

**PEMBUATAN *WEBSITE* LAPORAN PENJUALAN KOPERASI
DI RRI BOGOR**

TUBAGUS ARIQ NAUFAL



**PROGRAM STUDI MANAJEMEN INFORMATIKA
SEKOLAH VOKASI
INSTITUT PERTANIAN BOGOR
BOGOR
2022**

PERNYATAAN MENGENAI LAPORAN AKHIR DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA

Dengan ini saya menyatakan bahwa laporan akhir dengan judul “Pembuatan Aplikasi Pelaporan Koperasi Jual Beli Berbasis Website Di RRI Bogor” adalah karya saya dengan arahan dari dosen pembimbing dan belum diajukan dalam bentuk apa pun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir laporan akhir ini.

Dengan ini saya melimpahkan hak cipta dari karya tulis saya kepada Institut Pertanian Bogor.

Bogor, Juni 2022

Tubagus Ariq Naufal
J3C219181

RINGKASAN

TUBAGUS ARIQ NAUFAL. Pembuatan *Website* Laporan Penjualan Koperasi Modul Admin di RRI Bogor (*Making of the Cooperative Sales Report Website at RRI Bogor*). Dibimbing oleh INNA NOVIANTY.

RRI Bogor merupakan lembaga penyiaran radio nasional yang dapat dijangkau seluruh Indonesia dengan menyiarkan berbagai bidang kegiatan, seperti siaran hiburan, pendidikan, kebudayaan, baik berita nasional maupun internasional. Dalam penyiarannya, RRI Bogor memiliki fokus tujuan menjaga dan mencerminkan budaya bangsa.

RRI Bogor memiliki berbagai divisi diantaranya adalah Lembaga Pengembangan Usaha yang menangani koperasi dan pemesanan iklan pada penyiaran. Dalam penanganan koperasi divisi Lembaga Pengembangan Usaha menyusun serta membuat laporan akhir pembagian Sisa Hasil Usaha (SHU) dari karyawan yang telah terdaftar menjadi anggota koperasi. Proses dokumentasi laporan koperasi sebelumnya dilakukan pencatatan secara manual melalui buku besar dan *software* Excel. Hal itu menyebabkan lambatnya proses dokumentasi dan memiliki kemungkinan besar kesalahan perhitungan, sehingga perlu dilakukannya sistem digitalisasi agar proses dokumentasi dapat menghemat waktu dan tersimpan secara otomatis oleh sistem.

Pembuatan *website* laporan penjualan koperasi merupakan aplikasi berbasis *website* yang menyediakan pengelolaan hasil penjualan yang terjadi di koperasi yang dapat menjadi acuan untuk pemotongan gaji dan pembagian Sisa Hasil Usaha (SHU) anggota koperasi karyawan RRI Bogor. Transaksi yang dilakukan akan terdata dalam bentuk tabel sehingga pengguna dapat melihat riwayat hasil transaksi. Transaksi yang terlibat hanya anggota yang telah terdaftar dalam keanggotaan koperasi. *Website* laporan penjualan koperasi dapat memanipulasi data anggota, data barang dan data transaksi, seperti menambah, mengubah dan menghapus. Pengelolaan sistem ini hanya memiliki satu jenis aktor yaitu admin, nantinya dapat dilakukan penambahan aktor admin jika diperlukan.

Modul yang digunakan dalam pembuatan aplikasi pelaporan koperasi berbasis *website* adalah metode *Prototype*, yang terdiri dari beberapa langkah yaitu komunikasi, perancangan cepat, pemodelan perencanaan cepat, pembentukan *prototype* dan penyerahan dan umpan balik. Pembuatan *website* ini menggunakan *framework* Codeigniter dan Bahasa pemrograman PHP serta *database* MySQL. Fitur-fitur yang tersedia pada aplikasi pelaporan koperasi berbasis *website* adalah Login, grafik penjualan barang, grafik pembelian terbaru, CRUD anggota, CRUD barang, CRUD pembelian, melihat riwayat pembelian, download *report*, melihat *report*, export *report*, dan *logout*.

Pengguna yang terlibat dalam *website* laporan penjualan aplikasi yaitu pegawai koperasi yang bekerja sebagai pengelola koperasi di RRI Bogor. Pegawai koperasi dapat memanipulasi data anggota, barang, transaksi, melihat diagram penjualan dan diagram pendapatan pertahun. Pengguna juga dapat mengunduh laporan transaksi dalam bentuk PDF maupun *excel* sebagai data untuk acuan pemotongan gaji dan pembagian SHU anggota koperasi di RRI Bogor.

Kata kunci: Aplikasi, Koperasi, Pelaporan SHU, PHP, Prototype, *website*

© Hak Cipta milik IPB, tahun 2022
Hak Cipta dilindungi Undang-Undang

Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan atau menyebutkan sumbernya. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik, atau tinjauan suatu masalah, dan pengutipan tersebut tidak merugikan kepentingan IPB.

Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apa pun tanpa izin IPB.

**PEMBUATAN *WEBSITE* LAPORAN PENJUALAN KOPERASI
MODUL ADMIN DI RRI BOGOR**

TUBAGUS ARIQ NAUFAL

Laporan Akhir
sebagai salah satu syarat untuk memperoleh gelar
Ahli Madya pada
Program Studi Manajemen Informatika

**PROGRAM STUDI MANAJEMEN INFORMATIKA
SEKOLAH VOKASI
INSTITUT PERTANIAN BOGOR
BOGOR
2022**

Penguji pada ujian Laporan Akhir: Aep Setiawan, S.Si., M.Si.

Judul Laporan : Pembuatan *Website* Laporan Penjualan Koperasi Di RRI Bogor

Nama : Tubagus Ariq Naufal

NIM : J3C219181

Disetujui oleh

Pembimbing:

Dr. Inna Novianty, S.Si., M.Si.

Diketahui oleh

Ketua Program Studi:

Sofiyanti Indriasari, S.Kom., M.Kom.

NPI 201807198410052001

Dekan Sekolah Vokasi:

Prof. Dr. Ir. Arief Darjanto, M.Ec.

NIP 196108181986091001

Tanggal Ujian:

17 Juni 2022

Tanggal Lulus:

PRAKATA

Puji dan syukur dipanjatkan kepada Allah *Subhanaahu Wa ta'ala* atas segala karunia-Nya sehingga Tugas Akhir berhasil diselesaikan. Tema yang dipilih dalam Praktik Kerja Lapangan (PKL) yang dilaksanakan sejak bulan Januari sampai bulan Maret 2022 ialah sistem informasi dengan judul “Pembuatan *Website* Laporan Penjualan Koperasi Modul Admin di RRI Bogor”.

Oleh karena itu, ucapan terima kasih disampaikan kepada pihak yang telah berperan dalam penyelesaian tugas akhir, yaitu:

1. Allah SWT yang telah memberikan hikmah, rahmat, dan karunia-Nya dalam penyelesaian Tugas Akhir.
2. Tubagus Akhsani sebagai ayah kandung, Yati Ruhayati sebagai ibu kandung, Tubagus Hasan Fauzi sebagai kakak, Tubagus Ahnaf faeyza dan Tubagus Abiyyu Raki sebagai adik, yang senantiasa mendoakan, mendukung dan memberikan semangat selama masa perkuliahan maupun dalam penyusunan tugas akhir.
3. Sofiyanti Indriasari, S.Kom., M.Kom. selaku Ketua Program Studi Manajemen Informatika Sekolah Vokasi Institut Pertanian Bogor.
4. Dosen Manajemen Informatika Sekolah Vokasi Institut Pertanian Bogor yang telah memberikan ilmu selama kegiatan perkuliahan.
5. Dr. Inna Novianty, S.Si., M.Si., sebagai dosen pembimbing yang telah memberikan bimbingan dan saran serta masukan dalam penyusunan tugas akhir.
6. RRI Bogor yang telah menerima memberikan kesempatan untuk melaksanakan kegiatan PKL.
7. Hadi Irawan, S.Kom., selaku pembimbing lapangan yang telah membantu, membimbing, serta memberikan ilmunya selama kegiatan PKL dilaksanakan.
8. Seluruh teman seperjuangan Manajemen Informatika angkatan 56 yang telah banyak membantu selama masa perkuliahan.

Semoga Tugas Akhir ini dapat bermanfaat bagi pihak yang membutuhkan, khususnya untuk Program Studi Manajemen Informatika di Sekolah Vokasi Institut Pertanian Bogor.

