

BAB III

PELAKSANAAN KERJA PROFESI

3.1 Bidang Kerja

Dalam kerangka program kerja profesi ini, praktikan akan bekerja dalam sebuah tim yang terdiri dari satu lead programmer, dua fullstack web programmer dan satu projek manager. Praktikan akan memainkan peran sebagai salah satu dari dua fullstack web programmer dalam tim ini. Tugas utama praktikan adalah merancang database, membuat tampilan dan mengembangkan aplikasi web penjualan dan distrisbusi ayam untuk klien, yaitu perusahaan Ayam Sri Lestari.

1. Peran dalam Tim

Sebagai Fullstack Web Programmer, praktikan akan berperan sebagai bagian integral dari tim yang dipimpin oleh seorang Lead Programmer. Kolaborasi dan koordinasi yang efektif di antara anggota tim akan menjadi kunci kesuksesan proyek ini. Praktikan akan berpartisipasi dalam pembahasan strategi pengembangan, pemecahan masalah teknis, serta implementasi keputusan dan arahan yang diberikan oleh Lead Programmer.

2. Tugas dan Tanggung Jawab Praktikan

Praktikan akan mengemban beberapa tugas kunci dalam proyek ini, termasuk:

- **Merancang Database:** Sebagai Fullstack Web Programmer, praktikan akan memainkan peran sentral dalam merancang struktur database yang mendukung efisiensi dan konsistensi dalam pengelolaan data terkait penjualan dan distribusi ayam. Hal ini mencakup tipe data dan attribute dari semua master table sekaligus relasi antar table.
- **Desain Mockup Tampilan:** Praktikan akan bertanggung jawab atas desain mockup antarmuka pengguna yang akan menghadirkan pengalaman pengguna yang intuitif dan menarik. Ini melibatkan pemahaman mendalam terhadap

kebutuhan pengguna dan implementasi prinsip-prinsip desain terbaik untuk menciptakan antarmuka yang menarik dan fungsional.

- **Pengembangan Aplikasi Web:** Dalam tahap pengembangan, praktikan akan aktif terlibat dalam mengimplementasikan logika bisnis, fungsionalitas, dan interaksi antarmuka. Hal ini mencakup pengkodean, pengujian unit, serta integrasi berbagai komponen sistem.

3. **Kolaborasi dengan Projek Manager**

Selama seluruh proses pengembangan, praktikan akan berkolaborasi dengan Projek Manager. Komunikasi yang efektif dengan Projek Manager adalah kunci untuk memastikan bahwa setiap tahap proyek sesuai dengan harapan dan kebutuhan klien. Praktikan akan menerima arahan dan masukan yang diberikan oleh Projek Manager untuk memastikan proyek berjalan sesuai dengan visi dan tujuan yang telah ditetapkan.

4. **Pengembangan Aplikasi Web Penjualan dan Distribusi Ayam**

Fokus proyek ini adalah pada pengembangan aplikasi web yang dapat memfasilitasi penjualan dan distribusi ayam secara efisien. Praktikan akan bekerja sama dengan anggota tim lainnya untuk memastikan bahwa fitur-fitur yang diimplementasikan sesuai dengan kebutuhan bisnis perusahaan Ayam Sri Lestari. Ini mencakup fungsi transaksi online, manajemen stok, pelacakan pesanan, dan berbagai elemen kunci lainnya.

Melalui peran praktikan dalam proyek ini, diharapkan bahwa hasil akhirnya akan menjadi solusi teknologi informasi yang handal, inovatif, dan sesuai dengan ekspektasi klien. Dengan fokus pada kolaborasi tim, pengembangan kualitas, dan penerapan prinsip-prinsip desain terbaik, proyek ini diharapkan dapat memberikan nilai tambah yang signifikan bagi perusahaan Ayam Sri Lestari.

Selain itu, praktikan akan berkolaborasi dengan rekan-rekan tim lainnya, termasuk lead programmer, fullstack web programmer lain dan projek manager. Tim akan bekerja sama dalam merancang,

mengembangkan, dan menguji aplikasi web penjualan dan distribusi ayam ini. Kerja sama tim yang baik adalah kunci keberhasilan proyek ini, dan komunikasi yang efektif antara semua anggota tim sangat penting untuk memastikan bahwa aplikasi web dapat memenuhi kebutuhan dan harapan klien, yaitu perusahaan Ayam Sri Lestari.

Selama pelaksanaan proyek, praktikan juga akan memiliki kesempatan untuk mengembangkan keterampilan pemrograman, perancangan database, desain tampilan, dan manajemen proyek yang berharga. Kerja praktik ini tidak hanya merupakan kesempatan untuk mengaplikasikan pengetahuan akademis dalam konteks nyata, tetapi juga untuk berkontribusi pada pengembangan solusi teknologi yang bermanfaat bagi perusahaan dan penggunanya.

3.2 Pelaksanaan Kerja

Praktikan memulai kerja profesi ini pada tanggal 19 Maret sampai 9 September 2023. Bekerja secara *Work From Home (WFH)* yang dimulai pada bulan Maret dimana praktikan dan perusahaan tempat praktikan bekerja mulai dengan tahap diskusi tentang proyek yang akan dikerjakan ini bersama dengan proyek manager. Disini proyek manager memiliki peran untuk menjabarkan seluruh kebutuhan serta keinginan dari klien terhadap aplikasi yang akan dirancang ini kepada seluruh tim pengembang, termasuk dengan praktikan sendiri.

Aplikasi yang dikembangkan, yakni "Aplikasi Manajemen Penjualan dan Distribusi Ayam," dirancang untuk memberikan solusi efisien dan terintegrasi dalam mengelola penjualan dan distribusi ayam pada Perusahaan Ayam Sri Lestari. Fitur-fitur utama yaitu manajemen produk di pusat dan cabang-cabang, pengelolaan stok di setiap cabang, dan manajemen penjualan di tingkat cabang. Tim juga memperhatikan fitur-fitur minor seperti pengiriman dari cabang ke pelanggan dan manajemen tabungan pelanggan, sesuai dengan kebutuhan dan keinginan dari klien. Dengan proses pengembangan aplikasi ini, aplikasi ini diharapkan mampu meningkatkan operasional sehari-hari perusahaan Ayam Sri Lestari.

Berikut adalah beberapa user requirement yang dijelaskan projek manager :

1. **Aplikasi dapat Memfasilitasi Pusat dan Cabang**

Aplikasi ini dirancang untuk memberikan fasilitas kepada kedua entitas, pusat dan cabang, dalam mengelola dan mengakses data terkait penjualan dan distribusi ayam. Fasilitas ini mencakup pengelolaan data produk, informasi cabang, dan transaksi antar pusat dan cabang.

2. **Aplikasi dapat Diakses Melalui Web**

Sejalan dengan kebutuhan fleksibilitas dan aksesibilitas, aplikasi ini dapat diakses melalui platform web. Kemampuan akses melalui web mempermudah pengguna, baik dari pusat maupun cabang, untuk mengelola dan memonitor aktivitas bisnis secara efisien dari berbagai lokasi.

3. **Pusat dapat Mengelola Data Tipe Produk**

Pusat memiliki hak akses untuk mengelola data tipe produk, termasuk menambah, mengedit, atau menghapus informasi terkait jenis atau kategori produk yang dijual di seluruh cabang.

4. **Pusat dapat Mengelola Data Produk**

Pusat juga dapat mengelola data produk secara menyeluruh, termasuk informasi produk, harga, dan atribut lainnya. Kemampuan ini memungkinkan pusat untuk menjaga konsistensi data produk di seluruh jaringan cabang.

5. **Pusat dapat Mengelola Data Cabang**

Pengelolaan data cabang menjadi fungsi penting pusat. Pusat memiliki wewenang untuk menambah, mengedit, atau menghapus informasi terkait cabang, termasuk alamat, kontak, dan detail lainnya.

6. **Pusat dapat Mentracking Transaksi Pengiriman Barang ke Cabang**

Pusat memiliki kemampuan untuk memantau dan melacak setiap transaksi pengiriman barang yang dilakukan ke cabang. Ini mencakup informasi terkait jumlah barang, tanggal pengiriman, dan rincian lainnya.

7. Cabang dapat Mengelola Informasi Cabang, Termasuk Data Karyawan

Cabang memiliki hak akses untuk mengelola informasi cabang, termasuk data karyawan yang bekerja di cabang tersebut. Fungsi ini memungkinkan cabang untuk menjaga data karyawan dan informasi cabangnya tetap terkini.

8. Cabang dapat Mengelola Data Customer

Cabang dapat mengelola informasi pelanggan, termasuk menambah, mengedit, atau menghapus data pelanggan. Hal ini memberikan fleksibilitas kepada cabang untuk memelihara basis data pelanggan sesuai dengan kebutuhan lokalnya.

9. Cabang dapat Mengelola Data Produk Cabang

Cabang memiliki kewenangan untuk mengelola data produk cabangnya sendiri, termasuk menambah atau memperbarui informasi produk yang tersedia di cabang.

10. Cabang dapat Mengelola Stok Cabang

Fungsi ini memungkinkan cabang untuk mengelola stok ayam di cabang, termasuk pembaruan stok, pemantauan tingkat persediaan, dan pemberitahuan jika stok mencapai batas tertentu.

