

BAB III

METODE PENELITIAN

3.1 Alat dan Bahan Penelitian

Dalam penelitian dibutuhkan beberapa alat dan bahan untuk mendukung berjalannya perancangan dan implementasi *aplikasi web*.

3.1.1. Alat

Alat yang digunakan berupa perangkat keras (*hardware*) dan perangkat lunak (*software*).

a. Perangkat Keras (*Hardware*)

1. *Personal Computer* (PC) atau laptop.
 - Interl® Core™ i7 *processor*,
 - 7.20 GB RAM,
 - 64 bit *Operating System*,
 - Windows 10 *windows edition*,
 - *Mouse*
2. *Printer* dokumen untuk mencetak laporan.

b. Perangkat Lunak (*Software*)

1. *Microsoft Visio 2013*
2. *Microsoft Visual Studio 2015*
3. *SQL Server Management 2014*
4. *Crystal Report*
5. *Web Browser Google Chrome*

3.1.2. Bahan

Dalam penelitian ini bahan yang digunakan adalah:

- a. Data yang diperoleh melalui studi literatur berdasarkan kepada penelitian langsung kepada user. Dari data yang diperoleh, maka didapatkan kebutuhan dari sistem pada saat pengembangan sistem.
- b. Data setoran, pendapatan, penjualan dalam satu periode Koperasi Pala Obilatu.

3.2. Teknik Pengumpulan Data

3.2.1. Wawancara

Pengambilan data untuk penelitian menggunakan metode wawancara yang di dapat seperti menanyakan lansung kepada warga atau anggota koperasi dengan bertujuan mengetahui kekurangan dan kelebihan dari segi pengenalan yang luas.

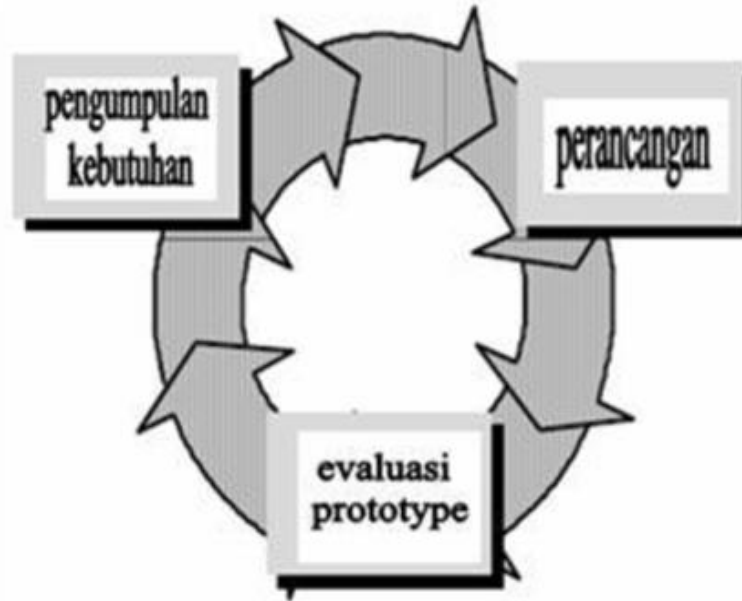
Pada teknik wawancara ini, peneliti menanyakan langsung ke pihak pengurus atau anggota Koperasi Pala Obilatu untuk mendapatkan informasi yang akurat dari segi kekurangan, kelebihan serta kebutuhan.

3.2.2. Observasi

Metode observasi ini dilakukan untuk mengamati secara langsung objek penelitian agar peneliti dapat mengumpulkan data dan menyimpulkan data secara langsung. Pada penelitian ini, peneliti melakukan pengamatan terhadap berkas penyeteran anggota, penjualan anggota, pendapatan setiap pasca panen, dan laporan-laporan lain di Koperasi Pala Obilatu.

3.2.3. Alur Penelitian

Menurut (Fajar Fani Hartono, Hendry, Ramos, Somya, 2013), berjudul “Aplikasi *Reservasi* Tiket Bus pada *Handphone* Android menggunakan *Web Service*”. Perancangan sistem ini menggunakan metode *prototyping*. *Prototyping* adalah proses yang digunakan untuk membantu pengembangan perangkat lunak dalam membentuk model dari perangkat lunak yang harus dibuat. Metode ini dilakukan secara bertahap, yaitu dengan mengembangkan suatu *prototype* yang sederhana terlebih dahulu baru kemudian dikembangkan dari waktu ke waktu sampai perangkat lunak selesai dikembangkan. *Prototype* merupakan bentuk dasar atau model awal dari suatu sistem atau subsistem.



Gambar 3. 1 *Metode Prototyping*

Secara garis besar terdapat tiga tahapan dalam metode *prototyping* ini, seperti terlihat pada Gambar 3.1

1. Pengumpulan Kebutuhan

Tahap pertama yang dilakukan pada metode *prototyping* adalah pengumpulan kebutuhan.

2. Perancangan Sistem

Langkah berikutnya dari metode ini adalah membuat rancangan sistem aplikasi, yaitu membangun sistem berdasarkan hasil dari tahap sebelumnya. Pada tahap ini perancangan sistem dilakukan dengan menggunakan UML (*Unified Model Language*) yang memanfaatkan tools *rational rose*, pada tahap ini pula dilakukan desain terhadap *User Interface*. Kemudian dilakukan perancangan tabel yang dibutuhkan dalam pembuatan aplikasi. Setelah semuanya selesai maka pada tahap ini aplikasi dibuat sampai selesai sesuai dengan *user requirement* yang diperoleh pada tahap awal.

3. Evaluasi *Prototyping*

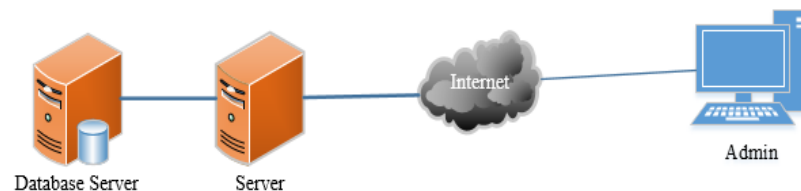
Setelah aplikasi selesai dibuat, maka pada tahap ini dilakukan *testing* aplikasi terhadap *user*. *Testing* aplikasi ini bertujuan untuk mempermudah melakukan evaluasi terhadap kekurangan yang ada di dalam aplikasi sesuai dengan *user requirement* pada tahap awal. Ini merupakan tahap paling akhir yang ada di dalam *prototyping*, apakah tujuan umum dari pembuatan software ini telah tercapai.

Apabila belum tercapai maka tahap selanjutnya yang harus dilakukan adalah kembali ke tahap paling awal, begitu seterusnya sampai tujuan umum dapat tercapai.

3.3. Analisi Penelitian

Analisis penelitian bertujuan untuk mencari kebutuhan fungsional dan kebutuhan *non fungsional*, untuk lebih jelasnya bisa di lihat sebagai berikut:

3.3.1. Arsitektur



Gambar 3. 2 Arsitektur Rill

Untuk *database* yang digunakan pada *aplikasi web* adalah *SQL Server 2014 Management Studio* yang kemudian diolah melawati *IIS server local*. *Aplikasi web* yang digunakan oleh pengguna menggunakan *Web Application* dengan metode MVC (*Model View Controller*) berbahasa *ASP.NET*. Saat admin mengakses *website*, *local server* membuat antarmuka dan memberikan data yang diperlukan dari *request* admin. Melalui antarmuka yang ditampilkan admin dapat mengolah data seperti menyimpan, memperbaharui, menghapus, melihat dan mencari data sesuai dengan yang diinginkan.

3.3.2. Analisis Pengguna

Sistem yang dibuat akan digunakan oleh *admin* sendiri. Seorang *admin* yang sekaligus berperan sebagai administrator memiliki hak akses full atau dapat melakukan apapun dalam *website*, seperti mengolah data penyetoran, data penjualan dan data pendapatan. Disetiap fitur data *administrator* dapat menambah, menghapus, dan mengubah data pengaturan data *master* yang ada di dalam *sistem*. Sedangkan *user* lain masing-masing tidak memiliki kemampuan

seperti seorang admin, user lain seperti anggota hanya mendapat laporan hasil setoran dan pendapatn dalam setiap periode.