Bogor, Juni 2022

Tubagus Ariq Naufal

DAFTAR ISI

DAFTAR TABEL	x
DAFTAR GAMBAR	x
DAFTAR LAMPIRAN	xi
I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	3
1.4 Manfaat	3
1.5 Ruang Lingkup	3
II METODE	4
2.1 Lokasi dan Waktu PKL	4
2.2 Prosedur Kerja	4
III KEADAAN UMUM PERUSAHAAN	8
3.1 Sejarah	8
3.2 Kegiatan Lembaga	8
3.3 Struktur Organisasi	8
3.4 Fungsi dan Tujuan	9
IV HASIL DAN PEMBAHASAN PEMBUATAN <i>WEBSITE</i> LAPORAN PENJUALAN KOPERASI	10
4.1 <i>Prototype</i> Iterasi Pertama	10
4.2 <i>Prototype</i> Iterasi Kedua	30
V SIMPULAN DAN SARAN	39
5.1 Simpulan	39
5.2 Saran	39
DAFTAR PUSTAKA	40
LAMPIRAN	41
RIWAYAT HIDUP	97

DAFTAR TABEL

1 Simbol <i>Use Case Diagram</i>	5
2 Simbol <i>Class Diagram</i>	6
3 Simbol <i>Activity Diagram</i>	6
4 Kebutuhan Fungsional	12
5 <i>Use Case Scenario Login</i>	15
6 <i>Use Case Scenario</i> Iterasi Pertama	15
7 Deskripsi Tabel <i>tbl_admin</i>	17
8 Deskripsi Tabel Iterasi Pertama	17
9 Perancangan Antarmuka	19
10 Perancangan Masukan	19
11 <i>Activity Diagram</i> Iterasi Pertama	20
12 Perancangan Keluaran Iterasi Pertama	22
13 Implementasi Proses Iterasi Pertama	26
14 Implementasi Antarmuka Iterasi Pertama	27
15 Implementasi Masukan Iterasi Pertama	28
16 Implementasi Keluaran Iterasi Pertama	29
17 Hasil Pengujian <i>Login</i>	29
18 Hasil Pengujian Iterasi Pertama	30
19 Analisis Kebutuhan Fungsional Iterasi Kedua	31
20 Hasil Pengujian Melihat Data Laporan Berdasarkan Tanggal	38
21 Hasil Pengujian Melihat Grafik Pendapatan	38

DAFTAR GAMBAR

1 Tahapan Metodologi <i>Prototyping</i>	4
2 Struktur Organisasi RRI Bogor	8
3 Proses Bisnis	11
4 <i>Use Case Diagram</i> iterasi pertama	14
5 Perancangan <i>Class Diagram</i>	16
6 Perencanaan <i>Database</i>	17
7 Rancangan Antarmuka Halaman <i>Login</i>	18
8 Rancangan Antarmuka Halaman Beranda	18
9 Perancangan Masukan Tambah Data Barang	19
10 <i>Activity Diagram Login</i>	20
11 Perancangan Keluaran Tabel Data Barang	22
12 Implementasi <i>Database</i>	23
13 Implementasi Antarmuka <i>Login</i>	26
14 Implementasi Antarmuka Beranda	27
15 Implementasi Masukan Tambah Data Barang	28
16 Implementasi Keluaran Tabel Data Barang	29
17 <i>Use Case Diagram</i> Iterasi Kedua	32
18 Perancangan Proses Melihat Data Laporan Berdasarkan Tanggal	32
19 Perancangan Proses Melihat Grafik Pendapatan	32

20 Perancangan Antarmuka Melihat Data Laporan Berdasarkan Tanggal	33
21 Perancangan Antarmuka Melihat Grafik Pendapatan	33
22 Perancangan Masukan Melihat Data Laporan Berdasarkan Tanggal	
Perancangan Keluaran	33
23 Perancangan Keluaran Melihat Data Laporan	34
24 Implementasi Proses Melihat Data Laporan Berdasarkan Tanggal	35
25 Implementasi Proses Melihat Grafik Pendapatan	36
26 Implementasi Antarmuka Melihat Data Laporan Berdasarkan Tanggal	37
27 Implementasi Antarmuka Melihat Grafik Pendapatan	37
28 Implementasi Masukan Iterasi Kedua	37
29 Implementasi Keluaran Iterasi Kedua	38

DAFTAR LAMPIRAN

1 <i>Use Case Scenario</i> Menambah Akun	42
2 <i>Use Case Scenario</i> Mengubah Akun	42
3 <i>Use Case Scenario</i> Menghapus Akun	42
4 <i>Use Case Scenario</i> Menambah Data Anggota	43
5 <i>Use Case Scenario</i> Mengubah Data Anggota	43
6 <i>Use Case Scenario</i> Menghapus Data Anggota	44
7 <i>Use Case Scenario</i> Menambah Data Barang	44
8 <i>Use Case Scenario</i> Mengubah Data Barang	45
9 <i>Use Case Scenario</i> Menghapus Data Barang	45
10 <i>Use Case Scenario</i> Menambah Data Pembelian	46
11 <i>Use Case Scenario</i> Mengubah Data Pembelian	46
12 <i>Use Case Scenario</i> Menghapus Data Pembelian	47
13 <i>Use Case Scenario</i> Melihat Data Laporan	47
14 <i>Use Case Scenario</i> Mengunduh Data Laporan	47
15 <i>Use Case Scenario</i> Melihat Data Akun <i>Admin</i>	48
16 <i>Use Case Scenario</i> Melihat Data Anggota	48
17 <i>Use Case Scenario</i> Melihat Data Barang	48
18 <i>Use Case Scenario</i> Melihat Data Pembelian	49
19 <i>Use Case Scenario</i> Melihat Grafik Penjualan	49
20 <i>Use Case Scenario</i> Melihat Penjualan Barang Perbulan	49
21 <i>Use Case Scenario</i> Melihat Presentase Jenis Pembeli	50
22 <i>Use Case Scenario</i> Melihat <i>Detail</i> Data Pembelian	50
L23 <i>Use Case Scenario Logout</i>	50
24 Deskripsi Tabel <i>tbl_barang</i>	51
25 Deskripsi Tabel <i>tbl_anggota</i>	51
26 Deskripsi Tabel <i>tbl_pembelian</i>	52
27 Deskripsi Tabel <i>tbl_transaksi</i>	52
28 Perancangan Antarmuka Halaman Profil <i>Admin</i>	53
29 Perancangan Antarmuka Halaman Data Anggota	53
30 Perancangan Antarmuka Halaman Data Barang	54
31 Perancangan Antarmuka Halaman Data Pembelian	54
32 Perancangan Antarmuka Halaman Laporan	54
33 Perancangan Masukkan Tambah Akun <i>Admin</i>	54

34 Perancangan Masukkan Ubah Akun <i>Admin</i>	55
35 Perancangan Masukkan Tambah Data Anggota	55
36 Perancangan Masukkan Ubah Data Anggota	55
37 Perancangan Masukkan Ubah Data Barang	56
38 Perancangan Masukkan Tambah Data Pembelian	56
39 Perancangan Masukkan Ubah Data Pembelian	56
40 Perancangan Masukkan <i>Login</i>	57
41 <i>Activity Diagram</i> Menambah Akun <i>Admin</i>	57
42 <i>Activity Diagram</i> Mengubah Akun <i>Admin</i>	58
43 <i>Activity Diagram</i> Menghapus Akun <i>Admin</i>	58
44 <i>Activity Diagram</i> Menambah Data Anggota	59
45 <i>Activity Diagram</i> Mengubah Data Anggota	59
46 <i>Activity Diagram</i> Menghapus Data Anggota	60
47 <i>Activity Diagram</i> Menambah Data Barang	60
48 <i>Activity Diagram</i> Mengubah Data Barang	61
49 <i>Activity Diagram</i> Menghapus Data Barang	61
50 <i>Activity Diagram</i> Menambah Data Pembelian	62
51 <i>Activity Diagram</i> Mengubah Data Pembelian	62
52 <i>Activity Diagram</i> Menghapus Data Pembelian	63
53 <i>Activity Diagram</i> Melihat Data Laporan	63
54 <i>Activity Diagram</i> Mengunduh Data Laporan	64
55 <i>Activity Diagram</i> Melihat Data Akun <i>Admin</i>	64
56 <i>Activity Diagram</i> Melihat Data Anggota	65
57 <i>Activity Diagram</i> Melihat Data Barang	65
58 <i>Activity Diagram</i> Melihat Data Pembelian	66
59 <i>Activity Diagram</i> Melihat Grafik Penjualan	66
60 <i>Activity Diagram</i> Melihat Data Penjualan Barang Perbulan	67
61 <i>Activity Diagram</i> Presentasi Jenis Pembeli	67
62 <i>Activity Diagram</i> Melihat Detil Data Pembelian	68
63 <i>Activity Diagram Logout</i>	68
64 Perancangan Keluaran Berhasil Menambah Data	69
65 Perancangan Keluaran Berhasil Mengubah Data	69
66 Perancangan Keluaran Konfirmasi Menghapus Data	69
67 Perancangan Keluaran Berhasil Menghapus Data	69
68 Perancangan Keluaran Tabel Data Akun <i>Admin</i>	69
69 Perancangan Keluaran Tabel Data Anggota	69
70 Perancangan Keluaran Tabel Data Pembelian	70
71 Perancangan Keluaran Tabel Data Laporan	70
72 Implementasi Proses Halaman Beranda	70
73 Implementasi Proses Halaman Profil <i>Admin</i>	71
74 Implementasi Proses Halaman Data Anggota	74
75 Implementasi Proses Halaman Data Pembelian	76
76 Implementasi Proses Halaman Laporan	81
77 Implementasi Antarmuka Profil <i>Admin</i>	84
78 Implementasi Antarmuka Halaman Data Anggota	84
79 Implementasi Antarmuka Halaman Data Barang	85
80 Implementasi Antarmuka Halaman Data Pembelian	85
81 Implementasi Antarmuka Halaman Laporan	85