11. Cabang dapat Bertransaksi ke Customer

Cabang memiliki kemampuan untuk melakukan transaksi penjualan langsung kepada pelanggan. Ini mencakup pembelian produk, penjualan, dan semua aspek transaksi antara cabang dan pelanggan.

12. Cabang dapat Meminta Pengiriman dari Pusat

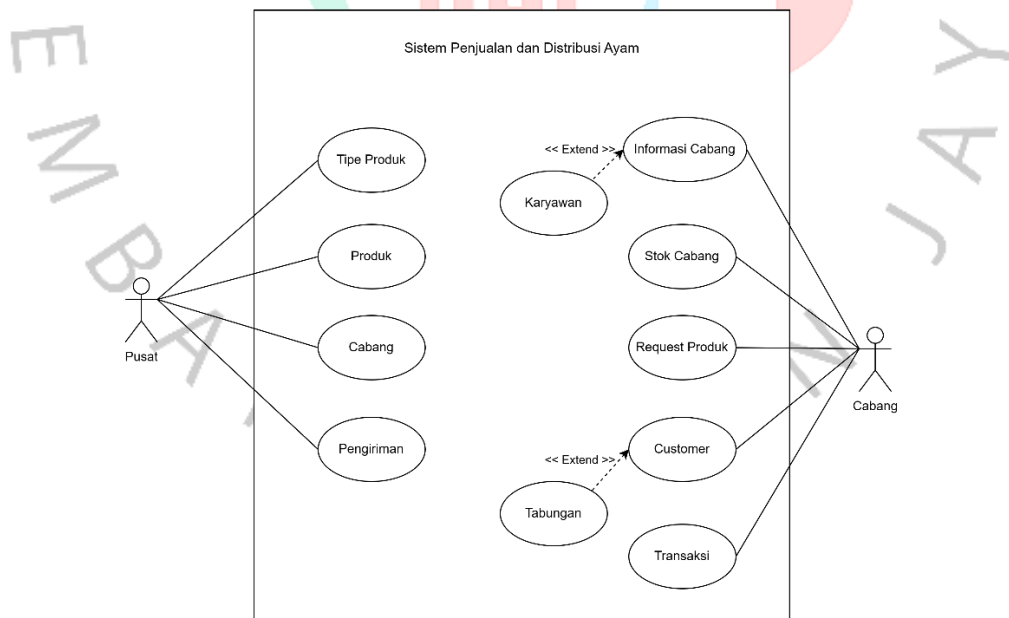
Cabang memiliki hak untuk meminta pengiriman stok dari pusat. Fungsi ini memungkinkan cabang untuk menjaga ketersediaan produk dengan mengajukan permintaan pengiriman sesuai dengan kebutuhan dan permintaan pelanggan.

13. Cabang dapat Mengelola Tabungan Customer

Cabang dapat mengelola tabungan pelanggan, mencakup informasi terkait saldo, transaksi tabungan, dan pembaruan lainnya. Hal ini memungkinkan cabang untuk memberikan layanan

tambahan kepada pelanggan melalui manajemen tabungan yang efisien.

Setelah mengetahui aplikasi apa yang akan dikembangkan serta semua user requirementnya, tim pengembangan memasuki tahap analysis serta pemahaman mendalam terhadap kebutuhan klien. Fokus utama adalah memahami dinamika bisnis, proses operasional, serta tujuan yang diinginkan melalui pengembangan aplikasi. Melibatkan seluruh anggota tim, termasuk praktikan, dalam sesi diskusi intensif untuk mengevaluasi dan merinci setiap aspek dari user requirement. Tujuan dari tahap ini adalah memastikan bahwa pemahaman yang seragam di antara anggota tim dan merancang solusi yang tidak hanya memenuhi kebutuhan pengguna tetapi juga mengatasi potensi masalah yang mungkin muncul. Tahap ini diwujudkan dalam pembuatan Usecase dan Entity Relationship Diagram (ERD) untuk merinci logika bisnis dan struktur database aplikasi. Berikut adalah Usecase dan ERD dari konsep awal aplikasi yang akan dibuat:

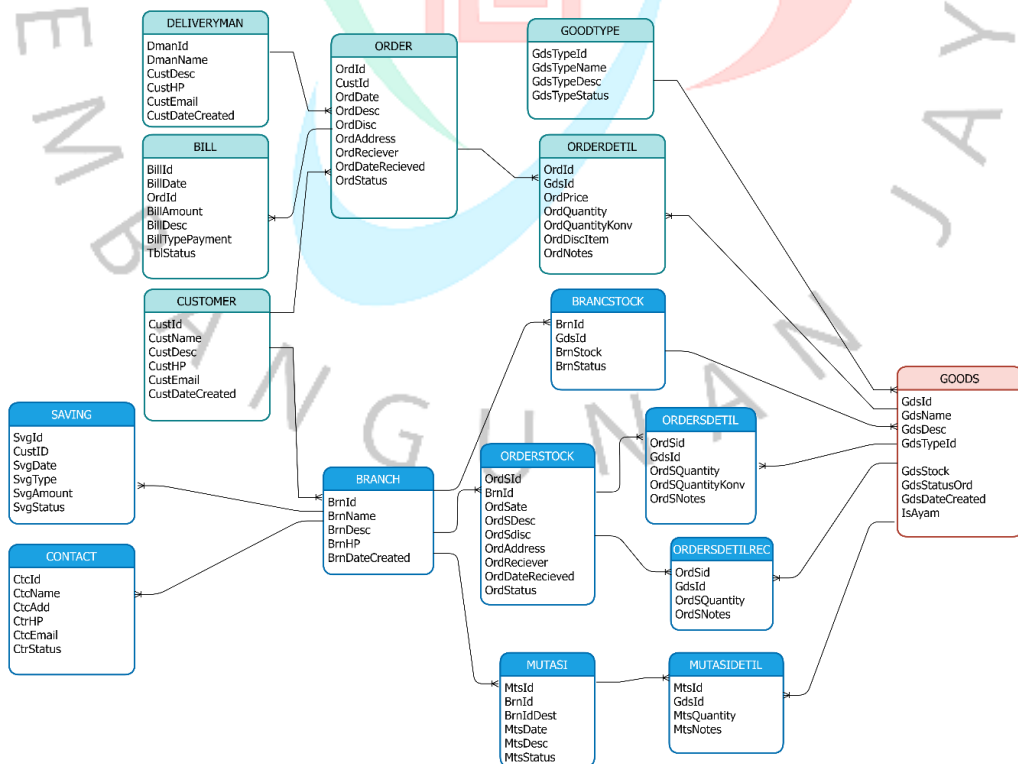


Gambar 3.1 Usecase Aplikasi Manajemen Penjualan dan Distribusi Ayam

Dalam diagram Usecase yang ada pada gambar 3.1, teridentifikasi dua actor dan beberapa usecase yang berperan dalam interaksi dengan sistem penjualan dan distribusi ayam yang akan dikembangkan.

Pertama-tama, terdapat Pusat sebagai aktor utama yang memiliki sejumlah use case. Pusat memiliki kemampuan untuk mengelola Tipe Produk, yaitu kategori-kategori produk yang ada, mengelola Produk dengan merinci informasi mengenai setiap produk, dan juga mengelola Cabang yang terdaftar di dalam sistem. Selain itu, Pusat memiliki kemampuan untuk memproses pengiriman berdasarkan permintaan yang diajukan oleh setiap cabang.

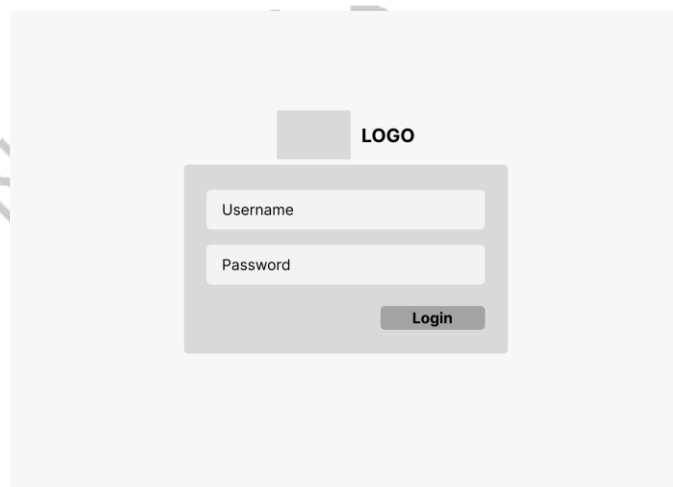
Selanjutnya, aktor kedua adalah Cabang, yang memiliki sejumlah use case yang terkait dengan manajemen informasi di tingkat cabang. Cabang dapat mengelola informasi cabang, dan di dalamnya terdapat ekstensi yang memungkinkan pengelolaan data karyawan cabang. Selain itu, Cabang juga dapat mengelola stok barang di cabangnya, melakukan permintaan produk dari Pusat, dan mengelola data pelanggan, dengan ekstensi khusus untuk fitur tabungan yang terdapat pada halaman customer. Yang terakhir, Cabang dapat melakukan Transaksi pada tingkat cabang masing-masing, mencakup kegiatan penjualan dan pembelian di lokasi tersebut.