3.3.3. Analisis Kebutuhan

Analisis kebutuhan didapat dari studi literatur dan diskusi bersama pendiri anggota Koperasi Pala Obilatu. Berikut ini merupakan analisis kebutuhan pada *aplikasi web* :

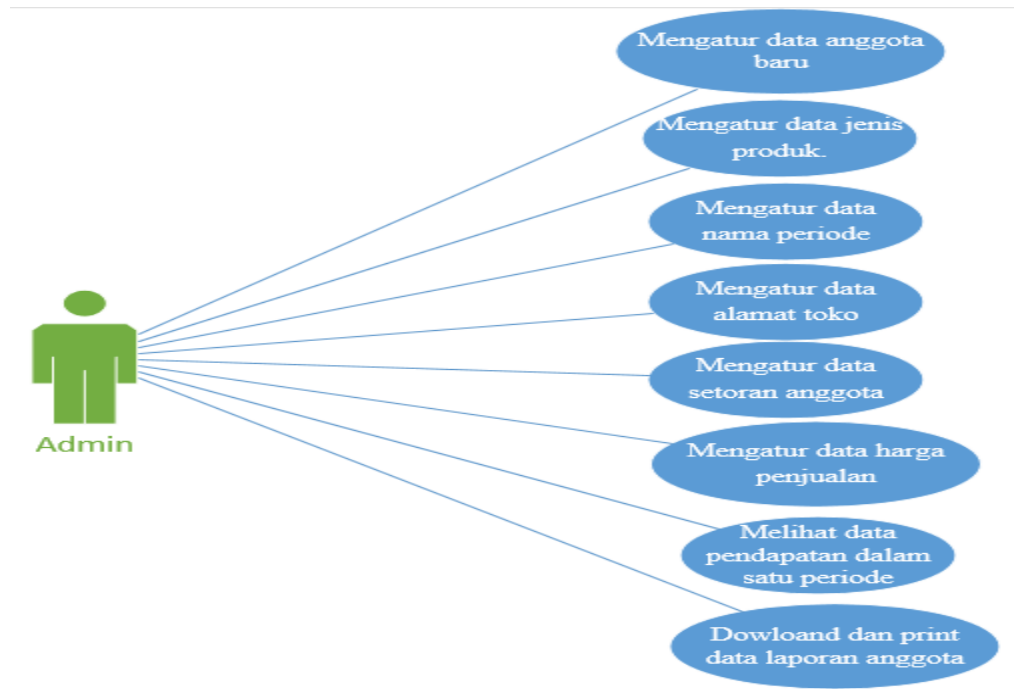
1. Terdapat *login admin*, *manage admin*, dan *logout* hanya untuk *admin* sendiri dan berfungsi *login* agar bisa masuk dalam aplikasi, *manage admin* berfungsi untuk menambahkan atau menghapus data *admin* lain yang sudah tidak dibutuhkan lagi, sedangkan *logout* berfungsi untuk keluar dari aplikasi.
2. Terdapat anggota, periode, jenis produk dan toko, beserta atribut - atributnya yang dikelola di menu *master*.
3. Terdapat halaman setoran, berfungsi untuk memasuka data setoran anggota, seperti dengan memili id anggota, tanggal setora, periode, jenis produk setelah itu kita mengisi jumlah setoran anggota.
4. Terdapat halaman penjualan. Halaman ini kita akan mengisi tanggal penjualan, periode penjualan, memili toko penjualan, jenis produk, serta harga per kilo.
5. Terdapat halaman pendapatan, halaman ini berisi hasil dari rekapan setoran anggota dan harga penjualan anggota dalam setiap periode.
6. Tedapat halaman laporan anggota, halaman ini berisi tentang jumlah setoran anggota, potongan serta pendapatan bersih dalam setiap periode.

3.4. Rancangan Sistem Basis Data

Sebelum membangun *aplikasi web* Koperasi Pala Obilatu, terlebih dahulu dituangkan dalam rancangan diagram - diagram. Diagram ini berfungsi sebagai patokan jalannya *aplikasi web*, kebutuhan, dan gambaran *database*. Metode yang digunakan dalam perancangan *aplikasi web* adalah *United Markup Language* (UML). Metode UML yang dipakai dalam pengembangan *aplikasi web* yaitu model diagram *Use Case Diagram*, *ER Diagram*, *Class Diagram*, dan *Activity Diagram*.

3.4.1. Use Case Diagram

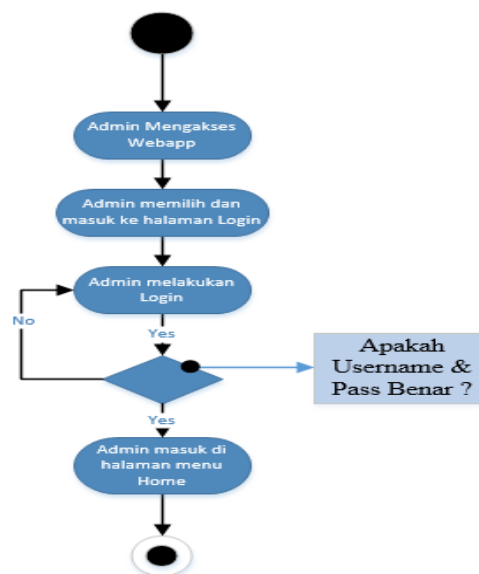
Gambaran *Use Case Diagram* yang digunakan dalam aplikasi dapat dilihat pada gambar 3.3



Gambar 3. 3 Use Case Diagram

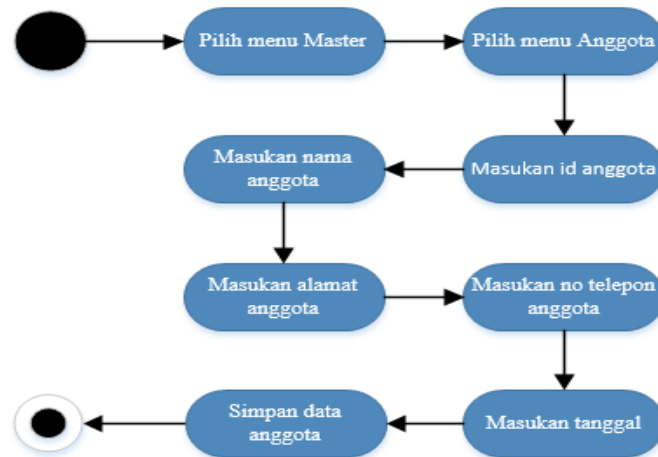
Gambar diatas menunjukkan bagaimana hubungan antar *actor* dan *use case*. Pada gambar diatas menjelaskan bahwa pada *aplikasi web*, dalam kegiatannya *admin* lah yang memiliki akses penuh untuk mengolah data anggota, data periode, data jenis produk, data toko, data setoran, data penjualan, data pendapatan, dan data laporan anggota. Sebelumnya admin harus *login* terlebih dahulu agar dapat melakukan CRUD (*Create, Update, dan Delete*) serta dapat membuat rekap laporan setiap periode setoran dan pendapatan anggota.

3.4.2. Activity Diagram



Gambar 3. 4 Activity Diagram Login

Pada Gambar 3.3 diatas menjelaskan seorang *admin* sebelum mengakses *webAPP*, bagaimana seorang *admin* saat melakukan login untuk dapat pertama kali masuk kedalam halaman utama *website* aplikasi. *Admin* diharapkan untuk memasukkan nama dan *password* yang sudah terdaftar jika nama dan *password* benar maka akan masuk kedalam halaman utama dan apabila salah satu dari nama atau *password* salah maka *admin* dikembalikan pada menu *login*, agar memasukkan nama dan *password* yang benar.



Gambar 0.5 Activity Diagram Anggota

Pada *activity* diagram anggota diatas seorang *admin* saat melakukan penambahan data anggota baru diharapkan mengisi semua data yang sudah disesuaikan. Jika semua data sudah terisi maka *admin* dapat menyimpan data dan pada tampilan semua data anggota *admin* dapat melakukan *edit*, *delete* dan lihat detail data anggota.



Gambar 3. 6 Activity Diagram Jenis Produk

Pada *activity* diagram jenis produk diatas seorang *admin* saat melakukan penambahan data jenis produk baru diharapkan mengisi semua data yang sudah disesuaikan. Jika semua data sudah terisi maka *admin* dapat menyimpan data dan pada tampilan semua data jenis produk *admin* dapat melakukan *edit*, *delete* dan lihat detail data jenis produk.



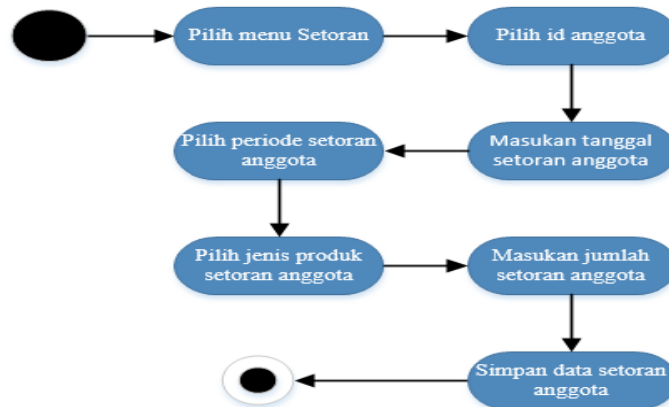
Gambar 3. 7 Activity Diagram Periode

Pada *activity* diagram periode diatas seorang admin saat melakukan penambahan data periode baru diharapkan mengisi semua data yang sudah disesuaikan. Jika semua data sudah terisi maka admin dapat menyimpan data dan pada tampilan semua data periode admin dapat melakukan *edit*, *delete* dan lihat detail data periode.



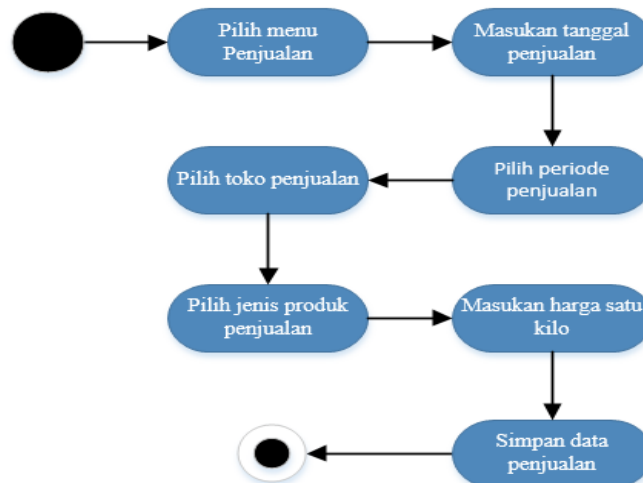
Gambar 3. 8 Activity Diagram Toko

Pada *activity* diagram toko diatas seorang *admin* saat melakukan penambahan data toko baru diharapkan mengisi semua data yang sudah disesuaikan. Jika semua data sudah terisi maka *admin* dapat menyimpan data dan pada tampilan semua data toko *admin* dapat melakukan *edit*, *delete* dan lihat detail data toko.



