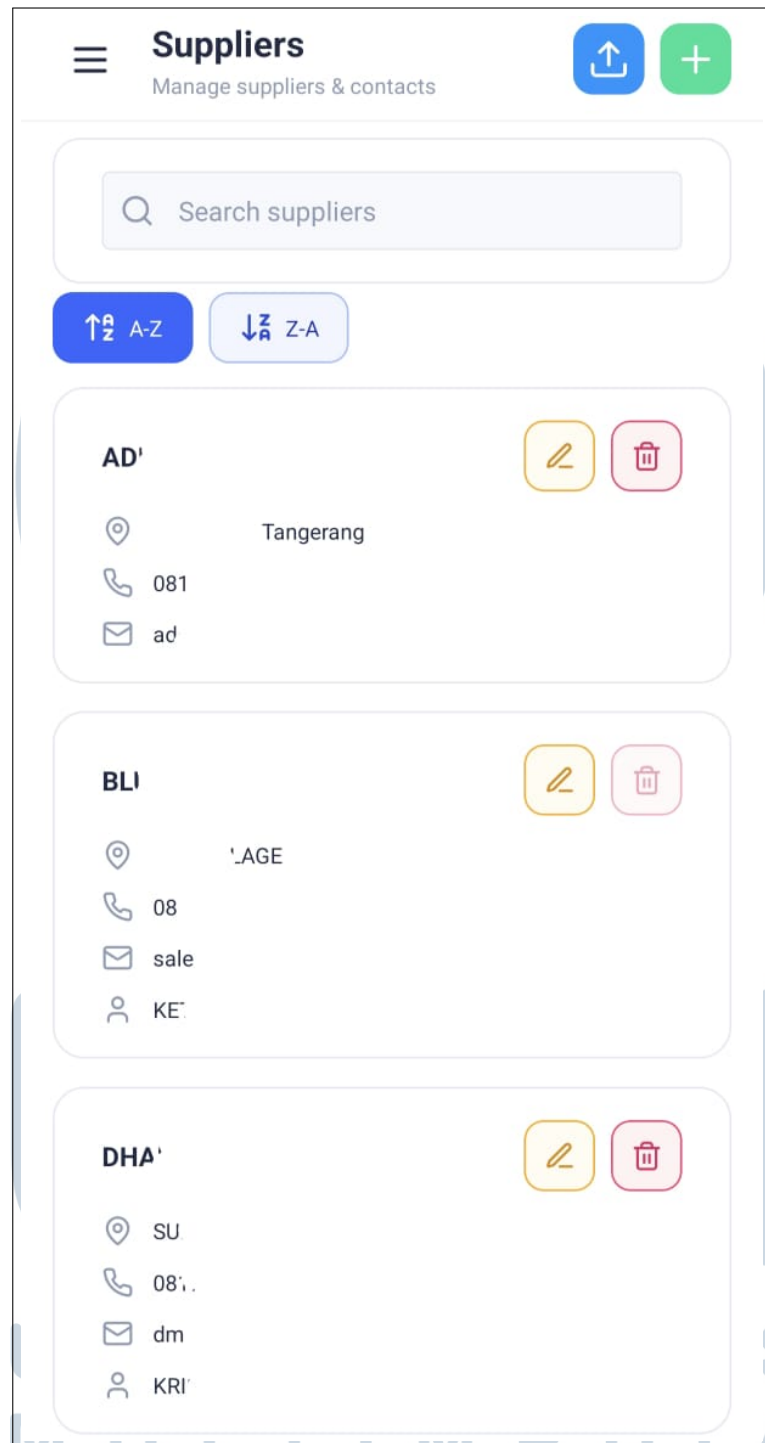
```

33         s.telepon.includes(searchQuery) ||
34         (s.email ? s.email.toLowerCase().includes(q) : false
35     )
36 )
37 );
38 };
39
40 // AUTO-UPDATE
41 Menjalankan ulang filter dan sorting ketika terjadi
    perubahan
42 useEffect(() => {
43     filterSuppliers();
44 }, [suppliers, searchQuery, sortAsc]);
45
46 // USER INTERFACE