Gambar 3.2 ERD Aplikasi Manajemen Penjualan dan Distribusi Ayam

Setelah tahap perencanaan, tim membuat mockup kasar aplikasi. Ini memberikan gambaran visual awal tentang desain antarmuka pengguna dan juga menjadi acuan untuk tahap pengembangan selanjutnya yaitu *Coding* atau Implementasi. Berikut adalah beberapa mockup yang praktingan dan tim rancang untuk aplikasi manajemen penjualan dan distribusi ayam :

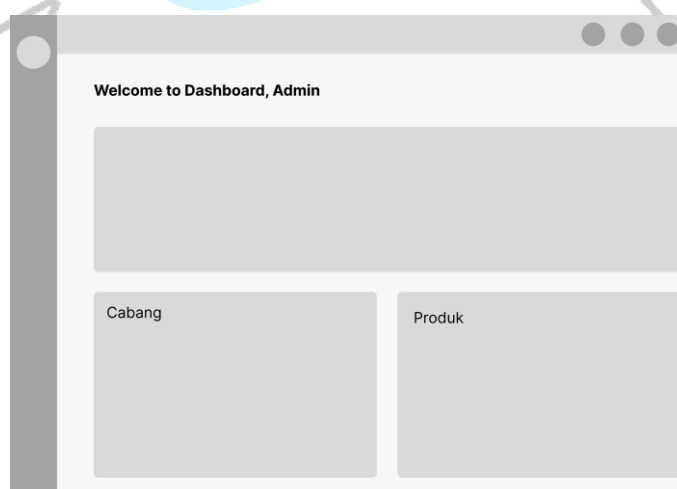
1. Mockup Login



Gambar 3.3 Mockup Login

Mockup login untuk aplikasi ini sangat simpel, dengan menampilkan logo perusahaan dan form login yang dilengkapi dengan kolom untuk username dan password, serta tombol login yang menonjol. Desainnya disusun agar memudahkan pengguna dalam berinteraksi dengan antarmuka aplikasi.

2. Mockup Dashboard

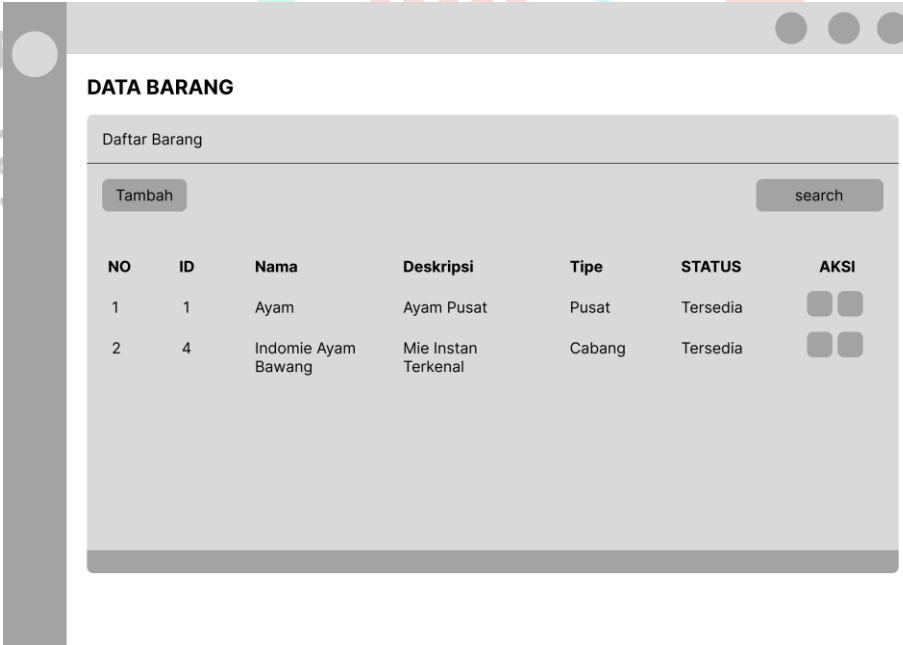


Gambar 3.4 Mockup Dashboard

Diatas adalah Mockup untuk halaman dashboard setelah user berhasil login ke aplikasi. Mockup ini mencakup sidebar yang menampilkan menu navigasi, mempermudah pengguna untuk mengakses berbagai fitur dan menu. Di pojok kanan, terdapat bagian profil yang memungkinkan pengguna untuk mengelola informasi profil mereka dan logout. Selain itu, mockup ini juga memuat beberapa elemen informatif seperti card statistik yang menampilkan data penting seperti statistik umum, jumlah cabang, dan jumlah barang atau produk. Desain ini diharapkan memberikan pengalaman pengguna yang intuitif dan memberikan akses cepat kepada informasi kunci setelah login.

Semua Master data seperti data barang, data cabang, data customer dan lainnya dibuat pada halaman index. Setiap halaman data yang menampilkan table dengan tombol tambah data dan action update dan delete data, yang contoh mockupnya yaitu halaman index data barang seperti berikut :

3. Mockup Index Data Barang

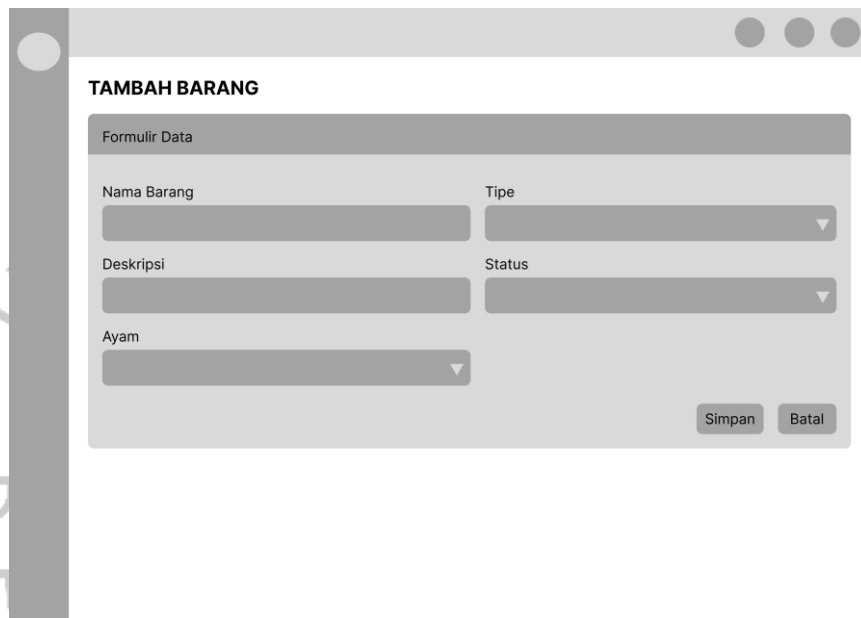


NO	ID	Nama	Deskripsi	Tipe	STATUS	AKSI
1	1	Ayam	Ayam Pusat	Pusat	Tersedia	<input type="checkbox"/> <input type="checkbox"/>
2	4	Indomie Ayam Bawang	Mie Instan Terkenal	Cabang	Tersedia	<input type="checkbox"/> <input type="checkbox"/>

Gambar 3.5 Mockup Index master data Barang

Diatas adalah mockup untuk halaman index data barang. Menyajikan informasi data dalam bentuk table, yang dapat diurutkan (sorting), dengan beberapa tombol seperti tombol tambah data serta aksi Edit dan Delete disetiap kolom data. Dan tampilan kolom search untuk fitur pencarian data.

4. Mockup Tambah Data Barang



The image shows a mockup of a 'TAMBAH BARANG' (Add Item) form. The form is titled 'Formulir Data' and contains several input fields: 'Nama Barang' (Item Name) as a text input, 'Tipe' (Type) as a dropdown menu, 'Deskripsi' (Description) as a text input, 'Status' as a dropdown menu, and 'Ayam' (Chicken) as a dropdown menu. At the bottom right of the form, there are two buttons: 'Simpan' (Save) and 'Batal' (Cancel). The form is presented in a window-like interface with a title bar and window controls.

Gambar 3.6 Mockup form tambah data barang

Mockup diatas adalah halaman tambah data barang, berfungsi sebagai antarmuka yang memungkinkan pengguna menambahkan informasi barang baru dengan mudah. Formulir tersebut didesain dengan kolom-kolom pengisian yang terstruktur, mencakup detail seperti nama barang, tipe, deskripsi, status, dan ayam atau bukan. Tombol simpan ditempatkan untuk mempermudah pengguna menyimpan data yang dimasukkan, sementara tombol batal memberikan opsi untuk membatalkan proses tanpa menyimpan perubahan.

5. Mockup Halaman Edit barang

EDIT BARANG

Formulir Data

Nama Barang

Tipe

Deskripsi

Status

Ayam

Simpan Batal

Gambar 3.7 Mockup form update data barang

Mockup halaman edit barang tampilannya hampir sama seperti halaman tambah barang, untuk mengedit informasi barang yang sudah ada dalam sistem. Formulir edit dirancang dengan kolom-kolom pengisian yang mencakup berbagai detail seperti nama barang, tipe, deskripsi status, dan ayam atau bukan. Selain itu dilengkapi juga dengan tombol simpan, yang memungkinkan pengguna menyimpan perubahan yang telah dilakukan, serta tombol batal untuk membatalkan proses tanpa menyimpan.