Gambar 3. 9 Activity Diagram Setoran

Gambar diatas menunjukkan *Activity Diagram* untuk kegiatan setoran anggota. Setelah anggota memberi setoran, maka *admin* akan mengisi data setoran anggota yang telah disesuaikan. Jika semua data sudah terisi maka *admin* dapat menyimpan data dan pada tampilan semua data setoran *admin* dapat melakukan *edit*, *delete* dan lihat detail data setoran anggota.



Gambar 3. 10 Activity Diagram Penjualan

Gambar diatas menunjukkan *Activity Diagram* untuk kegiatan penjualan. Seorang *admin* saat melakukan penambahan data penjualan baru diharapkan mengisi semua data yang sudah

disesuaikan. Jika semua data sudah terisi maka *admin* dapat menyimpan data dan pada tampilan semua data penjualan *admin* dapat melakukan *edit*, *delete* dan lihat detail data penjualan.



Gambar 3. 11 Activity Diagram Pendapatan

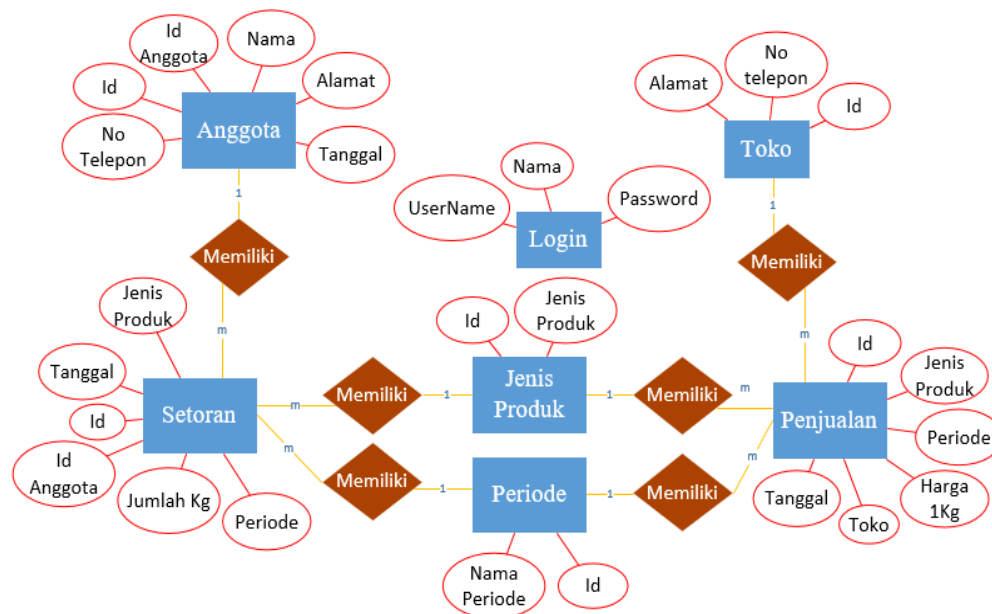
Gambar diatas menunjukkan *Activity Diagram* untuk kegiatan pendapatn. Seorang *admin* saat melakukan pencarian berdasarkan periode, maka akan muncul seluruh pendapatan serta potongan dalam setiap periode yang dicari seperti id produk, total jual, total potongan, total pendapatan, jumlah kilo terjual dan pendapatan satu kilo gram.



Gambar 3. 12 Activity Diagram Laporan Anggota

Gambar diatas menunjukkan *Activity Diagram* untuk kegiatan laporan anggota. Seorang *admin* saat melakukan pencarian berdasarkan periode dan id anggota, maka akan muncul seluruh data laporan anggota berdasarkan periode yang dicari, seperti data laporan id anggota, tanggal setor, nama anggota, nama toko, periode, jenis produk, jumlah kg setoran, harga penjualan satu kilo, pendapatan, potongan dan pendapatan bersih.

3.4.3. ER Diagram



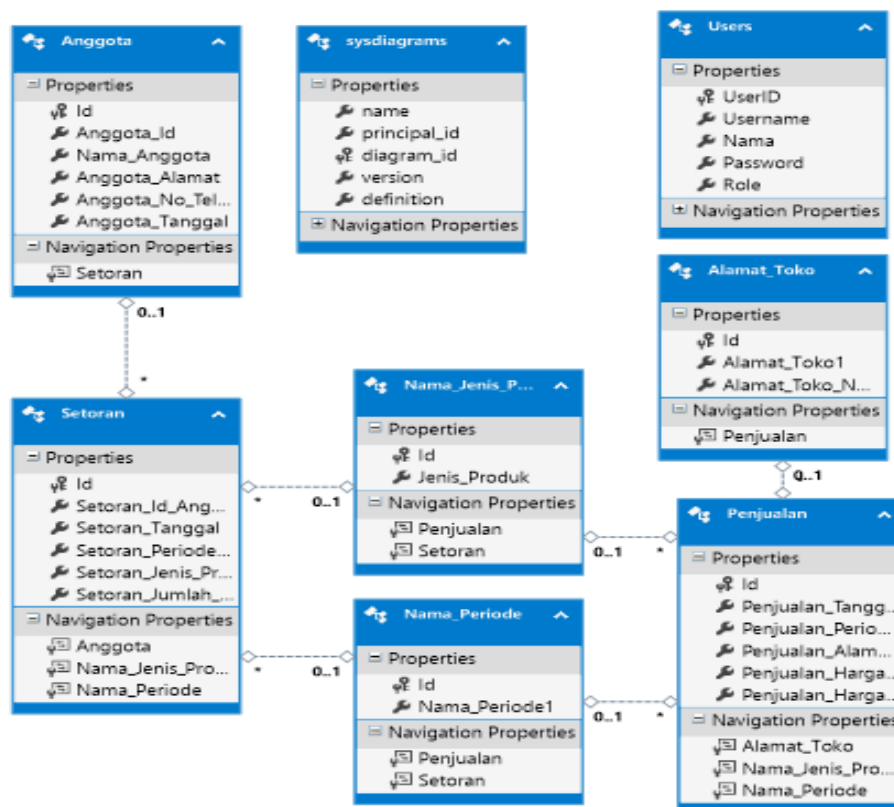
Gambar 3. 13 ER Diagram

Gambar 3.13 diatas adalah *ER Diagram* SISTEM INFORMASI KOPERASI PALA OBILATU yang menunjukkan entitas dengan atribut dan relasinya. Entitas ini yang nantinya akan menjadi rancangan tabel pada *database* program. *Entitas* diatas memiliki relasi dari satu dengan yang lain. Beberapa relasi akan dijelaskan sebagai berikut:

1. Entitas *Login*, tidak memiliki relasi *one-to-one* atau *one-to-many* dengan entitas lain. Hal ini disebabkan hanya *admin* yang menggunakan login dan dapat mengelola apapun dalam **SISTEM INFORMASI KOPERASI PALA OBILATU**.
2. Entitas *Setoran*, memiliki relasi dengan *one-to-one* pada entitas *Anggota*. Hal ini disebabkan satu anggota hanya bisa melakukan satu setoran begitupun sebaliknya.
3. Entitas *Penjualan*, memiliki relasi dengan *one-to-one* pada entitas *Toko*. Hal ini disebabkan harus satu toko memiliki satu penjualan sebaliknya hanya satu penjualan dalam satu toko.

4. Entitas Setoran, memiliki relasi *one-to-many* dengan entitas Jenis Produk. Hal ini disebabkan banyak setoran memiliki satu jenis produk dan sebaliknya satu jenis produk memiliki banyak setoran dari anggota.
5. Entitas Setoran, memiliki relasi *one-to-many* dengan entitas Periode. Hal ini disebabkan banyak setoran dalam satu periode sebaliknya satu periode memiliki banyak setoran.
6. Entitas Penjualan, memiliki relasi *one-to-many* dengan entitas Jenis Produk, karena banyak penjualan memiliki satu jenis produk dan sebaliknya jenis produk dapat di miliki banyak penjualan.
7. Entitas Penjualan, memiliki relasi *one-to-one* dengan entitas Periode. Karena banyak penjualan dapat dilakukan dalam satu periode sebaliknya satu periode hanya memiliki satu penjualan.

3.4.4. Class Diagram



Gambar 3. 14 Class Diagram

Class diagram yang ada pada sistem berupa sebuah *controller* dari aplikasi yang dibuat, dimana setiap *controller* memiliki fungsi yang hampir sama yaitu:

1. *Index* untuk menampilkan tabel dari *database*.

2. *CreateAwal* untuk memasukan *value* data awal ke dalam *create*.
3. *Create* untuk menambah data ke dalam *database*.
4. *Delete* untuk menghapus data dari *database*.
5. *DeleteConfirmed* untuk meneruskan kehalaman tertentu setelah data berhasil dihapus.
6. *Details* untuk menampilkan detail dari sebuah data dalam sebuah tabel.
7. *Dispose* untuk merilis *resource* pada aplikasi.
8. *Edit* untuk mengubah data dari sebuah database.
9. *Export* untuk mengubah data dalam bentuk PDF, yang mana ini dapat di gunakan untuk *print*.