82 Implementasi Masukkan Tambah Akun <i>Admin</i>	85
83 Implementasi Masukkan Ubah Profil <i>Admin</i>	86
84 Implementasi Masukkan Tambah Data Anggota	86
85 Implementasi Masukkan Ubah Data Anggota	86
86 Implementasi Masukkan Ubah Data Barang	86
87 Implementasi Masukkan Tambah Data Pembelian	87
88 Implementasi Masukkan Ubah Data Pembelian	87
89 Implementasi Keluaran Berhasil Menambah Data	87
90 Implementasi Keluaran Berhasil Mengubah Data	87
91 Implementasi Keluaran Konfirmasi Menghapus Data	87
92 Implementasi Keluaran Berhasil Menghapus Data	87
93 Implementasi Keluaran Tabel Data Akun <i>Admin</i>	88
94 Implementasi Keluaran Tabel Data Anggota	88
95 Implementasi Keluaran Tabel Data Pembelian	88
96 Implementasi Keluaran Tabel Data Laporan	88
97 Hasil Pengujian Menambah Akun <i>Admin</i>	89
98 Hasil Pengujian Mengubah Akun <i>Admin</i>	89
99 Hasil Pengujian Menghapus Akun	90
100 Hasil Pengujian Menambah Data Anggota	90
101 Hasil Pengujian Mengubah Data Anggota	91
102 Hasil Pengujian Menghapus Data Anggota	91
103 Hasil Pengujian Menambah Data Barang	91
104 Hasil Pengujian Mengubah Data Barang	92
105 Hasil Pengujian Menghapus Data Barang	92
106 Hasil Pengujian Menambah Data Pembelian	93
107 Hasil Pengujian Mengubah Data Pembelian	93
108 Hasil Pengujian Menghapus Data Pembelian	93
109 Hasil Pengujian Melihat Data Laporan	94
110 Hasil Pengujian Mengunduh Data Laporan	94
111 Hasil Pengujian Melihat Data Akun <i>Admin</i>	94
112 Hasil Pengujian Melihat Data Anggota	94
113 Hasil Pengujian Melihat Data Barang	94
114 Hasil Pengujian Melihat Data Pembelian	95
115 Hasil Pengujian Melihat Grafik Penjualan	95
116 Hasil Pengujian Melihat Data Penjualan Barang Perbulan	95
117 Hasil Pengujian Melihat Presentase Jenis Pembeli	95
118 Hasil Pengujian Melihat Detil Data Pembelian	96
119 Hasil Pengujian <i>Logout</i>	96

I PENDAHULUAN

1.1 Latar Belakang

Perkembangan zaman yang terjadi saat ini begitu pesat. Saat ini begitu banyak perkembangan yang mendorong kita untuk mengikuti zaman yang semakin modern dan terotomatisasi. Sistem informasi menjadi salah satu perkembangan yang terjadi saat ini. Instansi atau organisasi menjadi dampak dari perkembangan yang setiap hari mencatat data tertentu yang semakin hari kian banyak. Sehingga susahnyanya menyimpan catatan data dengan menyimpan secara manual dengan tulisan tangan. Sistem informasi menjadi salah satu efek pendorong perkembangan teknologi yang kian pesat. Sistem informasi adalah sistem didalam suatu organisasi yang membantu kebutuhan pengelolaan dan pengoperasian transaksi harian, bersifat manajerial dan strategis untuk suatu organisasi, dan menyediakan pihak tertentu terkait laporan yang dibutuhkan (Hutahaeen 2014). Dalam implementasinya, sistem informasi menyediakan berbagai macam untuk memenuhi kebutuhan dan mengatasi permasalahan yang sesuai, salah satunya penyimpanan data. Proses penyimpanan data seperti laporan dan riwayat dapat terdokumentasi dengan adanya sistem informasi.

Website adalah media yang terhubung dengan jaringan *internet* yang terdiri dari penghubung alamat satu ke alamat lain dengan menggunakan bahasa pemrograman (Prasetyo 2015). *Website* menjadi salah satu contoh implementasi sistem informasi. *Website* dapat menyimpan, mengelola, dan menampilkan informasi. Manfaat tersebut dibutuhkan dalam membuat sebuah proses dokumentasi. Hal ini menjadi penting dalam setiap proses bisnis, salah satunya koperasi jual beli.

Koperasi adalah perkumpulan orang yang memiliki tujuan bersama untuk mensejahterakan ekonomi anggotanya melalui badan usaha yang dikelola secara demokratis (Puspitasari 2015). Proses bisnis yang dilakukan koperasi yaitu koperasi jual beli terdapat riwayat data penjualan barang yang perlu didokumentasikan dengan baik. Penerapan sistem informasi dapat menyediakan wadah sebagai tempat penyimpanan data riwayat penjualan, sehingga riwayat data penjualan dapat dilihat kembali tanpa khawatir dengan hilang atau rusaknya data penjualan yang masih dalam bentuk kertas atau buku. Pendataan tersebut berguna untuk menghitung hasil akhir seperti sisa hasil usaha (SHU) untuk anggota yang terlibat. Dalam lingkungan tertentu, anggota akan mendapatkan laba dari SHU. Koperasi karyawan menjadi contoh lingkungan tertentu yang nantinya setiap anggota karyawan akan mendapatkan bagian laba dari badan usaha koperasi tersebut. Perhitungan yang dilakukan untuk mendapat laba secara adil dibuat dan disesuaikan dengan lingkungan yang terlibat. Koperasi karyawan yang berada di RRI Bogor melakukan perhitungan sesuai dengan jumlah barang dan harga yang dibeli.

RRI Bogor merupakan lembaga penyiaran radio nasional yang memberikan berita baik dalam lingkup nasional maupun internasional yang dapat dijangkau di seluruh Indonesia. RRI Bogor memiliki beberapa divisi diantaranya Lembaga Pengembangan Usaha (LPU). Salah satu tugasnya, LPU menangani koperasi dalam melakukan manajemen data penjualan koperasi. Kegiatan manajemen data yang dilakukan LPU membutuhkan pencatatan data penjualan koperasi untuk mendapatkan sebuah informasi. Informasi yang didapat berupa riwayat pembelian

karyawan dalam melakukan transaksi dengan koperasi, seperti nama barang, jumlah, harga, waktu, atau lainnya. Data tersebut akan dicatat dan dijadikan salah satunya laporan Sisa Hasil Usaha (SHU). Sebelumnya, proses manajemen data masih dilakukan secara manual dengan mencatat di buku. Lalu untuk menyimpan dan mengelolanya menggunakan *Microsoft Excel*. Hal tersebut membuat manajemen data penjualan memakan waktu yang cukup banyak. Tingkat keamanan pun cukup beresiko karena data bisa saja hilang atau rusak. Pada kajian ini dibuatlah *Website* Laporan Penjualan Koperasi di RRI Bogor yang dapat mengatasi proses *report* secara digital. Laporan yang dihasilkan berupa pencatatan data dari hasil riwayat pembelian yang telah di *input*.