6. Mockup Halaman Index Stok Cabang

STOK CABANG

Daftar Stok Cabang

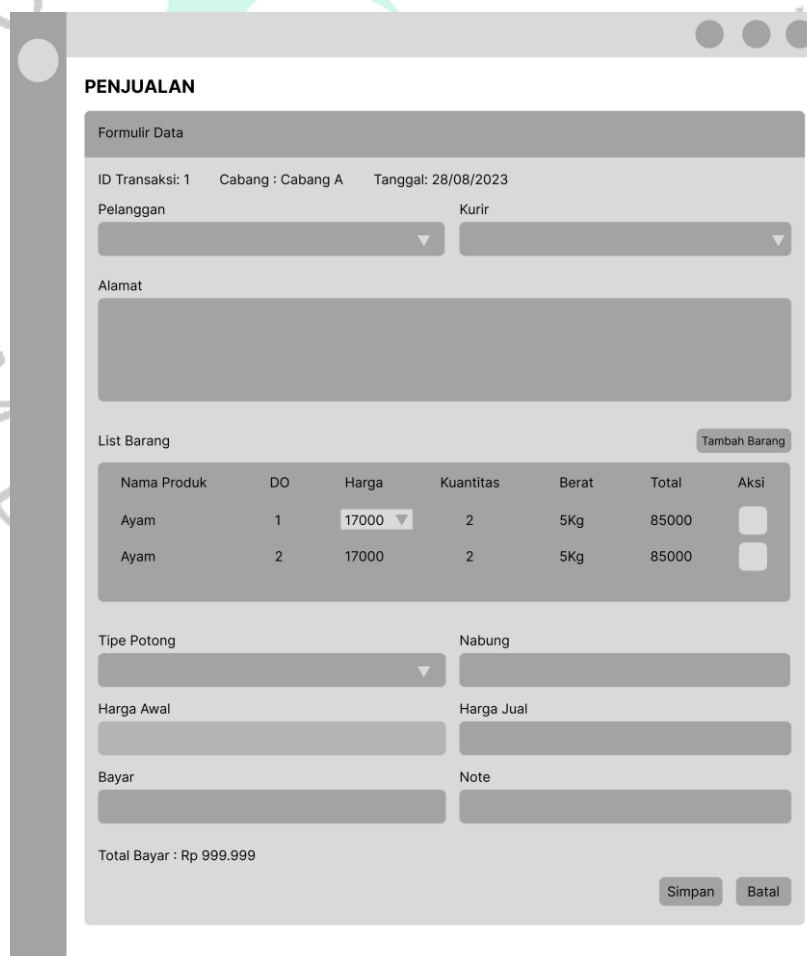
Tambah

NO	Nama	Stok	Jumlah	STATUS	AKSI
1	Indomie Kari	100	90	Tersedia	<input type="button" value="edit"/> <input type="button" value="delete"/>
2	Indomie Ayam Bawang	100	90	Tersedia	<input type="button" value="edit"/> <input type="button" value="delete"/>
3	Ayam	100	1000	Tersedia	<input type="button" value="Detail"/>

Gambar 3.8 Mockup index stok cabang

Mockup diatas untuk halaman index stok cabang memberikan informasi stok pada berbagai cabang dalam sistem. Tampilannya disusun dalam bentuk tabel yang dapat diurutkan (sorting), memudahkan pengguna untuk mengakses data dengan cepat. Setiap baris tabel mencakup informasi seperti nama barang, jumlah barang, rincian stok, dan status. Selain itu, halaman ini dilengkapi dengan tombol tambah data, memungkinkan pengguna untuk menambahkan informasi stok baru pada cabang tertentu. Setiap entri dalam tabel juga menyediakan opsi aksi, seperti tombol edit dan delete, memfasilitasi pengguna dalam mengelola dan memperbarui informasi stok cabang. Terdapat pula kolom pencarian yang mempermudah pengguna menemukan data stok cabang tertentu.

7. Mockup Halaman Penjualan



The image shows a mockup of a sales form titled "PENJUALAN". It includes a "Formulir Data" section with fields for "ID Transaksi: 1", "Cabang : Cabang A", and "Tanggal: 28/08/2023". There are dropdown menus for "Pelanggan" and "Kurir", and a text area for "Alamat". Below this is a "List Barang" table with columns for "Nama Produk", "DO", "Harga", "Kuantitas", "Berat", "Total", and "Aksi". The table contains two rows of "Ayam" with a price of 17000 and a total of 85000. At the bottom, there are fields for "Tipe Potong", "Nabung", "Harga Awal", "Harga Jual", "Bayar", and "Note". A "Total Bayar : Rp 999.999" is displayed, along with "Simpan" and "Batal" buttons.

Nama Produk	DO	Harga	Kuantitas	Berat	Total	Aksi
Ayam	1	17000	2	5Kg	85000	<input type="checkbox"/>
Ayam	2	17000	2	5Kg	85000	<input type="checkbox"/>

Gambar 3.9 Mockup form penjualan cabang

8. Mockup Pop-up tambah barang belanja

PENJUALAN

Formulir Data

ID Transaksi: 1 Cabang : Cabang A Tanggal: 28/08/2023

Pelanggan: Kurir:

Note:

List Barang

Nama Produk	DO	Total	Aksi
Ayam	1	85000	<input type="checkbox"/>
Ayam	2	85000	<input type="checkbox"/>

Tipe Potong: Nabung:

Harga Awal: Harga Jual:

Bayar: Total Bayar : Rp 999.999

List Barang

Search:

Table Header	
Ayam DO1	Stok : 10
Ayam DO2	Stok : 20
Produk 3	Stok : 3
Produk 4	Stok : 30

Gambar 3.10 Mockup form penjualan cabang (cari list barang)

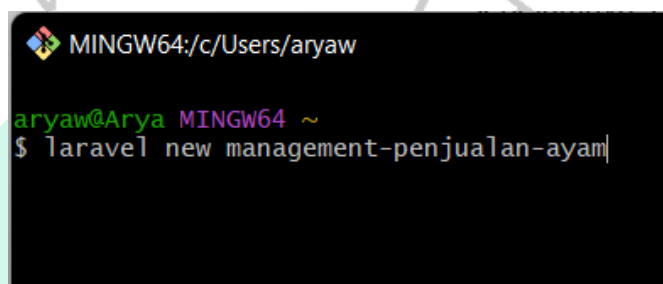
Mockup halaman input penjualan berisi formulir yang memfasilitasi pengguna dalam mencatat informasi penjualan dengan detail. Dengan kolom-kolom seperti data pelanggan, kurir, dan alamat pengiriman, serta kemampuan untuk memilih dan menambahkan barang ke dalam tabel list barang seperti pada gambar 3.10 di atas, pengguna dapat dengan mudah mengelola setiap aspek penjualan. Terdapat dropdown untuk memilih tipe penjualan, seperti potongan harga atau program nabung, sementara kolom harga awal dan harga jual memberikan ruang untuk informasi harga barang. Fitur tambahan berupa kolom bayar dan notes memungkinkan pengguna untuk mencatat pembayaran dan memberikan catatan khusus. Serta tombol simpan transaksi dan batal.

Setelah menyelesaikan rancangan mockup, praktikan dan tim memulai tahap koding dengan men desain mockup yang telah dirancang

sebelumnya. Berikut adalah langkah-langkah yang diambil praktikan dalam memulai pengembangan aplikasi manajemen penjualan dan distribusi ayam di PT Legospay Mitra Sejati yaitu:

1. Inisiasi Proyek Laravel:

Praktikan memulai dengan membuat proyek Laravel baru menggunakan perintah “**laravel new nama-project**”. Langkah ini menghasilkan struktur proyek dasar yang telah terkonfigurasi dengan baik oleh Laravel, termasuk file-file inti seperti model, controller, dan rute.

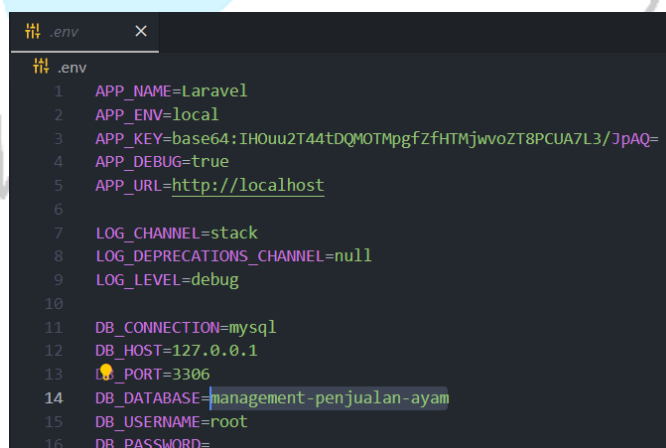


```
MINGW64/c/Users/aryaw
aryaw@Arya MINGW64 ~
$ laravel new management-penjualan-ayam|
```

Gambar 3.11 Inisiasi Proyek Laravel dengan `Laravel new (nama-project)`

2. Konfigurasi `.env`:

Setelah inisiasi proyek, praktikan melakukan konfigurasi file `.env`. Praktikan menyediakan informasi mengenai koneksi database dan konfigurasi server lainnya. Langkah ini memastikan bahwa proyek Laravel terhubung dengan database yang akan dibuat.



```
.env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:IH0uu2T44tDQMOTMpgfZfHTMjwvoZT8PCUA7L3/JpAQ=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=management-penjualan-ayam
15 DB_USERNAME=root
16 DB_PASSWORD=
```

Gambar 3.12 Konfigurasi `.env`

3. Integrasi dengan AdminLTE:

Praktikan melanjutkan dengan tahap integrasi, yaitu memasang template AdminLTE. Template ini digunakan untuk mempercantik antarmuka aplikasi dan memberikan tampilan yang responsive karena berbasis Bootstrap 5. Dengan menambahkan AdminLTE sebagai template, praktikan dapat fokus pada pengembangan fitur dan logika aplikasi tanpa perlu merancang antarmuka dari awal.