Sedangkan beberapa *class* ada yang memiliki fungsi - fungsi khusus. Berikut *class* dan fungsi khususnya yaitu:

1. Pada class *AccountController* :
 - a. *AccountController* berfungsi untuk mendeklarasikan *userManager* dan *signManager*.
 - b. *Login* berfungsi untuk masuk kedalam aplikasi *web*.
 - c. *Register* berfungsi untuk mendaftarkan pengelola.
 - d. *ConfirmEmail* berfungsi untuk mengkonfirmasi *email*.
 - e. *ForgotPassword* berfungsi untuk lupa *password*.
 - f. *ResetPassword* untuk memperbaharui *password*.
 - g. *ResetPasswordConfirmed* untuk konfirmasi *reset password*.
 - h. *AddError* berfungsi memberikan *error*.
 - i. *LogOff* berfungsi untuk keluar dari akun.

Berikut adalah penjelasan tentang *class* diagram pada gambar 3.14

1. Seluruh *class* yang ada didalam sistem ini terinisialisasi menjadi model tabel yang didasarkan pada database yang di bangun.
2. *Class* Anggota, Jenis Produk, Periode , Alamat Toko merupakan *class* yang berdiri sendiri dan diasosiasi oleh *class* lain.
3. *Class* Setoran berfungsi untuk menyimpan data setoran anggota seperti data id anggota, data tanggal setoran, data periode setoran, data jenis produk setoran, dan jumlah setoran dalam kilogram.

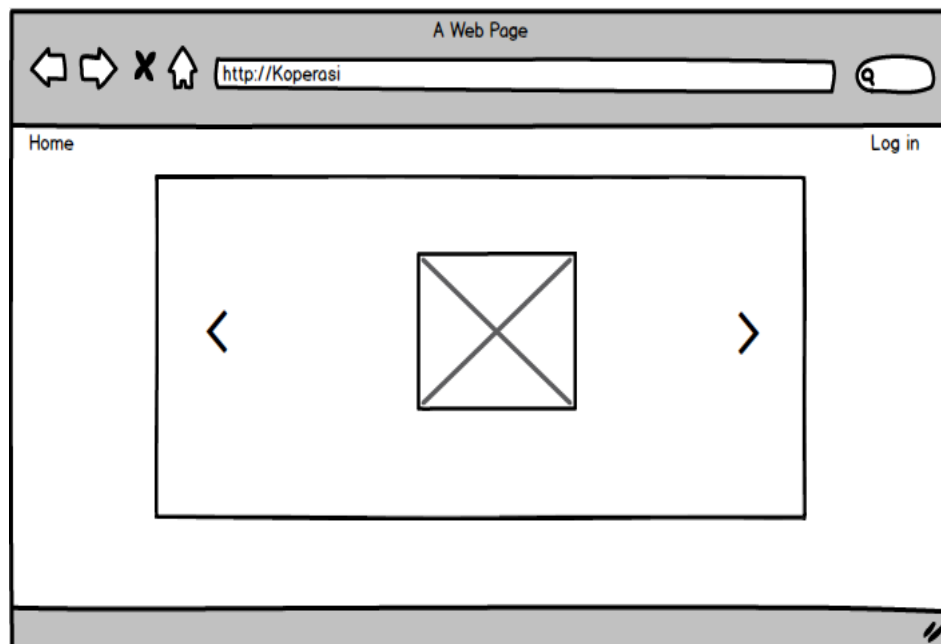
4. *Class* Penjualan berfungsi untuk menyimpan data penjualan seperti data alamat toko penjualan, data tanggal penjualan, data periode penjualan, data jenis produk penjualan dan data harga perkilogram.

3.5. Rancangan Antarmuka

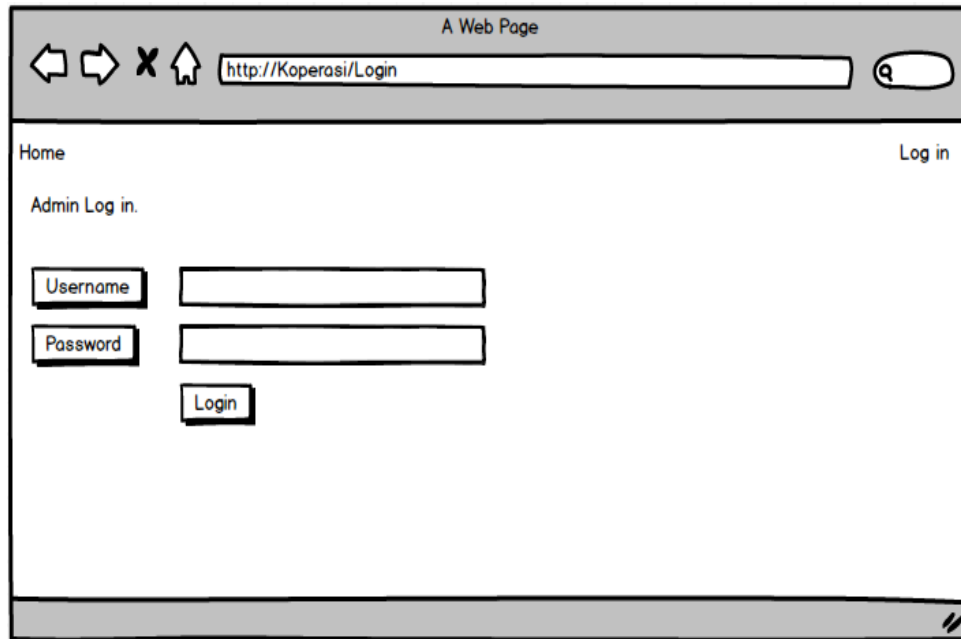
Dalam membangun sebuah *aplikasi web* rancangan antarmuka sangatlah penting. Dengan merancang antarmuka, akan memberikan kemudahan dalam mendesain *aplikasi web*. Antarmuka akan digunakan sebagai penghubung antara *user* dengan sistem *aplikasi web*. *Aplikasi web* akan memberikan *feedback* berupa antarmuka pada setiap kegiatan *user*. Komunikasi ini yang akan membuat sistem *aplikasi web* berjalan.

3.5.1. Rancangan Antarmuka Halaman Awal dan Login

Halaman awal adalah tampilan awal saat mengakses *aplikasi web* SISTEM INFORMASI KOPERASI PALA OBILATU. Semua *user* akan mendapatkan tampilan yang sama pada halaman awal, gambar rancangan halaman awal dapat dilihat pada gambar 3.15. Setelah halaman awal terdapat rancangan *login*. *Login* adalah autentikasi awal untuk *admin* mengelola data. Gambar rancangan *login* dapat dilihat pada gambar 3.16. Kemudian yang terakhir adalah registrasi. Pada halaman registrasi, ini hanya bisa diakses *admin*. Karena registrasi hanya untuk registrasi menjadi seorang *admin*. Gambar registrasi dapat dilihat pada gambar 3.17.



Gambar 3. 15 Halaman Awal



A Web Page

http://Koperasi/Login

Home Log in

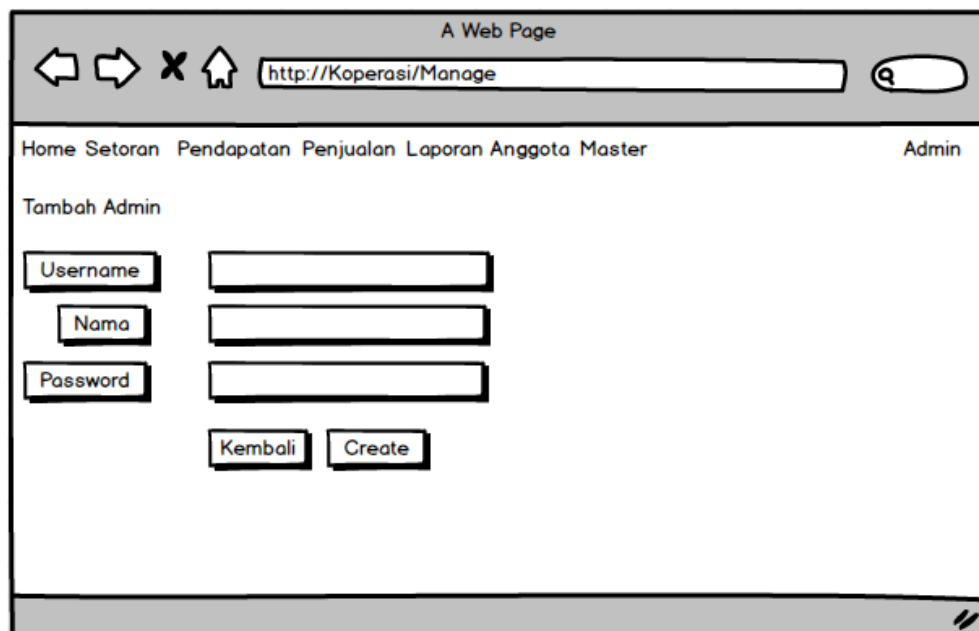
Admin Log in.