Fitur yang disediakan dalam *website* laporan penjualan koperasi berupa mengelola data admin, mengelola anggota koperasi, mengelola data barang dan stok, mengelola data pembelian, melihat laporan serta menghitung rekap penjualan, dan melihat grafik penjualan. Fitur admin dapat menambah, mengubah dan menghapus data admin, seperti mengelola nama, email dan *password*. Terdapat juga fitur lupa *password* apabila pengguna lupa akun untuk membuka sistem dengan memasukkan *email* yang telah terdaftar di akun. Fitur barang dapat menambah, mengubah dan menghapus data barang. Atribut yang ada yaitu, nama, kode, stok, foto dan harga barang. Fitur anggota dapat menambah, mengubah dan menghapus data anggota. Atribut yang ada yaitu, nama, dan kode anggota. Fitur pembelian dapat membuat, mengubah dan menghapus data pembelian. Fitur ini menampilkan tabel riwayat pembelian serta detail dari setiap pembelian barang. Atribut yang ada yaitu, nama pembeli, jumlah barang, total harga dan tanggal pembelian. Menambah data pembelian menampilkan seluruh barang yang tersedia dan anggota yang melakukan pembelian. Fitur laporan dapat menampilkan data rekap pembelian barang yang dilakukan. Halaman laporan memberikan fitur untuk melihat data laporan setiap tahun dan data laporan sesuai dengan tanggal pembelian.

Dibuatnya *website* laporan penjualan koperasi ini diharapkan dapat menjadi wadah untuk menyediakan penyimpanan data barang dan stok, serta data penjualan koperasi. Dengan demikian, riwayat pembelian dapat terdokumentasi dengan baik dan menghitung hasil akhir pertahun yaitu sisa hasil usaha (SHU) dan menghitung jumlah pengeluaran tiap anggota perbulan sebagai acuan untuk pemotongan gaji setiap anggota koperasi.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, rumusan masalah dalam Pembuatan *Website* Laporan Penjualan Koperasi di RRI Bogor adalah sebagai berikut:

1. Bagaimana merancang sistem informasi Administrator pada *website* yang dapat membantu dalam membuat laporan penjualan?
2. Bagaimana merancang sistem informasi Administrator pada *website* yang dapat mengelola data barang dan data penjualan?

1.3 Tujuan

Tujuan yang ingin dicapai dari pembuatan *website* Laporan Penjualan Koperasi di RRI Bogor, yaitu:

1. Menyediakan fitur laporan yang berfungsi menyimpan hasil rekap penjualan koperasi.
2. Menyediakan sistem pengelolaan barang dan penjualan koperasi.

1.4 Manfaat

Manfaat dari pembuatan *website* Laporan Penjualan Koperasi di RRI Bogor, yaitu:

1. Dapat membantu melihat hasil rekap penjualan dengan fitur laporan.
2. Mengelola data barang dan data penjualan koperasi yang diunggah oleh admin.
3. Dapat melihat data barang yang terjual dalam bentuk grafik sehingga dapat dijadikan acuan keputusan.

1.5 Ruang Lingkup

Manfaat dari pembuatan *website* Laporan Penjualan Koperasi di RRI Bogor, yaitu:

1. Aplikasi ini berbasis *website*.
2. *Website* dibuat menggunakan *web application framework* Codeigniter dengan *Database* MySQL.
3. Sistem yang dibuat adalah modul *Admin*.
4. Aktor hanya *Admin* sebagai pengelola *website*.
5. Aktor *Admin* telah dibuat saat proses pengembangan.

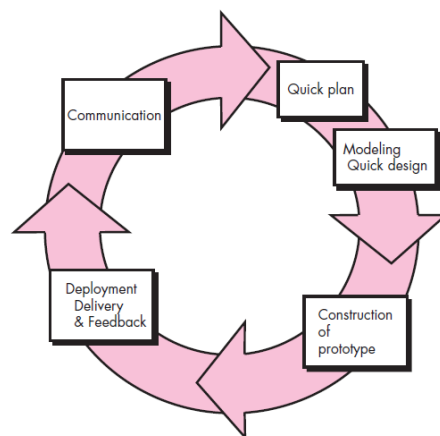
II METODE

2.1 Lokasi dan Waktu PKL

Kegiatan Praktik Kerja Lapangan (PKL) di RRI Bogor dilaksanakan secara *hybrid* dalam 3 hari seminggu dikarenakan terjadinya pandemi Covid-19. Waktu pelaksanaan PKL dimulai dari tanggal 3 Januari – 18 Maret 2022. Kegiatan dilakukan setiap hari kerja yaitu Senin s/d Jumat dari pukul 08.00 sampai 15.00 WIB.

2.2 Prosedur Kerja

Pembuatan *Website* Laporan Penjualan Koperasi di RRI Bogor ini menggunakan metodologi *prototype*. Hal yang mempengaruhi penggunaan metode *prototype* yaitu kondisi klien yang belum mengerti tentang apa saja fungsi-fungsi yang ingin ditampilkan pada *website*. Dalam hal ini, metodologi *prototype* menawarkan pendekatan yang cukup baik. Dengan metodologi *prototype*, perlu adanya pendekatan secara komunikasi dengan klien agar *website* yang dibuat sesuai dengan harapan yang diinginkan. Metodologi *prototype* merupakan model yang digunakan untuk memberikan penjelasan dengan klien terhadap teknis dan kebutuhan yang diberikan agar sistem dibuat dapat sesuai dengan yang diinginkan (Firmansyah Y et al. 2019). Tahapan metodologi *prototyping* dapat dilihat pada Gambar 1.



Gambar 1 Tahapan Metodologi *Prototyping*

Tahap *Communication* atau komunikasi melakukan pengumpulan data awal, dan analisis kebutuhan. *Quick Plan* atau perencanaan cepat membuat perencanaan kebutuhan. *Modelling Quick Design* atau pemodelan perencanaan cepat membuat perancangan *prototype*. *Construction of Prototype* atau pembentukan *prototype* melakukan pembuatan *prototype* sistem dan pengujian sistem. *Deployment Delivery & Feedback* melakukan evaluasi *prototype* dan mendapatkan umpan balik dari pengguna. Perbaikan *prototype* dilakukan berdasarkan hasil evaluasi *prototype* dari pengguna (Aditya et al. 2021).

2.2.1 Komunikasi

Tahap komunikasi melakukan komunikasi dengan pengembang dan klien terhadap permasalahan, kebutuhan dan tujuan yang diinginkan (Syaddad 2017). Tahap komunikasi dilakukan identifikasi permasalahan, kebutuhan yang diperlukan dan alur sistem yang diinginkan klien. Alur sistem akan di analisa dan menyesuaikan tujuan yang ingin dicapai.

2.2.2 Perencanaan Cepat




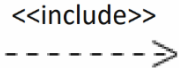
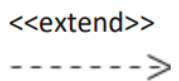
Tahap perencanaan cepat melakukan analisa dan penguraian kebutuhan sitem dari hasil identifikasi permasalahan dan tujuan yang didapat (Aditya *et al.* 2021). Perencanaan terhadap *website* laporan penjualan koperasi dilakukan penguraian kebutuhan yang sudah dianalisa dan didapat dari tahap komunikasi. Penguraian kebutuhan tersebut dievaluasi kembali agar sesuai dengan tujuan yang diharapkan.


2.2.3 Pemodelan Perencanaan Cepat

Tahap pemodelan perencanaan cepat berorientasi pada *user* yang dapat melihat segala aspek sistem dalam bentuk pemodelan, seperti masukan dan keluaran (Manumutu dan Wattimena 2019). Tujuan pemodelan perencanaan cepat untuk memberikan pemahaman untuk *user* terkait alur sistem yang dibuat. *User* dapat melihat pemodelan dari sistem sebagai gambaran alur sistem. Pemodelan yang dibuat antara lain *Use Case Diagram*, *Class Diagram*, dan *Activity Diagram*.

Use Case Diagram adalah diagram yang merepresentasikan kebutuhan sistem dari sudut pandang pengguna (Mandolang *et al.* 2017). Simbol yang digunakan terdapat pada Tabel 1.





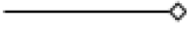
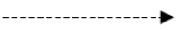
Tabel 1 Simbol *Use Case Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Actor/Aktor</i>	Mempresentasikan aktor yang berinteraksi dalam sistem.
2.		<i>Use Case</i>	Mempresentasikan fungsi sistem
3.		<i>Association/Asosiasi</i>	Mempresentasikan hubungan antara <i>use case</i> dengan aktor.
4.		<i>Include/Melingkupi</i>	Mempresentasikan bagian dari fungsi yang dituju
5.		<i>Extend/Ekstensi</i>	Mempresentasikan hubungan dengan <i>use case</i> lainnya.