Setelah Inisiasi proyek dan konfigurasi proyek yang akan dikembangkan, selanjutnya tim pengembangan menghadapi tahap krusial dalam proses pengembangan aplikasi, yaitu pengembangan database. Tahap ini menjadi fondasi penting untuk menyimpan dan mengelola data dengan efisien sesuai dengan kebutuhan aplikasi. Praktikan, sebagai salah satu fullstack web programmer, turut terlibat dalam merancang dan mengimplementasikan struktur database yang mendukung fungsionalitas aplikasi manajemen penjualan dan distribusi ayam.

Proses pengembangan database dimulai dengan menganalisis kebutuhan data dari seluruh aspek aplikasi, termasuk master data, transaksi penjualan, stok barang, dan informasi pelanggan. Berdasarkan analisis ini, praktikan bekerja sama dengan tim untuk merancang skema database yang sesuai, mencakup tabel-tabel, hubungan antar tabel, serta definisi atribut dan tipe data.

Dalam pengembangan database untuk aplikasi manajemen penjualan dan distribusi ayam untuk Ayam Sri Lestari, praktikan mengadopsi pendekatan yang modern dan efisien dengan tidak menggunakan query langsung atau antarmuka tradisional seperti query SQL mentah ataupun phpMyAdmin. Sebaliknya, praktikan memanfaatkan fitur migration yang disediakan oleh Laravel.

Pendekatan ini didasarkan pada penggunaan fitur migration, sebuah fitur yang memungkinkan pengembang mendefinisikan struktur database menggunakan kode pemrograman PHP daripada query SQL langsung. Dengan fitur migration, praktikan dapat dengan mudah membuat, mengubah, dan mengelola struktur database melalui skrip kode yang terorganisir secara baik.

Langkah pertama dalam penggunaan fitur migration adalah membuat skrip migrasi yang merepresentasikan struktur tabel yang diinginkan. Praktikan dapat mendefinisikan kolom-kolom, indeks, dan relasi antar tabel menggunakan sintaks khusus yang disediakan oleh Laravel. Setelah skrip migrasi dibuat, Laravel memungkinkan praktikan untuk menjalankan migrasi tersebut dengan mudah melalui perintah terminal, tanpa perlu menggunakan antarmuka grafis atau menulis query SQL manual. Berikut contoh pembuatan table menggunakan Migrations pada table produk (*goods*) :

```
public function up()
{
    Schema::create('goods', function (Blueprint $table) {
        $table->increments("gds_id");
        $table->string("gds_name", 30);
        $table->string("gds_desc");
        $table->unsignedBigInteger("gds_type_id");
        $table->foreign("gds_type_id")->references("gds_type_id")->on("good_type");
        $table->string("gds_status_ord", 20);
        $table->char("isayam", 1);
        $table->timestamps();
        $table->softDeletes();
    });
}
```

Gambar 3.13 Contoh penggunaan Migration pada Tabel Goods

Setelah menyiapkan semua struktur table yang sudah dirancang melalui file migrasi dan mengonfigurasi koneksi database di file `.env`, langkah krusial selanjutnya adalah mengaplikasikan perubahan tersebut ke dalam database lokal. Untuk melakukannya, praktikan menjalankan perintah khusus Laravel dengan menggunakan syntax:

```
aryaw@Arya MINGW64 /c/laragon/www/management-penjualan-ayam
$ php artisan migrate
```

Gambar 3.14 Perintah Migrate ke Database Local Server

Perintah ini memicu Laravel untuk mengeksekusi semua skrip migrasi yang sebelumnya telah didefinisikan. Dengan demikian, tabel-tabel dan struktur database yang diperlukan oleh aplikasi dapat dibuat dan

disesuaikan dengan kebutuhan desain. Penting untuk mencatat bahwa eksekusi perintah migrasi harus dilakukan dengan hati-hati, terutama saat bekerja di lingkungan produksi. Oleh karena itu, sebelum menjalankan migrasi, praktikan harus memastikan bahwa data telah dicadangkan untuk mencegah kehilangan informasi yang krusial. Langkah ini merupakan tahap kritis dalam proses pengembangan aplikasi, memastikan bahwa fondasi database telah siap untuk mendukung fungsionalitas yang akan diimplementasikan pada tahap selanjutnya.

Lalu keuntungan utama dari pendekatan menggunakan migration ini adalah kemudahan dalam pengembangan dan pemeliharaan struktur table & database. Praktikan dapat mengelola versi database dengan lebih terstruktur, sehingga memudahkan dalam melacak perubahan dan berkolaborasi dengan tim pengembangan. Selain itu, fitur rollback yang disediakan oleh migration memungkinkan praktikan untuk dengan cepat mengembalikan perubahan jika diperlukan.

Dengan menggunakan fitur migration dalam Laravel, praktikan memastikan bahwa pembuatan database menjadi lebih efisien, terdokumentasi dengan baik, dan mudah dikelola tanpa harus mengandalkan query SQL atau antarmuka pengelolaan database tradisional. Pendekatan ini mencerminkan komitmen praktikan untuk menggunakan teknologi yang memudahkan dan meningkatkan produktivitas dalam pengembangan aplikasi.

Setelah berhasil menjalankan migrasi dan menyiapkan struktur database, praktikan kemudian melanjutkan ke tahap coding fitur-fitur aplikasi dengan menerapkan konsep Model-View-Controller (MVC). Dalam konteks pengembangan aplikasi manajemen penjualan dan distribusi ayam, praktikan menerapkan fitur CRUD (Create, Read, Update, Delete) untuk data master produk sebagai contoh. Berikut adalah langkah-langkah implementasi fitur-fitur tersebut:

1. **MVC**

Praktikan menggunakan konsep MVC untuk memisahkan tanggung jawab masing-masing komponen. Model digunakan untuk berinteraksi dengan

database, View bertanggung jawab untuk menampilkan data ke pengguna, dan Controller untuk mengatur logika aplikasi.

2. Routing

Dengan Laravel, praktikan mendefinisikan rute (routes) yang mengarah ke fungsi-fungsi pada controller didalam file web.php pada folder Route. Sebagai contoh, rute untuk menampilkan data master Produk, menambah data baru, mengedit, dan menghapus data dapat didefinisikan seperti berikut:

```
// WEB ROUTER PRODUK
Route::get('/master-data/produk/i-produk', [GoodController::class, 'index']);
Route::get('/master-data/produk/add-produk', [GoodController::class, 'create']);
Route::post('/master-data/produk/store', [GoodController::class, 'store']);
Route::get('/master-data/produk/edit-produk/{id}', [GoodController::class, 'edit']);
Route::post('/master-data/produk/update-produk', [GoodController::class, 'update']);
Route::get('/master-data/produk/delete-produk/{id}', [GoodController::class, 'destroy']);
```

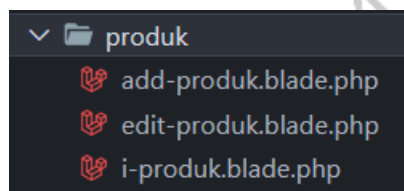
Gambar 3.15 Route Master data Produk

3. Controller

Praktikan membuat controller **GoodController** dengan fungsi-fungsi yang sesuai dengan operasi CRUD. Dalam controller ini, model dan view akan dipanggil untuk melakukan operasi yang diinginkan.

4. View

Praktikan membuat tiga view: **i-produk.blade.php** untuk menampilkan data produk, **add-produk.blade.php** untuk menambah data baru, dan **edit-produk.blade.php** untuk mengedit data produk. Setiap view dilengkapi dengan tombol aksi (action) untuk mengarahkan pengguna ke fungsi-fungsi pada controller. Berikut contoh hirarki file tampilan data produk:

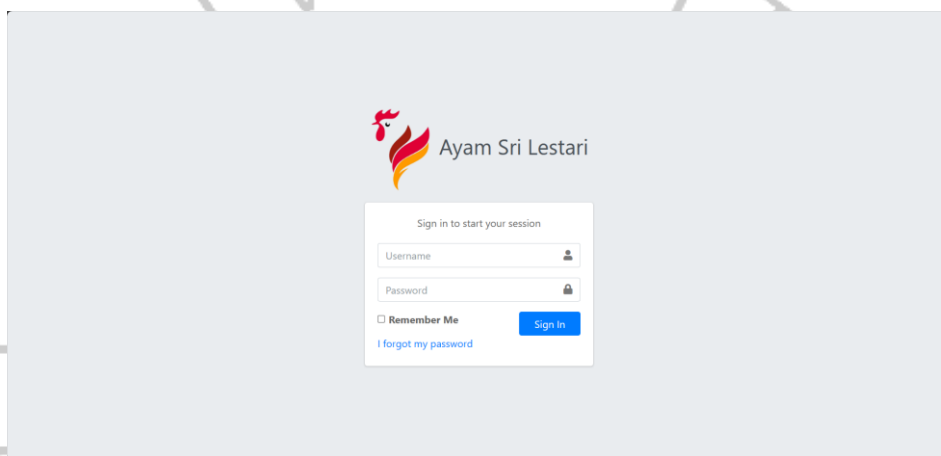


Gambar 3.16 Hirarki folder Master data Produk

Selama periode kerja profesi, praktikan telah berhasil mengembangkan sejumlah tampilan halaman yang menjadi bagian dari proyek yang sedang dikerjakan. Tampilan-tampilan tersebut mencakup berbagai aspek dan fitur yang