47 Tombol untuk mengatur urutan data A-Z dan Z-A
48 <View style={{ flexDirection: 'row', alignItems: 'center'
49     }}>
50     <TouchableOpacity onPress={() => setSortAsc(true)}>
51         <Text>A-Z</Text>
52     </TouchableOpacity>
53
54     <TouchableOpacity onPress={() => setSortAsc(false)}>
55         <Text>Z-A</Text>
56     </TouchableOpacity>
57 </View>

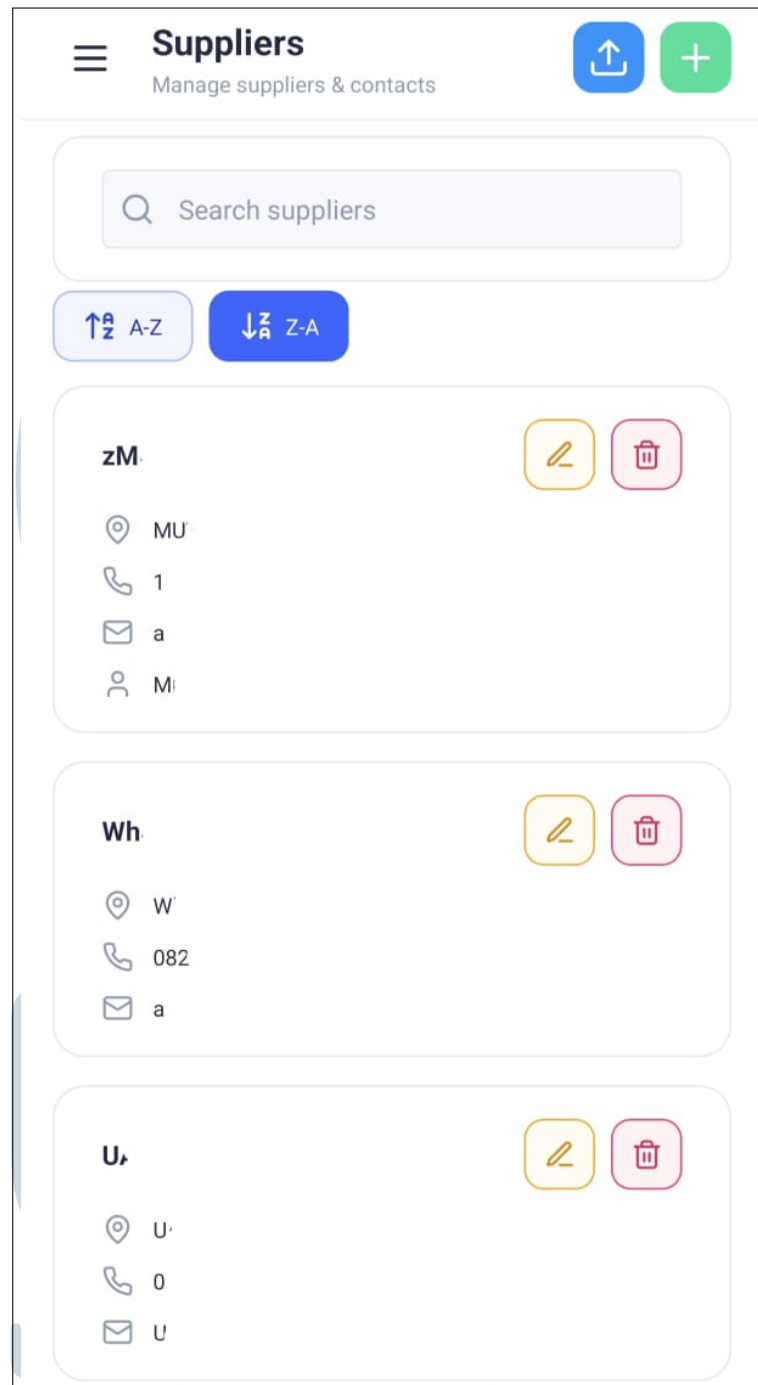
```