No.	Simbol	Nama	Keterangan
6.		<i>Generalization/</i> Generalisasi	Mempresentasikan satu fungsi yang lebih umum dari fungsi lainnya.





Class Diagram adalah diagram yang merepresentasikan struktur *class* serta hubungan antar *class* (Syaddad 2017). *Class Diagram* menggambarkan kerangka sistem yang terdiri dari *class* yang berisi fungsi sebagai pengelolaan sistem. Simbol yang digunakan terdapat pada Tabel 2.



Tabel 2 Simbol *Class Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Association/</i> Asosiasi	Mempresentasikan hubungan antar kelas.
2.		<i>Interface/</i> Antarmuka	Mempresentasikan antarmuka.
3.		<i>Class/</i> Kelas	Mempresentasikan kelas pada sistem.
4.		<i>Generalization/</i> Generalisasi	Mempresentasikan hubungan umum-khusus antar kelas.
5.		<i>Aggregation/</i> Agregasi	Mempresentasikan hubungan semua bagian antar kelas.
6.		<i>Dependency</i>	Mempresentasikan operasi antar kelas.

Activity Diagram adalah Penggambaran alur kerja berupa aktivitas-aktivitas yang terjadi pada sistem (Prinhandoyo 2018). *Activity Diagram* dapat memberikan informasi tentang kegiatan yang terjadi yang berhubungan dengan penerapan fungsi-fungsi yang ada di sistem. Simbol yang digunakan terdapat pada Tabel 3.

Tabel 3 Simbol *Activity Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Activity/</i> Aktivitas	Mempresentasikan aktivitas yang dilakukan sistem.
2.		<i>An Initial Node/</i> Status Awal	Mempresentasikan status awal sistem.
3.		<i>Control Flow</i>	Mempresentasikan alur eksekusi.
4.		<i>Decision/</i> Percabangan	Mempresentasikan asosiasi percabangan <i>use case</i> yang dituju.

No.	Simbol	Nama	Keterangan
5.		<i>Join/Penggabungan</i>	Mempresentasikan asosiasi penggabungan.
6.		<i>Swimlane</i>	Mempresentasikan siapa yang bekerja pada subset tertentu.

2.2.4 Pembentukan *Prototype*

Tahap pembentukan *prototype* merupakan proses implementasi dari pemodelan yang sudah dibuat. Implementasi *website* laporan penjualan koperasi dilakukan dengan menggunakan bahasa pemrograman untuk *server-side* yaitu PHP dengan *framework* Codeigniter. Bahasa pemrograman untuk tampilan yaitu menggunakan *framework* Bootstrap dan Javascript dan basis data MySQL. Pengujian *website* laporan penjualan koperasi dilakukan dengan menggunakan *Black-Box Testing*.

a) PHP

PHP merupakan Bahasa pemrograman untuk *server* yang memiliki kemampuan memproses data secara dinamis. PHP termasuk sebuah *server-side embedded scriptlanguage* yang berarti sintaks dan baris perintah dijalankan oleh *server* akan tetapi dapat tergabung dalam halaman HTML (Usada *et al.* 2012).

b) Codeigniter

Codeigniter adalah *framework* PHP dengan menggunakan konsep *Model, View, Controller* (MVC). *Model* digunakan untuk mengelola *database*. *View* digunakan untuk menampilkan halaman ke *browser*. *Controller* digunakan untuk mengontrol alur aplikasi, termasuk *Model* dan *View* (Ishari *et al.* 2020).

c) Bootstrap

Bootstrap adalah *framework* yang dibangun dengan HTML, CSS yang dapat membuat *website* menarik dan sesuai dengan kebutuhan. (Febrianto *et al.* 2018).

d) Javascript

Javascript adalah bahasa pemrograman untuk *web* yang dapat berinteraksi dan merespon *event* yang dikaitkan dengan HTML. Javascript dapat digunakan baik dari *client-side* maupun *server-side* (Sianipar 2015).

e) MySQL

MySQL merupakan sistem *database* yang penamaannya menggunakan *Structured Query Language* (SQL) dan dapat menerima dan mengirimkan data secara *multi-user* dengan cepat. MySQL dapat digunakan baik dari sisi *client* maupun *server* (Usada *et al.* 2012).

2.2.1 Peluncuran dan Umpan Balik

Pembentukan *prototype* diakhiri dengan pengujian dan umpan balik dari klien. Tahap pengujian dilakukan untuk mengetahui kelayakan sistem agar bisa dipakai dengan baik. Selanjutnya, sistem akan diberikan kepada klien untuk uji coba sistem. Klien memberikan umpan balik terhadap sistem yang telah dibuat. Implementasi uji coba yang dilakukan menggunakan pengujian *Black-Box Testing*.

Black-Box Testing adalah metode pengujian berdasarkan uji coba fungsional dengan memilih semua kemungkinan input untuk menguji hasil yang didapat (Putra *et al.* 2020). Metode *black-box testing* menguji setiap fungsi yang terdapat pada sistem tanpa melihat struktur pemrograman.

III KEADAAN UMUM PERUSAHAAN

3.1 Sejarah

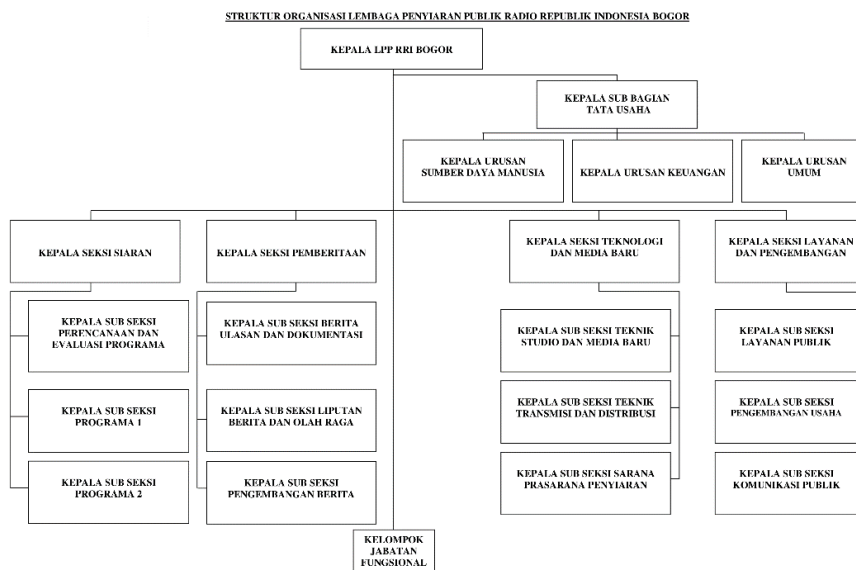
RRI Bogor merupakan lembaga penyiaran radio nasional yang dapat dijangkau seluruh Indonesia dengan menyiarkan berbagai bidang kegiatan, seperti siaran hiburan, pendidikan, kebudayaan, baik berita nasional maupun internasional. Dalam penyiarannya, RRI Bogor memiliki fokus tujuan menjaga dan mencerminkan budaya bangsa. RRI Bogor didirikan pada tanggal 04 Agustus 1968. Kantor pusatnya berada di Jakarta, tepatnya di Jalan Medan Merdeka Barat 4-5, Jakarta Pusat. Saat ini, siaran RRI Bogor mengudara selama 19 jam/hari sejak pukul 05.00 sampai dengan pukul 24.00 WIB, melalui 2 programa yaitu Pro 1 FM 102 dan Pro 2 FM 106,8 di Jalan Pangrango No. 34, Kota Bogor, Jawa Barat, 16151.

3.2 Kegiatan Lembaga

Kegiatan RRI Bogor dimulai pukul 08.00 hingga pukul 17.00 WIB dari hari senin sampai jumat. Waktu lainnya seperti istirahat, sholat, makan diberikan pada pukul 12.00 sampai dengan pukul 13.00 WIB. Kegiatan dilaksanakan di kantor RRI Bogor di Jalan Pangrango No. 34, Kota Bogor, Jawa Barat, 16151.

3.3 Struktur Organisasi

Kepala LPP RRI Bogor memimpin struktur organisasi yang ada di RRI Bogor. Dalam kepemimpinannya, Kepala LPP dibantu oleh sub bagian tata usaha yang dibantu oleh kepala urusan sumber daya manusia, kepala urusan keuangan, dan kepala urusan umum. Setiap kepala yang telah disebut membawahi kembali sub bagian kepala yang dapat dilihat pada Gambar 2.