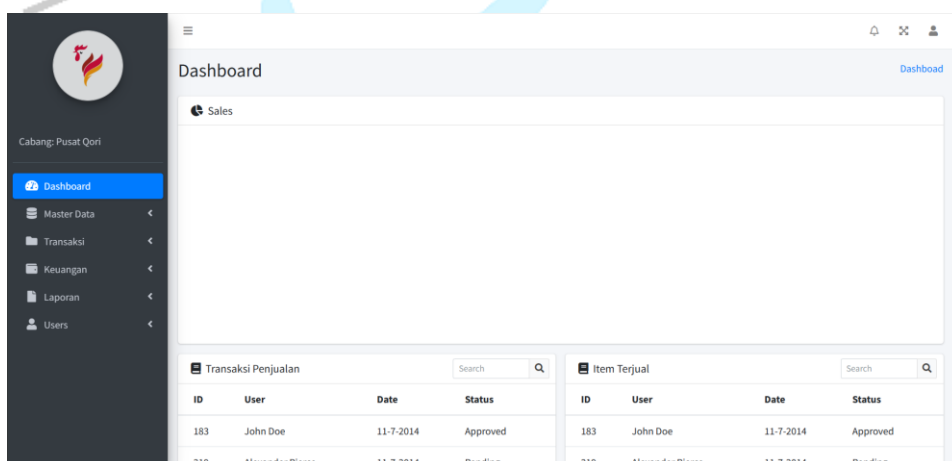
diintegrasikan ke dalam aplikasi yang tengah dikembangkan. Dengan merinci setiap elemen halaman, praktikan dapat menunjukkan kemampuan dalam membuat antarmuka yang intuitif dan fungsional. Proses ini menjadi landasan penting dalam menghadirkan solusi yang memadai untuk manajemen penjualan dan distribusi ayam, menciptakan tampilan yang tidak hanya estetik tetapi juga mendukung operasional yang efisien berdasarkan mockup yang sebelumnya telah dirancang. Berikut adalah beberapa tampilan aplikasi web yang praktikan kembangkan :

1. Tampilan Halaman Login



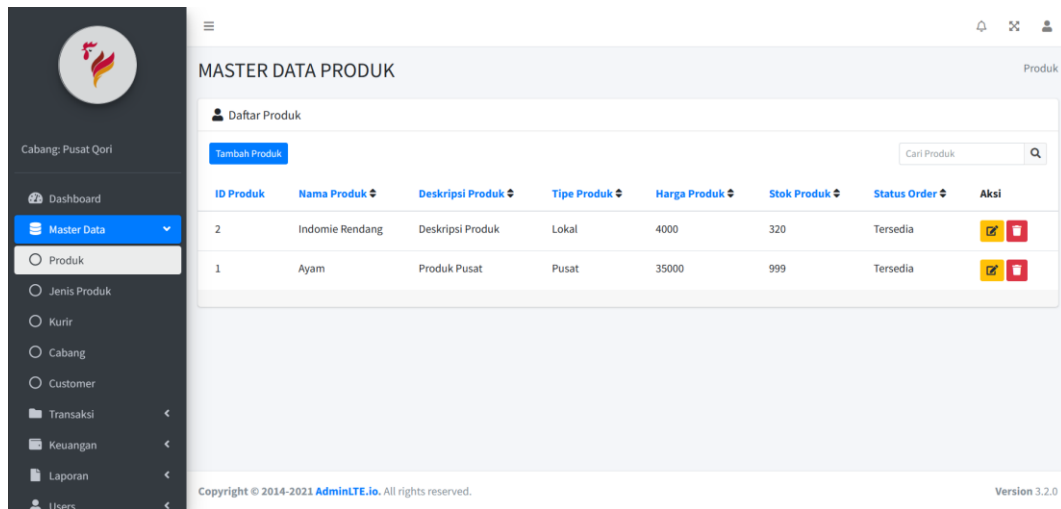
Gambar 3.17 Tampilan halaman Login

2. Tampilan Halaman Dashboard



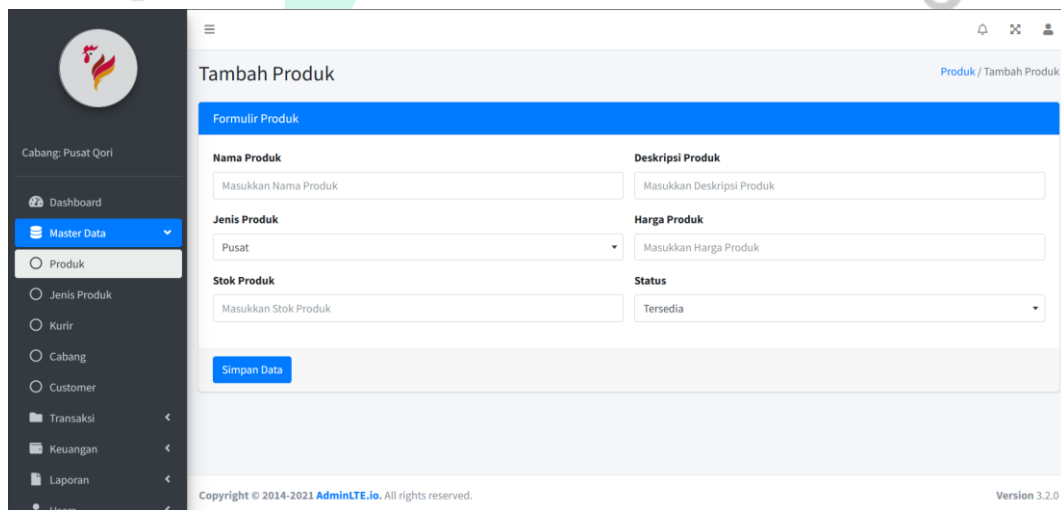
Gambar 3.18 Tampilan halaman Dashboard

3. Tampilan Halaman Index Produk



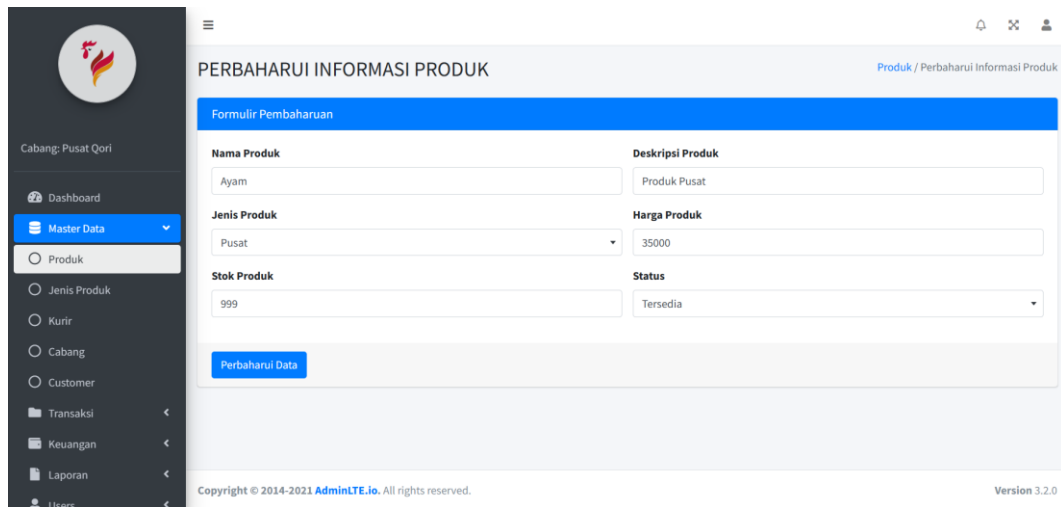
Gambar 3.19 Tampilan halaman Index Produk (Pusat)

4. Tampilan Halaman Tambah Produk



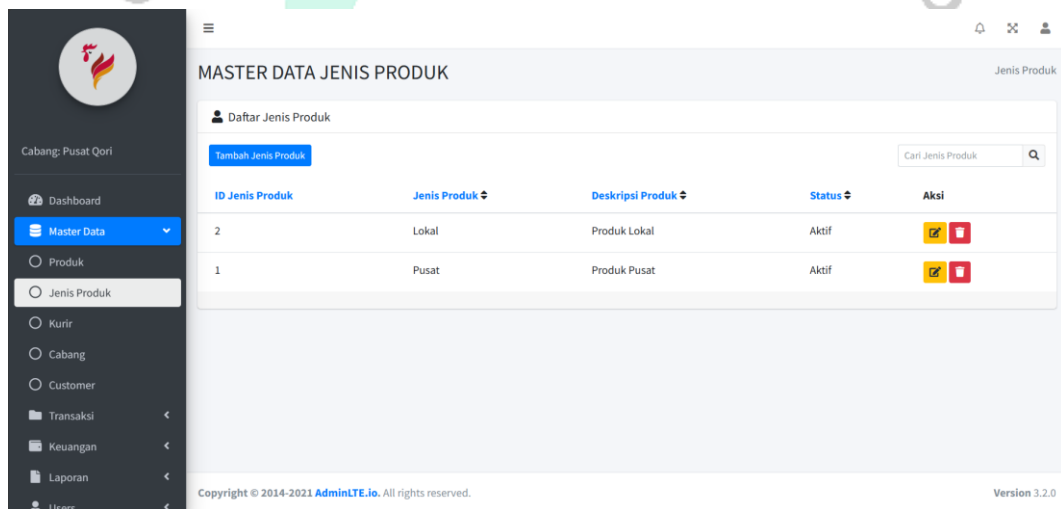
Gambar 3.20 Tampilan halaman tambah Produk (Pusat)