Username

Password

Login

Gambar 3. 16 Login



A Web Page

http://Koperasi/Manage

Home Setoran Pendapatan Penjualan Laporan Anggota Master Admin

Tambah Admin

Username

Nama

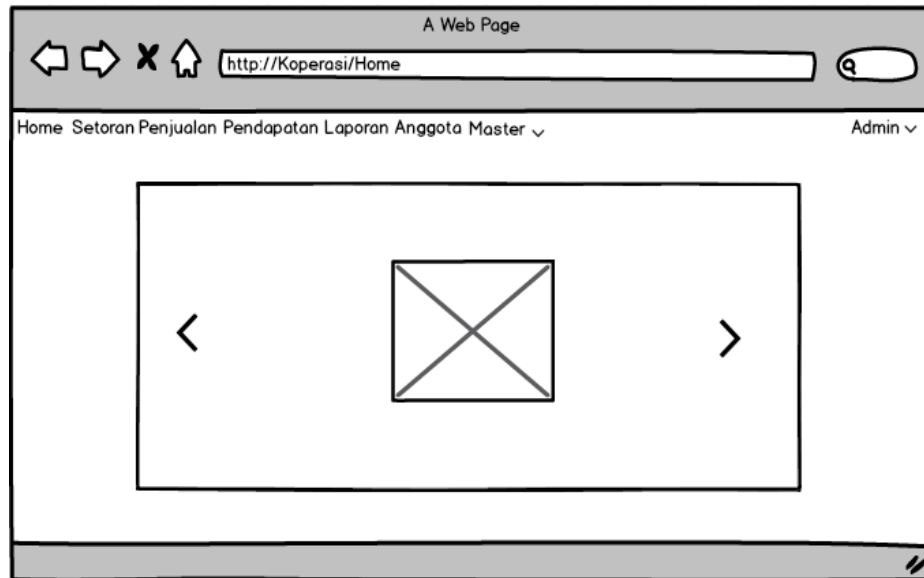
Password

Kembali Create

Gambar 3. 17 Registrasi Admin

3.5.2. Rancangan Antarmuka Halaman Utama

Setelah melakukan *login* (hanya *admin*) maka akan mendapatkan *feedback* halaman utama. Pada halaman ini terdapat menu-menu yang dapat terlihat setelah diautentikasi sebagai *admin*. Rancangan gambar halaman utama dapat dilihat pada gambar 3.18.



Gambar 3. 18 Halaman Utama

Pada gambar 3.18 terdapat menu-menu yang dapat dibuka oleh *admin*. Fungsi menu dapat dilihat pada tabel 3.1

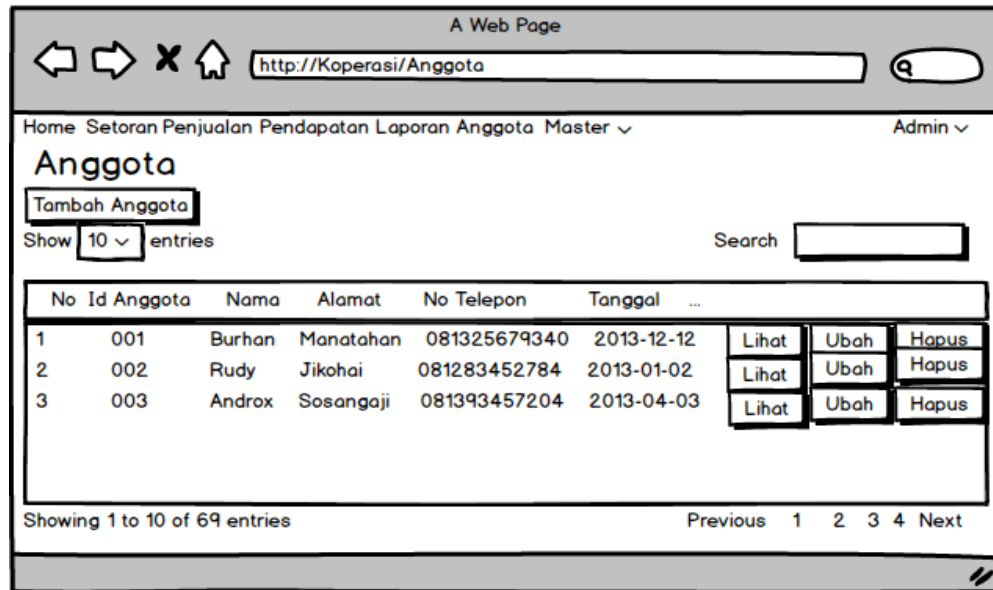
Tabel 3. 1 Menu Halaman Utama

| NO | Menu | Sub Menu | Keterangan |
|----|------------|----------|--------------------------------------------------------------------------------------------------------------------|
| 1 | Home | - | Menu home adalah sebuah <i>action link</i> untuk kembali ke halaman utama. |
| 2 | Setoran | - | Sub menu ini berfungsi menambahkan jumlah setorang masing-masing anggota berdasarkan periode dan jenis produk. |
| 3 | Penjualan | - | Sub menu ini berfungsi menambahkan penjualan berdasarkan toko, jenis produk serta harga dalam suatu periode. |
| 4 | Pendapatan | - | Sub menu ini berfungsi untuk menapilakan pendatan dari jumlah setoran anggota dan jumla harga dalam suatu periode. |

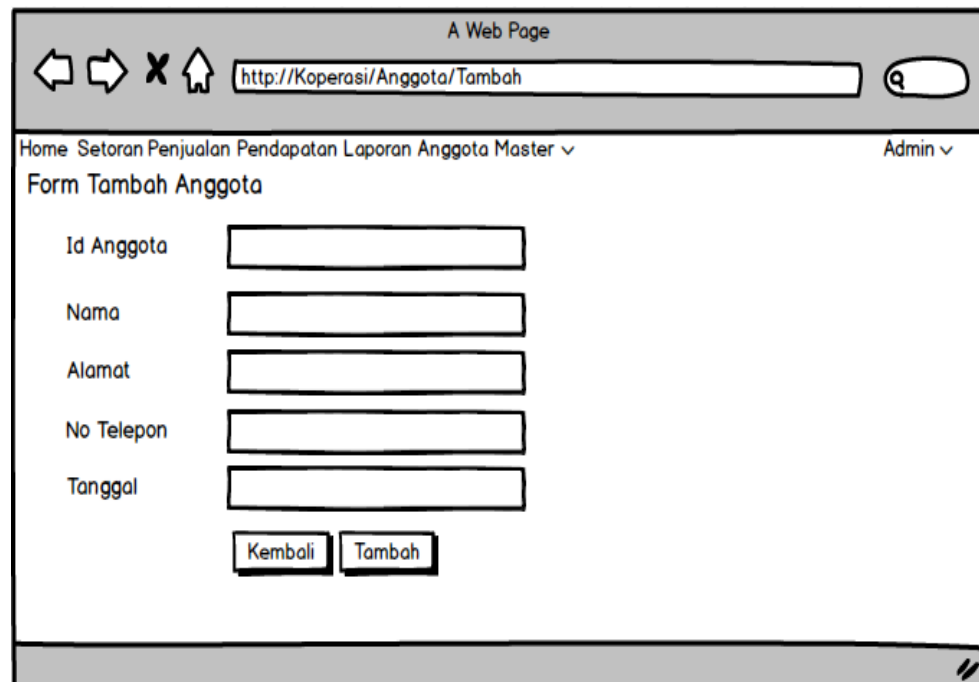
| | | | |
|---|-----------------|-------------------|--------------------------------------------------------------------------------------------------------------------|
| 5 | Laporan Anggota | - | Sub menu ini berfungsi sebagai hasil data laporan anggota dalam satu periode, seperti data setoran dan pendapatan. |
| 6 | Master | Anggota | Sub menu ini berfungsi menambahkan Anggota untuk pilihan setoran dan pendapatan. |
| | | Nama Jenis Produk | Sub menu ini berfungsi menambahkan Jenis produk untuk pilihan setoran, penjualan serta pendapatan. |
| | | Nama Periode | Sub menu ini berfungsi menambahkan Nama Periode untuk pilihan setoran, penjualan dan pendapatan. |
| | | Alamat Toko | Sub menu ini berfungsi menambahkan Alamat Toko untuk pilihan penjualan dan pendapatan. |
| 7 | Admin/Log in | Manage Admin | Sub menu ini berfungsi untuk menambahkan admin baru, atau menghapus data admin yang lama. |
| | | Log Out | Sub menu ini berfungsi untuk keluar dari aplikasi. |

3.5.3. Rancangan Antarmuka Halaman Anggota

Rancangan antarmuka anggota merupakan halaman yang menampilkan seluruh data anggota seperti data id, nama, alamat, no telepon serta tanggal yang dicatat oleh *admin*. *Admin* dapat mengelola data anggota seperti melihat data detail, menghapus data, dan mengubah data. Gambaran rancangan antarmuka halaman anggota dapat dilihat pada gambar di 3.19. Selain itu *admin* juga bisa menambahkan data baru untuk anggota baru, dengan mengisi data id anggota, nama, alamat, no telepon, dan tanggal kemudian disimpan. Rancangan antarmuka untuk halaman tambah anggota dapat dilihat pada gambar 3.20.