Kode 3.4: Implementasi fitur *sorting* data supplier.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.6. Tampilan fitur *sorting* data supplier secara A-Z



Gambar 3.7. Tampilan fitur *sorting* data supplier secara Z–A

3.5.2 Pengembangan Sistem Menggunakan Pendekatan Agile

Metodologi pengembangan sistem yang digunakan dalam pembangunan aplikasi Mobile Enterprise Resource Planning (ERP) ini menggunakan pendekatan

Agile. Metode Agile ini sangat cocok dengan perusahaan magang sebab perusahaan magang mahasiswa diminta klien secara langsung untuk membuat sebuah system program ERP berbasis mobile dan harus sesuai dengan kriteria dari klien sendiri. Pendekatan menggunakan metode Agile ini didasarkan pada karakteristik kebutuhan proyek yang bersifat fleksibel, serta banyaknya perubahan dan keinginan sendiri dari pihak klien selama proses pengembangan berlangsung. Pendekatan ini sangat membantu tim pengembang agar mudah untuk beradaptasi secara cepat terhadap perubahan apapun yang terjadi tanpa harus mengulang keseluruhan proses pengembangan dari tahap awal.

Pendekatan Agile yang di terapkan termasuk dalam pendekatan Scrum karena sangat mendukung pertemuan klien yang sangat rutin dan juga keterlibatan klien yang aktif saat proses pengembangan. Pendekatan ini membuat pengolahan tugas menjadi lebih fleksibel.

Sebagai Front-End Developer mahasiswa berperan dalam mengimplementasikan desain yang telah disusun oleh tim UI/UX serta permintaan klien. Proses implementasi dilakukan mahasiswa ini menggunakan framework React Native dan bahasa pemrograman TypeScript, yang mencakup pengembangan untuk form input, penerapan fitur Create, Read, Update, dan Delete (CRUD), serta penambahan fitur pendukung seperti kalender. Seluruh pengembangan difokuskan pada konsistensi desain, kemudahan navigasi, dan kenyamanan pengalaman pengguna (user experience) pada perangkat mobile.

3.6 Kendala dan Solusi yang Ditemukan

Selama pelaksanaan kegiatan magang dalam pengembangan aplikasi Mobile ERP, mahasiswa menghadapi beberapa kendala yang muncul dalam proses pengembangan tersebut. Akan tetapi Kendala tersebut kemudian diatasi dengan berupa solusi yang diterapkan agar pengembangan aplikasi tetap berjalan dengan baik dan sesuai dengan kebutuhan klien.

3.6.1 Kendala yang Ditemukan

Beberapa kendala yang dihadapi selama proses pengembangan aplikasi Mobile ERP antara lain sebagai berikut:

1. Terjadinya bug atau kesalahan sistem yang tidak terduga pada saat proses pengembangan dan pengujian aplikasi.

2. Kesulitan dalam manajemen waktu akibat adanya permintaan revisi dari klien yang bersifat mendadak.
3. Keterbatasan pemahaman teknis mahasiswa dalam mengimplementasikan kebutuhan dan keinginan klien ke dalam implementasi desain aplikasi.

3.6.2 Solusi yang Diterapkan

Untuk mengatasi kendala-kendala tersebut, telah diterapkan beberapa solusi sebagai berikut:

1. Melakukan proses *debugging* dan pengujian secara bertahap setiap kali terdapat perubahan atau penambahan fitur guna meminimalkan terjadinya bug.
2. Menyesuaikan prioritas pekerjaan dan menerapkan pola kerja yang lebih fleksibel agar revisi dari klien dapat diselesaikan tepat waktu.
3. Meningkatkan komunikasi dengan klien dan tim pengembang melalui diskusi dan klarifikasi kebutuhan, serta mempelajari kembali dokumentasi desain yang telah disediakan.