5. Tampilan Halaman Update Produk



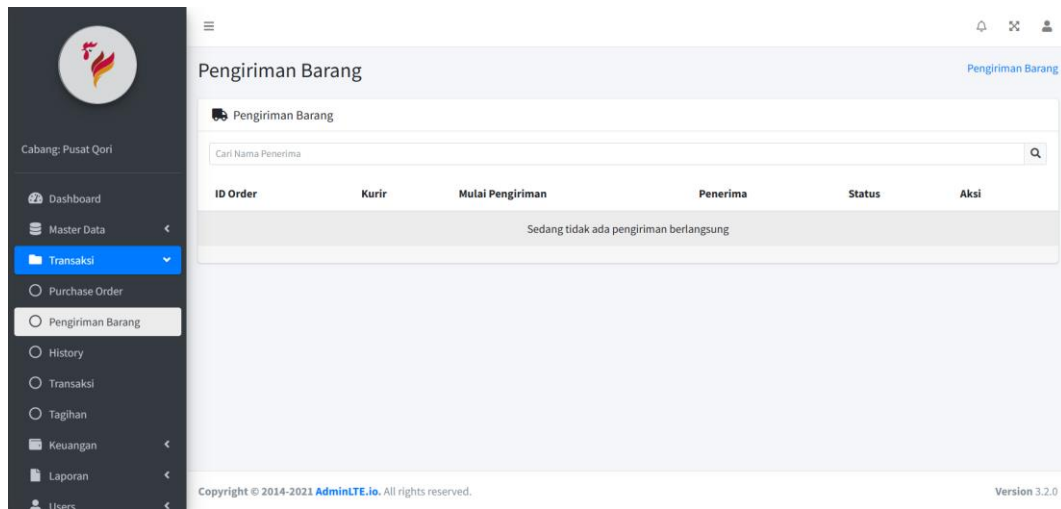
Gambar 3.21 Tampilan halaman update Produk (Pusat)

6. Tampilan Halaman Index Data Jenis Produk



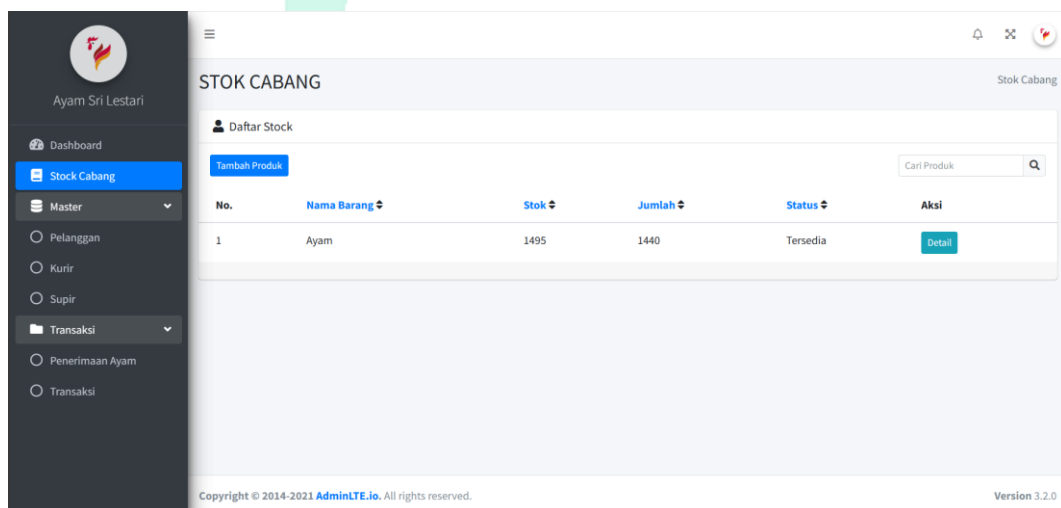
Gambar 3.22 Tampilan halaman index Jenis Produk (Pusat)

7. Tampilan Halaman Pengiriman Barang



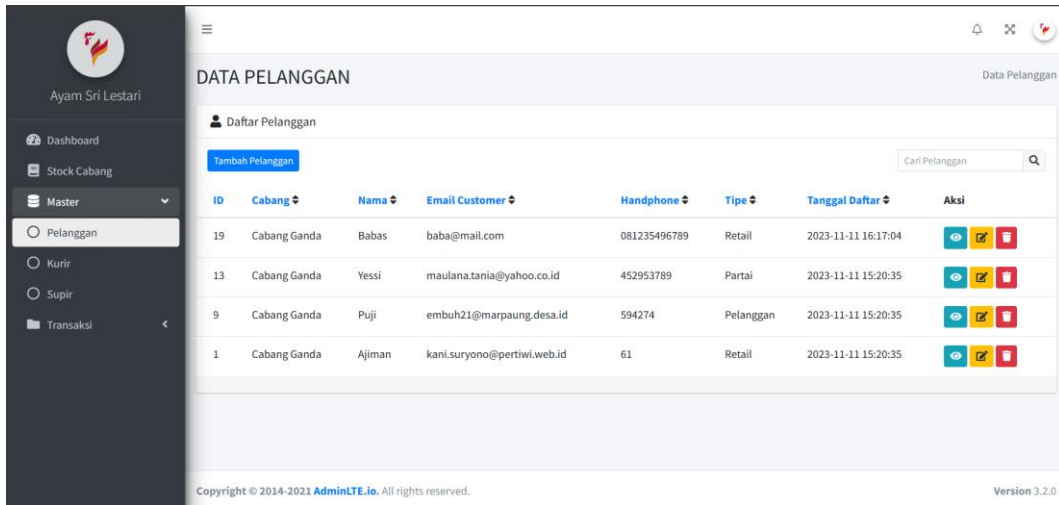
Gambar 3.23 Tampilan halaman pengiriman barang (Pusat)

8. Tampilan Halaman Stok Cabang



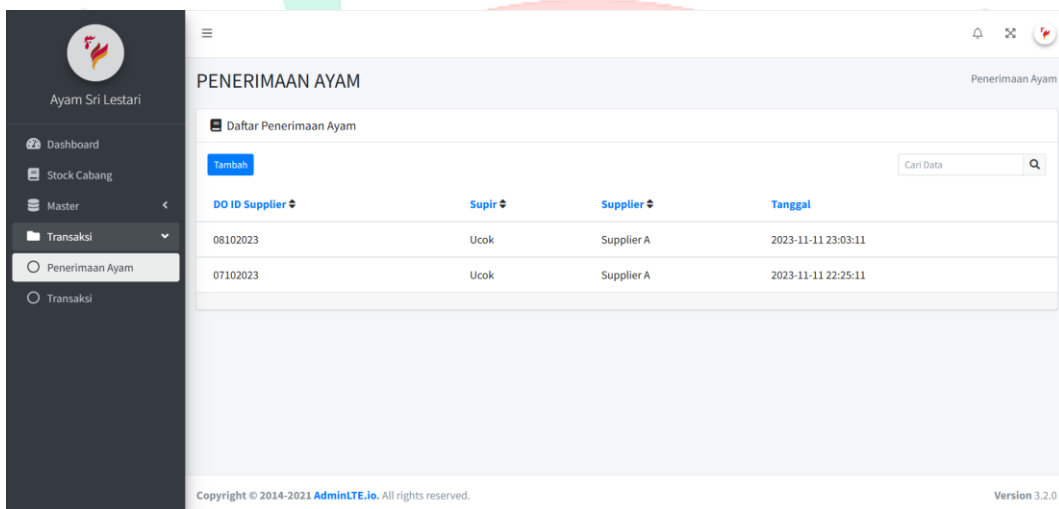
Gambar 3.24 Tampilan halaman stok cabang (Cabang)

9. Tampilan Halaman Index Pelanggan



Gambar 3.25 Tampilan halaman index pelanggan (Cabang)

10. Tampilan Halaman Penerimaan Ayam



Gambar 3.26 Tampilan halaman penerimaan ayam (Cabang)

11. Tampilan Halaman Penjualan

The screenshot shows a web application interface for 'Ayam Sri Lestari'. The main content area is titled 'PENJUALAN' (Sales) and contains a 'Formulir Data' (Data Form). The form includes the following fields and sections:

- ID Order:** 6, **Tanggal Order:** 2023-11-12 08:57:36
- Pelanggan:** Ajiman (dropdown menu)
- Kurir Pengantar:** Dodo (dropdown menu)
- Alamat:** Masukan alamat pengiriman order (text input)
- List Barang:** A table with columns: Nama Produk, Harga, Kuantitas, Bobot (Kg), Total, Aksi. The table contains one row: 'Fillet - Rp.5000' with 'Rp 0,00' in the Harga column and '0' in the Bobot (Kg) column. A 'Tambah Barang' button is located to the right of the table.
- Tipe Potong:** Fillet - Rp.5000 (dropdown menu)
- Nabung:** Masukan Nominal (text input)
- Bayar:** Masukan nominal (text input)
- Note:** Masukan note (text input)

At the bottom right of the form, there are two buttons: 'Buat Pesanan' (Create Order) and 'Batal' (Cancel). The footer of the page includes 'Copyright © 2014-2021 AdminLTE.io. All rights reserved.' and 'Version 3.2.0'.