Gambar 3. 19 Halaman Anggota

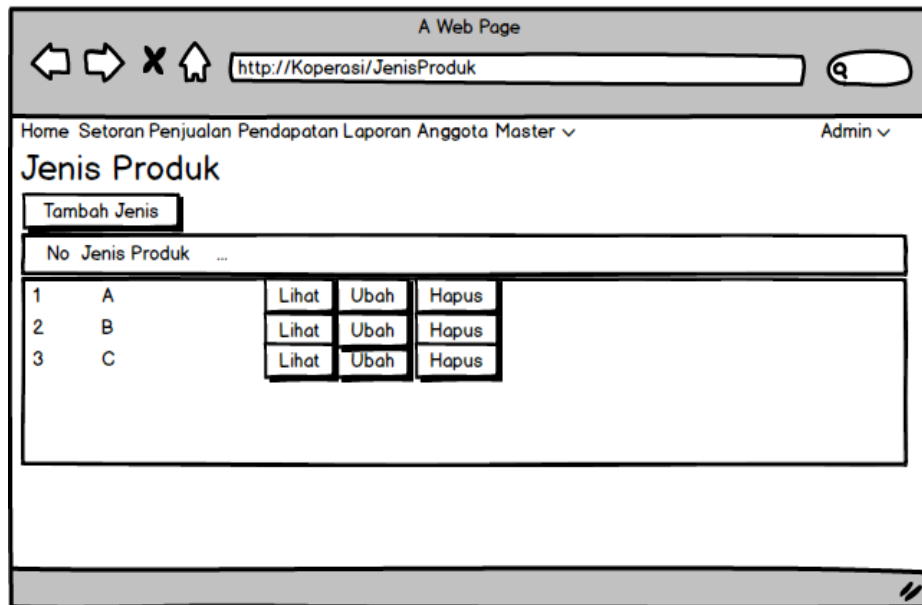


Gambar 3. 20 Halaman Tambah Anggota

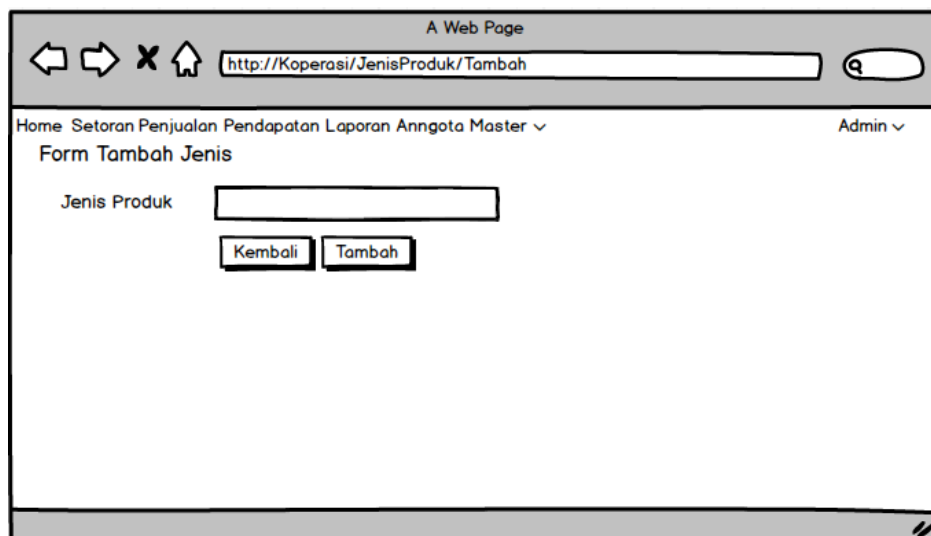
3.5.4. Rancangan Antarmuka Halaman Jenis Produk

Rancangan antarmuka jenis produk merupakan halaman yang menampilkan seluruh data jenis produk seperti data jenis produk A, B, C, serta Fuli yang dicatat oleh *admin*. *Admin* dapat mengelola data jenis produk seperti melihat data detail, menghapus data, dan mengubah data. Gambaran rancangan antarmuka halaman jenis produk dapat dilihat pada gambar 3.21. Selain itu *admin* juga bisa menambahkan data baru untuk jenis produk baru, dengan mengisi data

nama jenis produk baru, kemudian disimpan. Rancangan antarmuka untuk halaman tambah jenis produk dapat dilihat pada gambar 3.22.



Gambar 3. 21 Halaman Jenis Produk

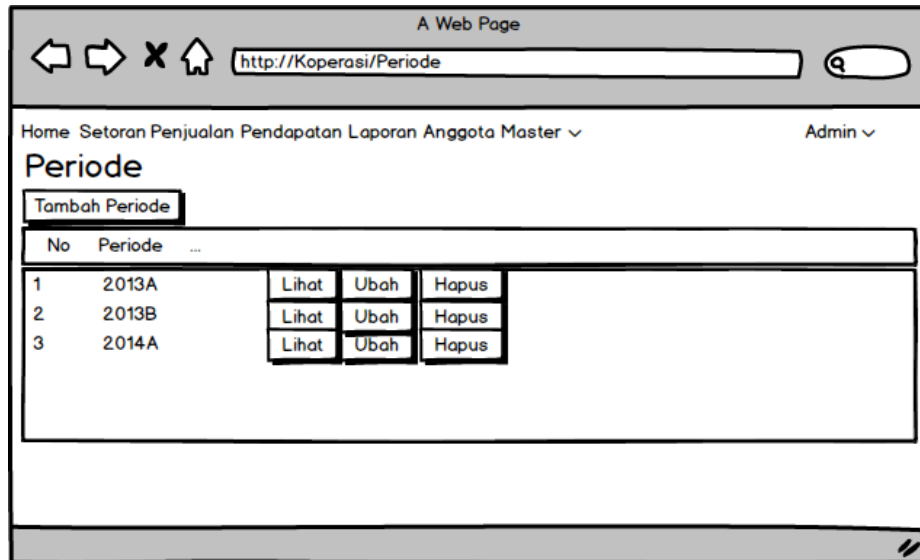


Gambar 3. 22 Halaman Tambah Jenis Produk

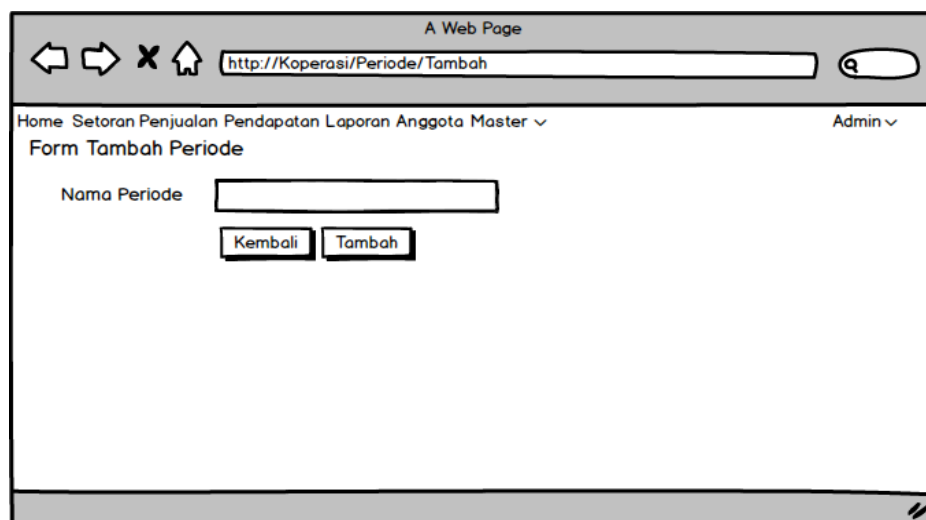
3.5.5. Rancangan Antarmuka Halaman Nama Periode

Rancangan antarmuka periode merupakan halaman yang menampilkan seluruh data periode seperti data periode 2015A, 2015B, yang dicatat oleh *admin*. *Admin* dapat mengelola data periode seperti melihat data detail, menghapus data, dan mengubah data. Gambaran rancangan antarmuka halaman periode dapat dilihat pada gambar 3.23. Selain itu *admin* juga bisa menambahkan data baru untuk periode baru, dengan mengisi data nama periode baru,

kemudian disimpan. Rancangan antarmuka untuk halaman tambah periode dapat dilihat pada gambar 3.24.



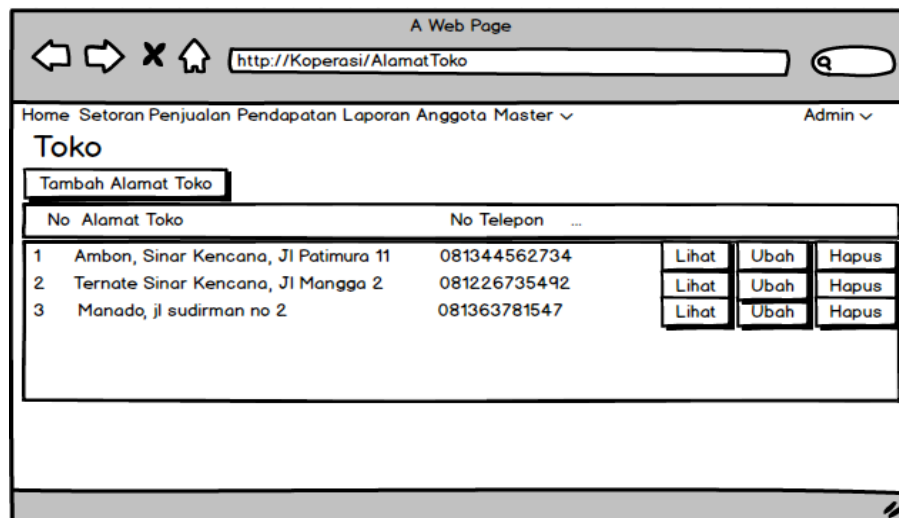
Gambar 3. 23 Halaman Nama Periode



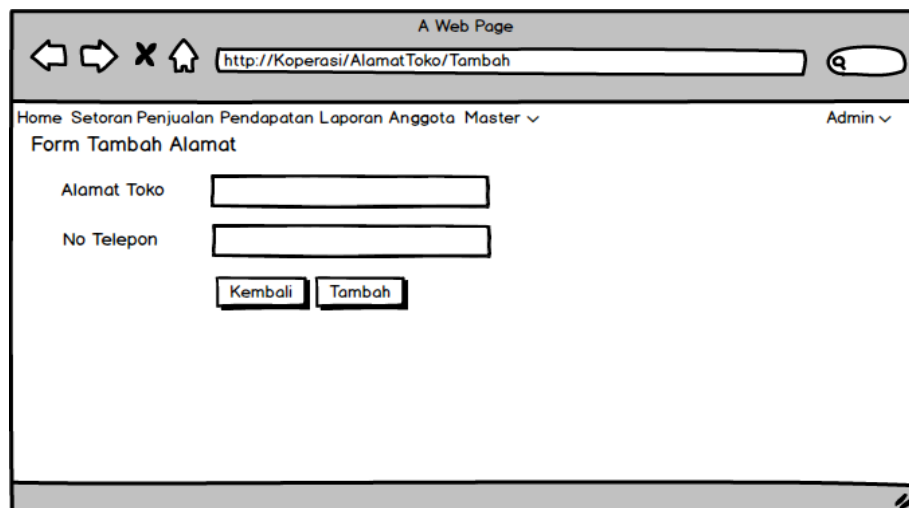
Gambar 3. 24 Halaman Tambah Nama Periode

3.5.6. Rancangan Antarmuka Halaman Toko

Rancangan antarmuka toko merupakan halaman yang menampilkan seluruh data toko seperti data alamat toko dan no telepon, yang dicatat oleh *admin*. *Admin* dapat mengelola data toko seperti melihat data detail, menghapus data, dan mengubah data. Gambaran rancangan antarmuka halaman toko dapat dilihat pada gambar 3.25. Selain itu *admin* juga bisa menambahkan data baru untuk toko baru, dengan mengisi data alamat toko baru dan no telepon baru, kemudian disimpan. Rancangan antarmuka untuk halaman tambah toko dapat dilihat pada gambar 3.26.