Gambar 2 Struktur Organisasi RRI Bogor

3.4 Fungsi dan Tujuan

RRI Bogor memiliki tujuan untuk meningkatkan kualitas produk siaran, daya pancar dan kekuatan jaringan. Visi dan misinya adalah sebagai berikut:

a. Visi

Menjadikan LPP RRI radio berjangkauan terluas, pembangun karakter bangsa dan berkelas dunia.

b. Misi

Memberikan pelayanan informasi terpercaya yang dapat menjadi acuan dan sarana kontrol sosial masyarakat dengan memperhatikan kode etik jurnalistik atau kode etik penyiaran, serta mengembangkan siaran pendidikan untuk mencerahkan, mencerdaskan, memberdayakan dan mendorong kreatifitas masyarakat dalam kerangka membangun karakter bangsa.

IV HASIL DAN PEMBAHASAN PEMBUATAN *WEBSITE* LAPORAN PENJUALAN KOPERASI

Pembuatan *website* laporan penjualan koperasi menggunakan metodologi *Prototype*. Dalam pembuatannya, dilakukan pengulangan iterasi. *Website* laporan penjualan koperasi dibuat dengan lima tahapan yaitu tahap komunikasi, perencanaan cepat, pemodelan perancangan cepat, pembentukan *prototype* dan peluncuran sistem dengan umpan balik.

4.1 *Prototype* Iterasi Pertama

4.1.1 Komunikasi (*Communication*)

Tahap Komunikasi dilakukan antara pengembang dan klien yaitu RRI Bogor. Komunikasi bertujuan untuk mendapatkan dan mengumpulkan informasi terkait permasalahan, kebutuhan dan harapan yang diinginkan klien.

Diskusi yang didapatkan dari hasil komunikasi dengan klien antara lain sebagai berikut.

- a) Membuat *website* untuk mengelola data penjualan koperasi
- b) *Website* dapat mengelola dan menampilkan barang yang tersedia di koperasi
- c) Hanya digunakan untuk *admin* yang dapat mengelola *website*
- d) Desain antarmuka sistem dibebaskan untuk proses pembuatannya.

4.1.5 Perencanaan Cepat (*Quick Plan*)

Perencanaan cepat pada iterasi pertama untuk *website* laporan penjualan koperasi yaitu membuat gambaran umum mengenai sistem berupa deskripsi sistem, proses bisnis dan menguraikan analisis kebutuhan pada *website* yang dibuat.

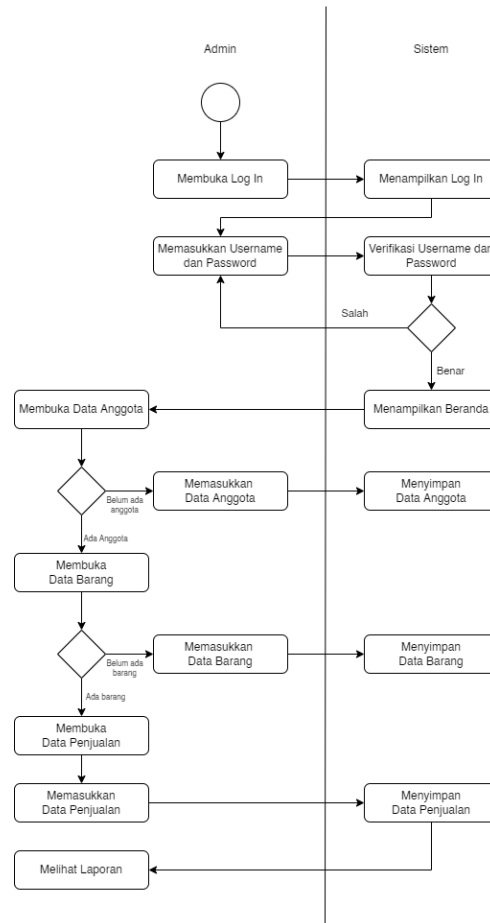
a) Deskripsi

Website laporan penjualan koperasi merupakan *website* yang dapat membantu RRI Bogor dalam mencatat penjualan yang terjadi di koperasi. *Website* ini mirip seperti sistem kasir yang mencatat transaksi dari penjualan barang yang terjual. Akan tetapi lebih difokuskan sebagai laporan penjualan dengan adanya fitur rekap laporan yang berguna untuk menghitung sisa hasil usaha dan jumlah biaya yang dikeluarkan oleh pembeli sebagai pemotongan gaji. Selain itu terdapat fitur pengelolaan barang, daftar anggota koperasi dan grafik penjualan.

b) Proses Bisnis

Proses bisnis *website* laporan penjualan koperasi dapat dilihat pada Gambar 3. Penjelasan proses bisnis dimulai dari pada saat *admin* melakukan *login*, sistem akan memvalidasi form yang dimasukkan saat login. Apabila berhasil *admin* akan masuk dan ditampilkan halaman beranda/*dashboard*. Selanjutnya *admin* dapat mengelola data anggota koperasi yang telah terdaftar. *Admin* pun dapat mengelola data barang yang nantinya akan digunakan sebagai pencatatan transaksi. Apabila barang sudah terdata dalam sistem, *admin* dapat mengelola dan mencatat transaksi yang dilakukan anggota koperasi yang ingin melakukan

pembelian. Data tersebut akan terkumpul dan tersimpan kedalam laporan penjualan. *Admin* dapat melihat data penjualan dimana terdapat rekap perbulan dan pertahun yang memiliki fungsi yang berbeda. Setelah selesai dalam melakukan kelola data, *admin* dapat keluar dari sistem dengan melakukan *logout*.



Gambar 3 Proses Bisnis

- c) Analisis Kebutuhan Perangkat Keras
Spesifikasi dalam pembuatan *website* laporan penjualan koperasi yaitu sebagai berikut.
1. *Space memory* sebesar 100 GB.
 2. Laptop atau PC yang dapat menjalankan dan mengimplementasikan kode program.
 3. RAM minimum 4 GB.
- d) Analisis Kebutuhan Perangkat Lunak
Perangkat lunak yang dibutuhkan dalam pembuatan *website* laporan penjualan koperasi yaitu sebagai berikut.
1. Windows 10 sebagai sistem operasi *device*.
 2. XAMPP sebagai *server* dan *localhost*.
 3. Visual Studio Code sebagai *text/code editor*.
 4. Draw.io sebagai tempat membuat diagram.
 5. Google Chrome sebagai *browser* untuk menguji coba sistem.

e) Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional digunakan untuk memberikan informasi terkait fungsi sistem yang telah dibuat sesuai dengan analisis kebutuhan yang telah dibuat. Penjabaran kebutuhan yang ditampilkan sesuai dengan hasil diskusi dan analisa yang telah dilakukan pada tahap komunikasi. KPR-XXX merupakan penamaan kode fungsional kebutuhan sistem laporan penjualan koperasi (KPR) yang dapat dilihat pada Tabel 3.

Tabel 4 Kebutuhan Fungsional

Kode Fungsi	Nama Fungsi	Deskripsi
KPR-001	Login	Melakukan login ke dalam <i>website</i> .
KPR-002	Menambah Akun <i>Admin</i>	Melakukan <i>input</i> data <i>admin</i> ke dalam sistem.
KPR -003	Mengubah Akun <i>Admin</i>	Melakukan <i>update</i> data akun yang telah dibuat oleh sistem.
KPR -004	Menghapus Akun <i>Admin</i>	Melakukan <i>delete</i> data <i>admin</i> ke dalam sistem.
KPR -005	Menambah Data Anggota	Melakukan <i>input</i> data anggota koperasi ke dalam sistem.
KPR -006	Mengubah Data Anggota	Melakukan <i>update</i> data anggota koperasi ke dalam sistem.
KPR -007	Menghapus Data Anggota	Melakukan <i>delete</i> data anggota koperasi dari sistem.
KPR -008	Menambah Data Barang	Melakukan <i>input</i> data barang koperasi ke dalam sistem.
KPR -009	Mengubah Data Barang	Melakukan <i>update</i> data barang koperasi ke dalam sistem.
KPR -010	Menghapus Data Barang	Melakukan <i>delete</i> data barang koperasi dari sistem.
KPR -011	Menambah Data Pembelian	Melakukan <i>input</i> data pembelian koperasi ke dalam sistem.
KPR-012	Mengubah Data Pembelian	Melakukan <i>update</i> data pembelian ke dalam sistem.
KPR-013	Menghapus Data Pembelian	Melakukan <i>delete</i> data pembelian koperasi dari sistem.