Gambar 3.27 Tampilan halaman penjualan (Cabang)

3.3 Kendala Yang Dihadapi

Dalam proses pengembangan aplikasi web penjualan ayam, tim kami mengalami beberapa kendala yang memengaruhi jalannya proyek. Salah satu kendala yang paling signifikan adalah perubahan permintaan dari klien setelah aplikasi hampir selesai sesuai dengan requirement awal. Klien menginginkan konsep yang berbeda dan merasa perlu menambahkan beberapa fitur baru ke dalam aplikasi yang sudah ada. Kendala ini dapat dibahas sebagai berikut:

1. Perubahan Konsep Aplikasi

Salah satu kendala utama yang dihadapi tim adalah perubahan konsep aplikasi oleh klien. Setelah aplikasi sesuai dengan requirement awal selesai, klien menginginkan perubahan signifikan dalam konsep dan tampilan aplikasi. Hal ini mengharuskan praktikan dan tim untuk mengubah arah pengembangan aplikasi yang telah ada, yang dapat mempengaruhi rencana dan jadwal proyek yang telah ditetapkan.

2. Penambahan Fitur

Selain perubahan konsep, klien juga menginginkan penambahan beberapa fitur baru ke dalam aplikasi. Ini termasuk fitur-fitur yang tidak terdapat dalam requirement awal, yang memerlukan pengembangan tambahan dan pengujian. Penambahan fitur ini dapat mempengaruhi rencana waktu dan sumber daya yang telah dialokasikan sebelumnya.

3. Kesulitan dalam Koordinasi

Kendala yang muncul akibat perubahan konsep dan penambahan fitur adalah kesulitan dalam koordinasi. Tim harus berkoordinasi dengan baik untuk memahami perubahan yang diminta oleh klien, menyesuaikan rencana pengembangan, dan memastikan bahwa semua anggota tim memahami dan menjalankan tugas sesuai dengan perubahan tersebut.

4. Keterbatasan Sumber Daya

Penambahan fitur dan perubahan konsep bisa mempengaruhi keterbatasan sumber daya yang telah dialokasikan sebelumnya, seperti waktu dan tenaga kerja. Praktikan dan tim harus berusaha untuk memenuhi kebutuhan tambahan ini dengan tetap mempertahankan kualitas pekerjaan dan memastikan bahwa proyek tidak terlambat.

3.4 Cara Mengatasi Kendala

Untuk mengatasi kendala-kendala ini, tim harus memiliki kemampuan untuk beradaptasi dengan perubahan, berkomunikasi dengan klien secara efektif, dan melakukan manajemen proyek yang baik. Selain itu, tim harus menjaga kualitas hasil akhir dan memastikan bahwa perubahan yang diminta oleh klien dapat diintegrasikan ke dalam aplikasi dengan baik. Meskipun kendala ini bisa menantang, klien juga memberikan peluang untuk meningkatkan kemampuan dalam pengelolaan perubahan dan kolaborasi tim yang efektif.

Solusi dari Praktikan dan tim pada kendala yang dihadapi adalah memberhentikan proses pembuatan aplikasi secara sementara ketika klien merasa tidak sesuai dengan proses yang dikerjakan dan memberikan beberapa masukan dan tambahan. Selama pemberhentian pembuatan,

Praktikan dan tim mengevaluasi kembali kebutuhan yang dibutuhkan oleh klien. Praktikan dan tim harus mengevaluasi proses bisnis dan pembuatan agar sesuai dengan kemampuan klien namun tidak secara terus menerus memperbesar scope proyek yang dikerjakan.

Tidak semua perubahan yang diminta oleh klien dapat dilakukan Praktikan dan tim secara cuma-cuma. Perubahan major/besar yang terjadi setelah tahap evaluasi akan dikenakan fee/charge dan akan dikerjakan ketika scope yang disetujui kedua belah pihak sudah selesai. Hal ini bertujuan untuk mencegahnya perubahan ketika tahap pengembangan berlangsung. Praktikan dan tim hanya mempertimbangkan perubahan minor/kecil yang tidak berdampak memperlambat proses pengembangan.

3.5 Pembelajaran Yang Diperoleh dari Kerja Profesi

Pada perjalanan pengembangan "Sistem Penjualan dan Distribusi Ayam" di PT Legospay Mitra Sejati, praktikan menemukan sejumlah pembelajaran berharga yang mencakup beragam aspek, mulai dari kerja tim hingga pengelolaan kebutuhan pengguna. Pengalaman ini tidak hanya meningkatkan keterampilan teknis praktikan dalam pengembangan aplikasi web, tetapi juga membawa wawasan mendalam tentang esensi kolaborasi tim, analisis kebutuhan pengguna, dan adaptabilitas terhadap dinamika proyek yang berubah.

Dalam sub bab ini, akan diuraikan dengan rinci pembelajaran yang diperoleh sepanjang perjalanan kerja profesi ini.

1. Pentingnya Kolaborasi Tim

Seiring perjalanan kerja profesi, satu pembelajaran utama yang praktikan peroleh adalah pentingnya kolaborasi tim. Dalam mengembangkan sistem manajemen penjualan dan distribusi ayam, praktikan bekerja sebagai satu tim dengan lead programmer, fullstack web programmer, dan projek manager telah memberikan wawasan mendalam tentang dinamika kerja kelompok. Pemahaman ini mencakup bagaimana berkomunikasi secara efektif, memecahkan masalah bersama, dan memastikan bahwa kontribusi masing-masing anggota tim dapat terintegrasi secara harmonis dalam proyek.

2. Analisis Kebutuhan Pengguna sebagai Landasan Pengembangan

Proses pengembangan aplikasi dimulai dengan tahap analisis kebutuhan pengguna. Pembelajaran yang diperoleh adalah bahwa analisis yang mendalam tentang kebutuhan pengguna membentuk landasan yang kokoh untuk seluruh pengembangan proyek. Pemahaman yang mendalam tentang kebutuhan dan ekspektasi pengguna memungkinkan pengembang untuk merancang solusi yang sesuai dan inovatif.

3. Fleksibilitas dan Aksesibilitas Melalui Platform Web

Pengalaman mengembangkan aplikasi web untuk sistem manajemen penjualan dan distribusi ayam menyoroti pentingnya fleksibilitas dan aksesibilitas. Melalui keputusan membuat aplikasi yang dapat diakses melalui web, pembelajaran ini menggarisbawahi bahwa memberikan akses yang mudah dan fleksibilitas kepada pengguna untuk mengelola bisnis dari berbagai lokasi adalah elemen penting dalam perancangan aplikasi.

4. Manajemen Data yang Terstruktur

Dalam konteks sistem penjualan dan distribusi, pentingnya manajemen data yang terstruktur menjadi sebuah pembelajaran. Dengan kemampuan pusat untuk mengelola tipe produk, data produk, data cabang, dan data lain secara terstruktur, aplikasi ini memberikan pengalaman dalam merancang sistem manajemen data yang efisien dan mudah dipelihara.

5. Keterlibatan Pengguna dalam Proses Pengembangan

Proses pengembangan aplikasi ini memberikan pengalaman yang kaya tentang keterlibatan pengguna. Dengan berfokus pada interaksi aktif antara tim pengembang dan pengguna, pembelajaran ini menekankan pentingnya mendengarkan kebutuhan dan umpan balik pengguna selama seluruh siklus pengembangan.

6. Penggunaan Framework dan Teknologi yang Tepat

Pemilihan framework (Laravel 9), bahasa pemrograman (PHP), dan teknologi (MySQL, Bootstrap 5, AdminLTE) menjadi pembelajaran tentang pentingnya memilih alat dan teknologi yang tepat untuk proyek. Pilihan ini dapat membantu dalam kecepatan, keberlanjutan,

kinerja optimal, dan tampilan yang responsif dalam pengembangan aplikasi.

7. Ketanggapan terhadap Perubahan Konsep dari Klien

Dalam menghadapi tantangan setelah selesai sesuai requirement pertama, pembelajaran yang diperoleh adalah tentang ketanggapan terhadap perubahan konsep dari klien. Fleksibilitas dan kemampuan beradaptasi tim pengembang menjadi kunci untuk memenuhi perubahan kebutuhan klien dengan efektif.

Melalui serangkaian pengalaman ini, praktikan tidak hanya memperoleh keterampilan teknis yang lebih dalam pada pengembangan aplikasi web tetapi juga memahami aspek-aspek kritis dalam manajemen proyek, kerja tim, dan adaptabilitas terhadap dinamika proyek yang berubah. Pengalaman kerja profesi ini memberikan pondasi yang kokoh untuk pertumbuhan profesional praktikan di masa depan.