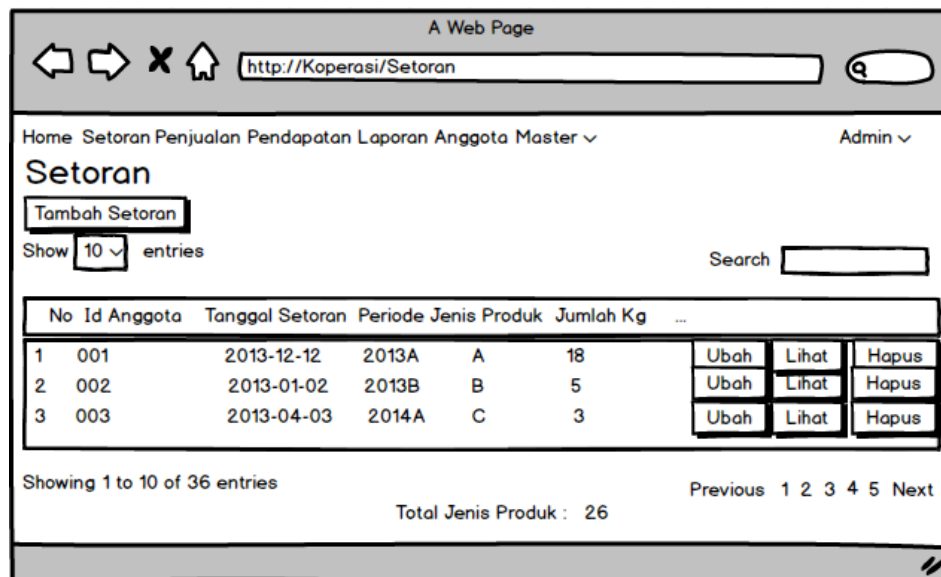
Gambar 3. 25 Halaman Toko



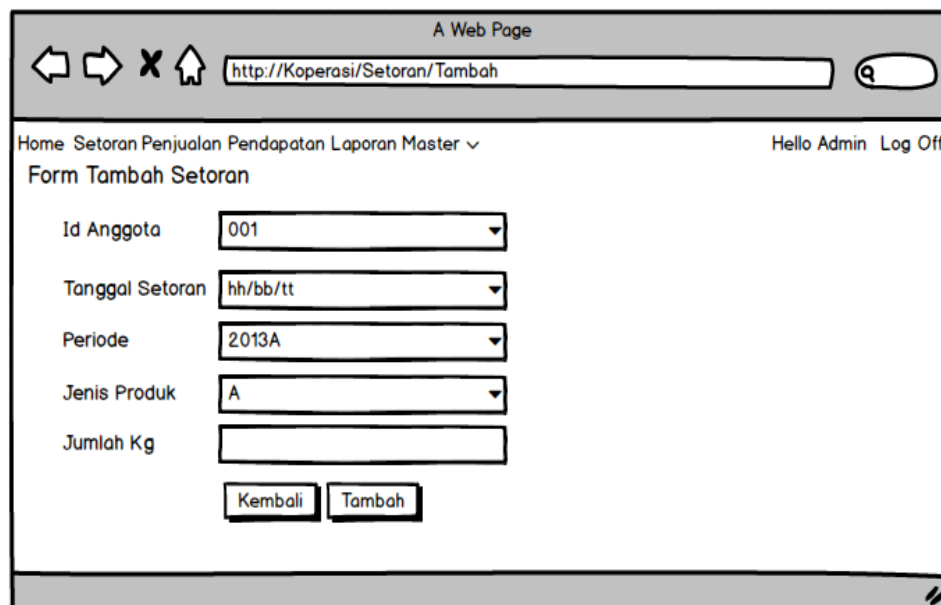
Gambar 3. 26 Halaman Tambah Toko

3.5.7. Rancangan Antarmuka Halaman Setoran

Rancangan antarmuka setoran merupakan halaman yang menampilkan seluruh data setoran seperti data id anggota, tanggal setoran, periode setoran, jenis produk, jumlah kilogram setoran, selain itu juga ada fitur search yang berfungsi mengetahui jumlah total setoran jenis produk dalam setiap periode. *Admin* dapat mengelola data setoran seperti melihat data detail, menghapus data, dan mengubah data. Gambaran rancangan antarmuka halaman setoran dapat dilihat pada gambar 3.27. Selain itu *admin* juga bisa menambahkan data baru untuk setoran baru, dengan mengisi data id anggota, tanggal setoran, periode setoran, jenis produk, jumlah kilogram setoran, kemudian disimpan. Rancangan antarmuka untuk halaman tambah setoran dapat dilihat pada gambar 3.28.



Gambar 3. 27 Halaman Setoran

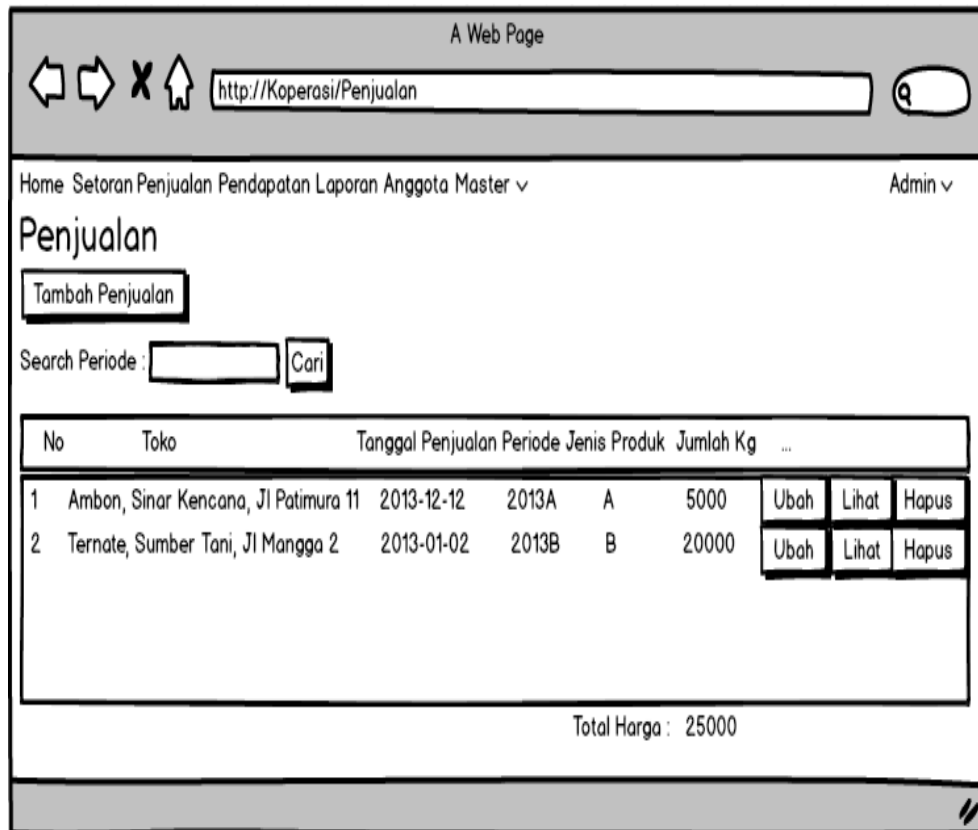


Gambar 3. 28 Halaman Tambah Setoran

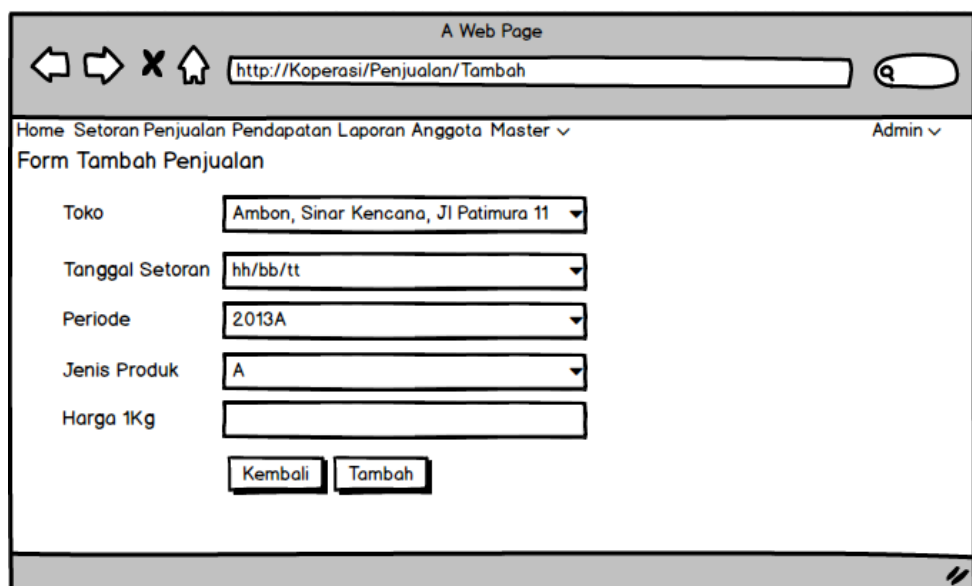
3.5.8. Rancangan Antarmuka Halaman Penjualan