Kode Fungsi	Nama Fungsi	Deskripsi
KPR-014	Melihat Data Laporan	Melihat data laporan penjualan koperasi
KPR-015	Mengunduh Data Laporan	Melakukan pengunduhan data laporan penjualan dalam bentuk file .xls dan .pdf.
KPR-016	Melihat Data Akun <i>Admin</i>	Melihat data akun <i>admin</i> koperasi.
KPR-017	Melihat Data Anggota	Melihat data anggota koperasi.
KPR-018	Melihat Data Barang	Melihat data barang koperasi.
KPR-019	Melihat Data Pembelian	Melihat data pembelian koperasi.
KPR-020	Melihat Grafik Penjualan	Melihat grafik penjualan barang berdasarkan jumlah barang yang terjual.
KPR-021	Melihat Data Penjualan Barang Perbulan	Melihat data barang yang baru terjual.
KPR-022	Melihat Presentase Jenis Pembeli	Melihat presentase pembelian barang anggota dan non-anggota.
KPR-023	Melihat Detail Data Penjualan	Melihat rincian data penjualan barang setiap transaksi.
KPR-024	Logout	Melakukan logout dari <i>website</i> . Melakukan logout dari <i>website</i> .

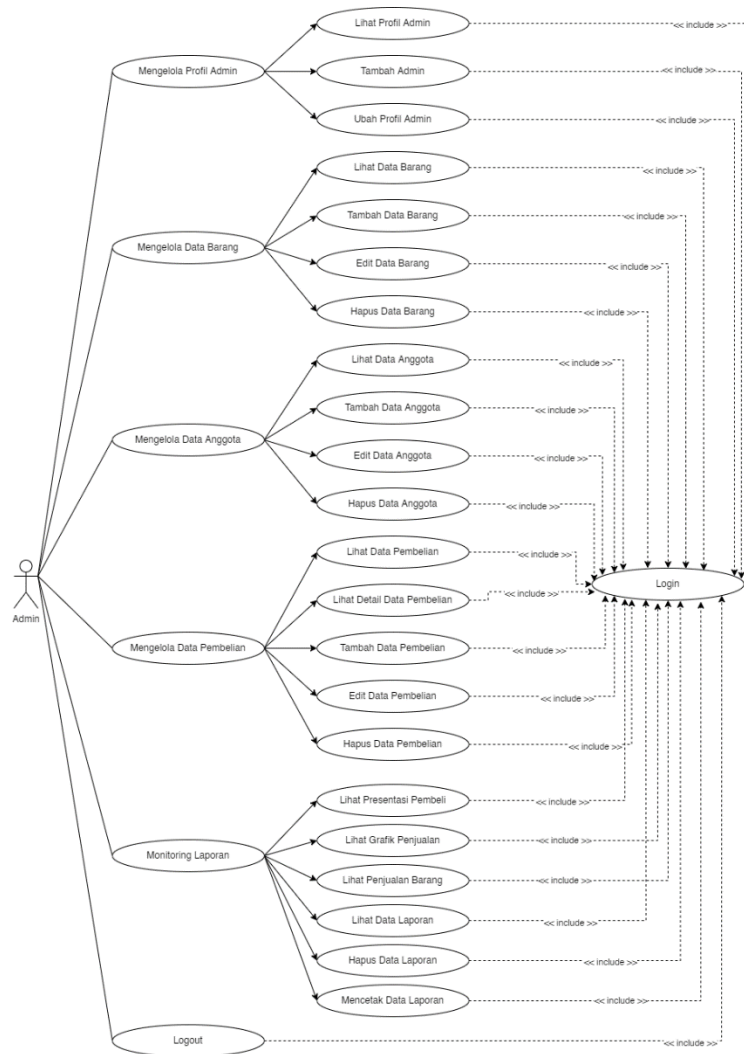
4.1.5 Pemodelan Perencanaan Cepat (*Modelling Quick Design*)

Tahap pemodelan perencanaan cepat dilakukan pembuatan model rancangan sistem yang bertujuan untuk memberikan gambaran kepada klien terkait karakteristik maupun kebutuhan yang akan dibuat ke dalam sistem. Gambaran yang akan dibuat berupa *Use Case Diagram*, perancangan *Class Diagram*, perancangan *database*, perancangan proses, perancangan antarmuka dan perancangan keluaran.

a) *Use Case Diagram*

Use Case Diagram merupakan diagram yang mempresentasikan kebutuhan sistem dari sudut pandang pengguna (Mandolang *et al.* 2017). *Use Case diagram* membedakan antar aktor dalam menggunakan sistem. Interaksi yang terjadi antara satu atau lebih aktor yang terjadi di sistem. Karena aktor pada sistem ini hanya *admin*, jadi hanya *admin* yang dapat berinteraksi dengan sistem.

Mulanya, aktor harus melakukan *login* untuk dapat mengakses fungsi sistem. *Use Case Diagram* pada iterasi pertama terdapat pada Gambar 4.



Gambar 4 *Use Case Diagram* iterasi pertama

Use Case Diagram akan diperjelas alurnya dengan skenario (*Use Case Scenario*). *Use Case Scenario* merupakan penjabaran deskripsi terhadap kebutuhan fungsional yang sesuai dengan sistem (Anwar *et al.* 2014). *Use Case Scenario* menjabarkan dokumentasi bergantung pada format kebutuhannya, diantaranya skenario normal dan skenario alternatif. Skenario normal akan menjelaskan apabila fungsi yang digunakan sukses atau sesuai harapan, sedangkan skenario alternatif akan menjelaskan skenario tidak normal yang akan menampilkan *error*. Salah satu *Use Case Scenario* yaitu *Login* (KPR-001) yang dapat dilihat pada Tabel 5.

Tabel 5 *Use Case Scenario Login*

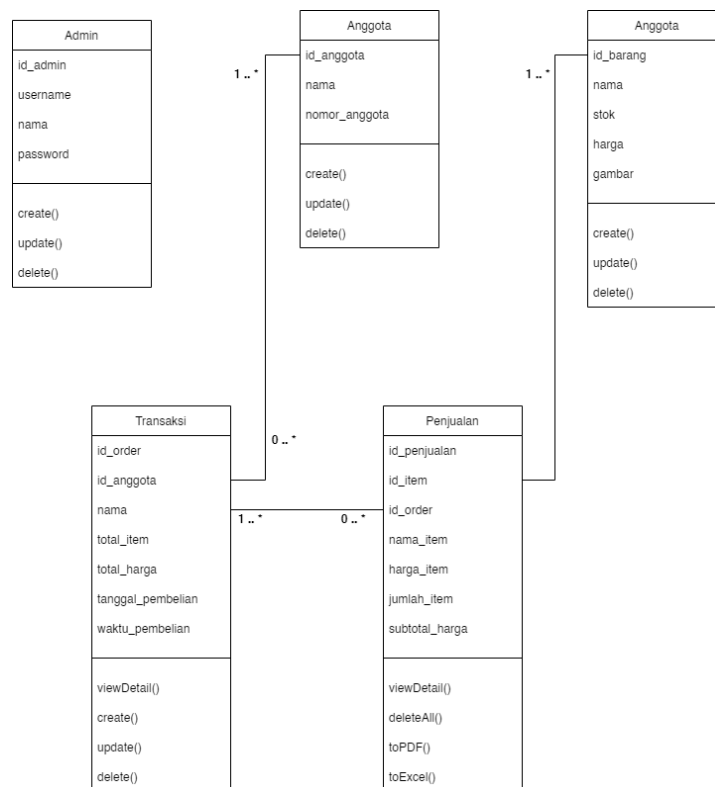
No.	<i>Use Case</i>	Deskripsi
1.	Nama	<i>Login</i>
	Deskripsi	Menampilkan <i>form</i> untuk mengakses sistem.
	Prekondisi	Aktor mengakses halaman <i>login</i> .
	Aliran Utama	1. Membuka halaman <i>login</i> . 2. Isi <i>username</i> dan <i>password</i> . 3. Pilih tombol " <i>Login</i> ".
	Aliran Alternatif	Jika <i>username</i> atau <i>password</i> tidak diisi atau tidak sesuai, maka tidak bisa masuk kedalam sistem.
	Kondisi Akhir	Aplikasi menampilkan halaman utama.