Rancangan antarmuka penjualan merupakan halaman yang menampilkan seluruh data penjualan seperti data alamat toko, tanggal penjualan, periode penjualan, jenis produk, harga per kilogram, selain itu juga ada fitur search periode yang berfungsi mengetahui jumlah total harga dalam setiap periode. *Admin* dapat mengelola data penjualan seperti melihat data detail, menghapus data, dan mengubah data. Gambaran rancangan antarmuka halaman penjualan dapat dilihat pada gambar 3.29. Selain itu *admin* juga bisa menambahkan data baru untuk penjualan baru, dengan mengisi data alamat toko, tanggal penjualan, periode penjualan, jenis

produk, harga per kilogram, kemudian disimpan. Rancangan antarmuka untuk halaman tambah penjualan dapat dilihat pada gambar 3.30.



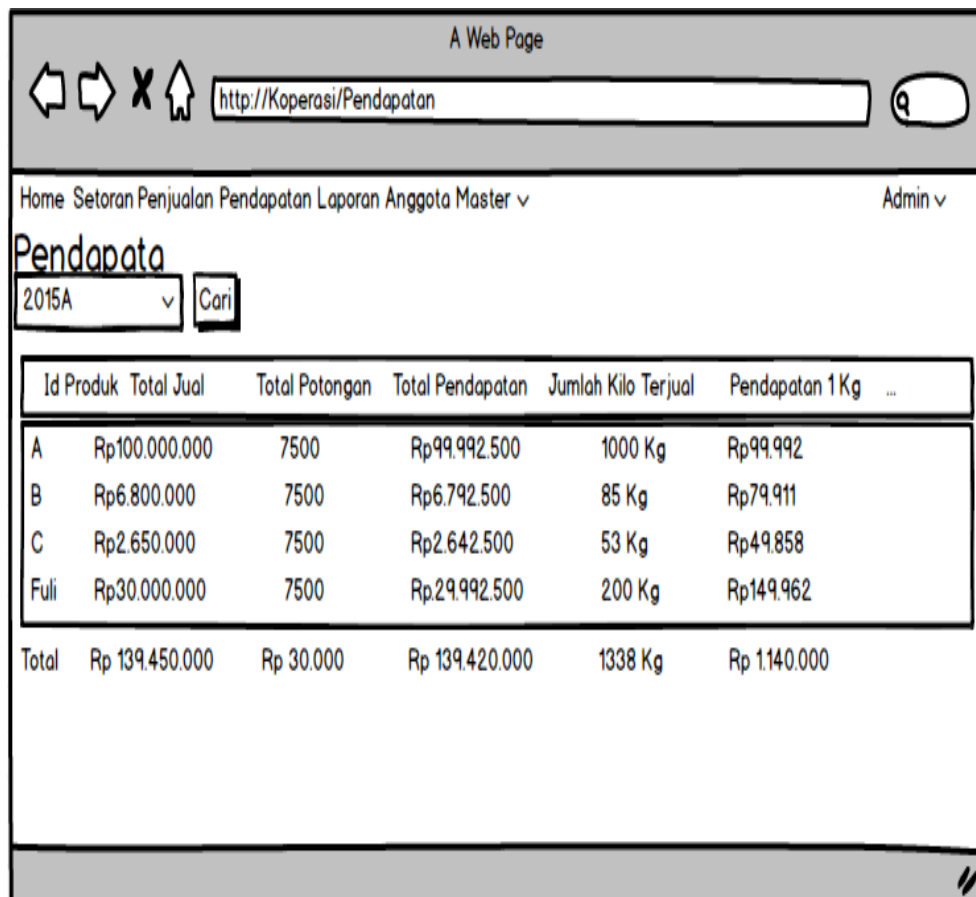
Gambar 3. 29 Halaman Penjualan



Gambar 3. 30 Halaman Tambah Penjualan

3.5.9. Rancangan Antarmuka Halaman Pendapatan

Rancangan antarmuka pendapatan merupakan halaman yang menampilkan seluruh data pendapatan dari jumlah setoran anggota dan harga penjualan seperti data pendapatan jenis produk, total jual, total potongan, total pendapatan, jumlah kilo terjual, pendapatan satu kilo gram. Data ini tidak bisa diedit atau dihapus karna bersifat otomatis yang diambil dari data setoran dan harga penjualan dalam satu periode. Gambaran rancangan antarmuka halaman pendapatan dapat dilihat pada gambar 3.31.



| Id Produk | Total Jual | Total Potongan | Total Pendapatan | Jumlah Kilo Terjual | Pendapatan 1 Kg | ... |
|-----------|----------------|----------------|------------------|---------------------|-----------------|-----|
| A | Rp100.000.000 | 7500 | Rp99.992.500 | 1000 Kg | Rp99.992 | |
| B | Rp6.800.000 | 7500 | Rp6.792.500 | 85 Kg | Rp79.911 | |
| C | Rp2.650.000 | 7500 | Rp2.642.500 | 53 Kg | Rp49.858 | |
| Fuli | Rp30.000.000 | 7500 | Rp29.992.500 | 200 Kg | Rp149.962 | |
| Total | Rp 139.450.000 | Rp 30.000 | Rp 139.420.000 | 1338 Kg | Rp 1.140.000 | |

Gambar 3. 31 Pendapatan

3.5.10. Rancangan Antarmuka Halaman Laporan Anggota

Rancangan antarmuka laporan anggota merupakan halaman yang menampilkan seluruh data setoran dan pendapatan anggota antara lain seperti id anggota, tanggal setoran, nama anggota, nama toko, periode, jenis produk jumlah setoran, harga penjualan satu kilogram, pendapatan, potongan serta pendapatan bersih yang harus diteriama oleh anggota tersebut dan langsung bisa diprint berformat *PDF* dan diserahkan kepada anggota tersebut. Gambaran rancangan antarmuka halaman pendapatan dapat dilihat pada gambar 3.32.

| IdAngta | Tglsetor | NmAngta | NmTk | Prd | JnsPrdk | JmlhStr | HrgPnjln1Kg | Pndptn | Ptngn | PndptnBrsh | ... |
|---------|------------|---------|-------|-------|-----------|---------|-------------|--------------|----------|--------------|-----|
| 001 | 05/05/2015 | Burhan | Ambon | 2015A | Pala A | 650 | Rp100.000 | Rp65.000.000 | Rp2.500 | Rp64.997.500 | |
| 001 | 05/05/2015 | Burhan | Ambon | 2015A | Pala B | 50 | Rp80.000 | Rp4.000.000 | Rp2.500 | Rp3.997.500 | |
| 001 | 05/05/2015 | Burhan | Ambon | 2015A | Pala C | 35 | Rp50.000 | Rp1.750.000 | Rp2.500 | Rp1.747.500 | |
| 001 | 05/05/2015 | Burhan | Ambon | 2015A | Pala Fuli | 130 | Rp150.000 | Rp19.500.000 | Rp2.500 | Rp19.497.500 | |
| Total | | | | | | | | Rp90.250.000 | Rp10.000 | Rp90.240.000 | |

Gambar 3. 32 Halaman Laporan Anggota

3.6. Pengujian

Menurut (M. Sidi Mustaqbal, Roeri Fajri Firdaus, Hendra Rahmadi, 2015) dengan judul “PENGUJIAN APLIKASI MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS”. Pengujian adalah suatu proses pelaksanaan suatu program dengan tujuan menemukan suatu kesalahan. Suatu kasus *test* yang baik adalah apabila *test* tersebut mempunyai kemungkinan menemukan sebuah kesalahan yang tidak terungkap. Suatu *test* yang sukses adalah bila *test* tersebut membongkar suatu kesalahan yang awalnya tidak ditemukan. Salah satu dari jenis pengujian yang ada adalah *Black Box Testing*.

Pada penelitian ini menggunakan metode *black box* untuk menguji serta mengevaluasi bagaimana *website* ini berjalan. Tujuan pengujian pada *website* ini adalah untuk mengetahui apakah *website* Koperasi Pala Obilatu telah memenuhi kebutuhan yang diperlukan oleh *admin* sebagai media untuk pendataan setoran, penjualan serta pendaptan kepada anggota koperasi.

Metode *Black box testing* atau tes fungsional adalah pengujian yang dilakukan hanya dengan mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari *website* tersebut. Pengujian program dalam *website - website* ini dilakukan oleh calon *admin* dan pengurus koperasi yang terlibat untuk memberikan data yang akan di-*input* ke dalam *website* koperasi. Setelah itu pengunjung dapat melihat bagaimana *website* tersebut berjalan untuk menyampaikan informasi mengenai data setoran, penjualan serta pendapatan di Koperasi Pala Obilatu yang dapat diakses melalui *internet*.