Tabel 6 *Use Case Scenario Iterasi Pertama*

No.	<i>Use Case Scenario</i>	Deskripsi
1.	Menambah Akun <i>Admin</i>	Lampiran 1
2.	Mengubah Akun <i>Admin</i>	Lampiran 2
3.	Menghapus Akun <i>Admin</i>	Lampiran 3
4.	Menambah Data Anggota	Lampiran 4
5.	Mengubah Data Anggota	Lampiran 5
6.	Menghapus Data Anggota	Lampiran 6
7.	Menambah Data Barang	Lampiran 7
8.	Mengubah Data Barang	Lampiran 8
9.	Menghapus Data Barang	Lampiran 9
10.	Menambah Data Pembelian	Lampiran 10
11.	Mengubah Data Pembelian	Lampiran 11
12.	Menghapus Data Pembelian	Lampiran 12
13.	Melihat Data Laporan	Lampiran 13
14.	Mengunduh Laporan	Lampiran 14
15.	Melihat Data Akun <i>Admin</i>	Lampiran 15
16.	Melihat Data Anggota	Lampiran 16
17.	Melihat Data Barang	Lampiran 17
18.	Melihat Data Pembelian	Lampiran 18
19.	Melihat Grafik Penjualan	Lampiran 19
20.	Melihat Data Penjualan Barang Perbulan	Lampiran 20
21.	Melihat Presentase Jenis Pembeli	Lampiran 21
22.	Melihat Detail Data Pembelian	Lampiran 22
23.	<i>Logout</i>	Lampiran 23

b) Perancangan *Class Diagram*

Class Diagram merupakan diagram yang merepresentasikan struktur *class* serta hubungan antar *class* (Syaddad 2017). *Class Diagram* memberikan gambaran struktur sistem berdasarkan kelas-kelas pembentuk sistem yang berisi fungsi-fungsi dari kebutuhan sistem. Tujuan dibuatkan *Class Diagram* yaitu pengembang memiliki acuan dalam membangun sistem dengan kelas-kelas yang sudah dibuat melalui rancangan di dalam *Class Diagram*. Pada iterasi pertama, terdapat 5 kelas yang terdiri dari masing-masing atribut dan fungsinya. *Class Diagram* yang dibuat untuk *website* laporan penjualan koperasi pada iterasi pertama terdapat pada Gambar 5.



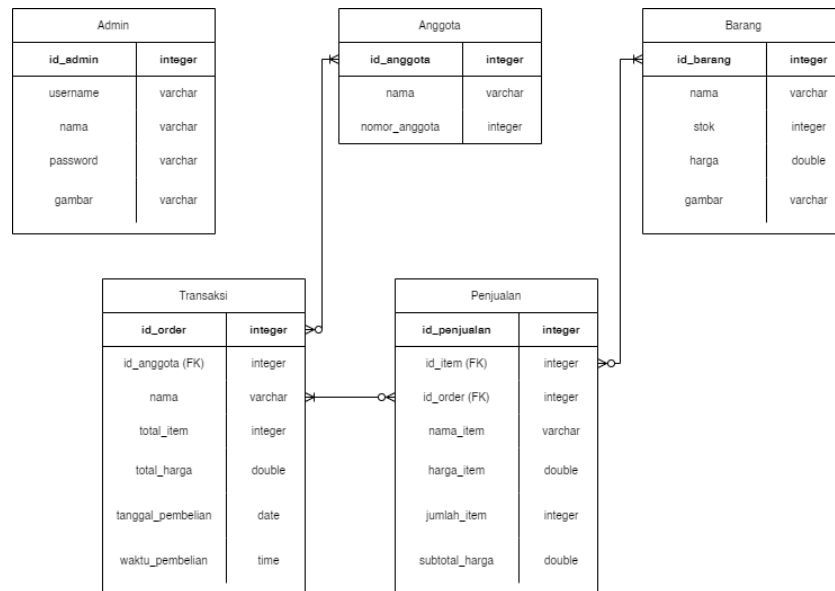
Gambar 5 Perancangan *Class Diagram*

c) Perancangan *Database*

Perancangan *database* menggambarkan struktur *database* yang digunakan dalam sistem. Kumpulan data yang ada di sistem tersimpan kedalam *database* dan berada pada aplikasi *Database Management System* (DBMS). Setiap data yang tersimpan dapat diolah dan dikelola dari DBMS. Sistem basis data yang digunakan menggunakan format *Structured Query Language* (SQL) untuk memanipulasi data yang berada di aplikasi DBMS.

Database yang dibuat bernama “koperasi_db” dan berada pada aplikasi DBMS MySQL. Perancangan yang dibuat memiliki lima tabel. Salah satu tabel pada *database* “koperasi_db” yaitu *tbl_admin* yang berfungsi untuk menyimpan data *admin*. Perancangan *database* dapat dilihat pada Gambar 6. Deskripsi *tbl_admin* berisi kolom *id_admin*, *username*, *nama* dan *password* yang dijelaskan pada Tabel 7. *Tbl_barang* berisi kolom *id_item*, *nama_item*, *harga_item*, *stok_item*, *foto_item*, *kode_item*, *tanggal_tambah* yang dijelaskan

pada lampiran 24. Tbl_anggota berisi kolom id_anggota, nama_anggota, nomor_anggota yang dijelaskan pada lampiran 25. Tbl_pembelian berisi kolom id_pembelian, id_pembeli, nama_pembeli, total_item, total_harga, tanggal_pembelian, penanggung_jawab yang dijelaskan pada lampiran 26. Tbl_transaksi berisi kolom id_jual, id_item, id_order, nama_item, harga_item, jumlah_item, subtotal_item yang dijelaskan pada lampiran 27.



Gambar 6 Perencanaan *Database*

Tabel 7 Deskripsi Tabel *tbl_admin*

No.	Nama <i>Field</i>	Tipe (<i>length</i>)	Null	Keterangan	Deskripsi
1.	<i>id_admin</i>	int (5)	No	<i>Primary Key</i>	Berisi angka unik dengan <i>increment</i> untuk setiap data
2.	<i>username</i>	int (255)	No	-	Berisi nama unik untuk setiap data
3.	<i>nama</i>	varchar (255)	No	-	Berisi nama pengguna yang menggunakan akun
4.	<i>password</i>	varchar (32)	No	-	Berisi kata sandi dengan menggunakan enkripsi sha1

Tabel 8 Deskripsi Tabel Iterasi Pertama

No.	Nama Tabel	Deskripsi
1.	tbl_barang	Lampiran 24
2.	tbl_anggota	Lampiran 25

No.	Nama Tabel	Deskripsi
3.	tbl_pembelian	Lampiran 26
10.	tbl_transaksi	Lampiran 27

d) Perancangan Antarmuka

Perancangan Antarmuka dibuat untuk memberikan gambaran tampilan pada sistem. Tampilan antarmuka ini berdasarkan deskripsi dari *website* laporan penjualan koperasi. Rancangan antarmuka halaman *login* terdapat form untuk masuk kedalam sistem dengan masukan *username* dan *password* yang dapat dilihat pada Gambar 7. Rancangan antarmuka halaman beranda terdapat hasil rekap dan informasi umum mengenai penjualan barang yang dapat dilihat pada Gambar 8. Rancangan antarmuka halaman profil *admin* terdapat tabel informasi dan pengelolaan data *admin* yang dapat dilihat pada Lampiran 28. Rancangan antarmuka halaman data anggota terdapat tabel informasi dan pengelolaan data anggota yang terdaftar yang dapat dilihat pada Lampiran 29. Rancangan antarmuka halaman data barang terdapat tabel informasi dan pengelolaan barang siap jual yang dapat dilihat pada Lampiran 30. Rancangan antarmuka halaman data pembelian terdapat tabel informasi dan pengelolaan transaksi pembelian barang yang dapat dilihat pada Lampiran 31. Rancangan antarmuka halaman laporan terdapat tabel hasil pembelian tiap anggota terdaftar yang dapat dilihat pada Lampiran 32. Tabel lampiran rancangan antarmuka dapat dilihat pada Tabel 9.

Gambar 7 Rancangan Antarmuka Halaman *Login*

Gambar 8 Rancangan Antarmuka Halaman Beranda

Tabel 9 Perancangan Antarmuka

No.	Perancangan Antarmuka	Deskripsi
1.	Halaman Profil <i>Admin</i>	Lampiran 28
2.	Halaman Data Anggota	Lampiran 29
3.	Halaman Data Barang	Lampiran 30
4.	Halaman Data Pembelian	Lampiran 31
5.	Halaman Laporan	Lampiran 32

e) Perancangan Masukan

Perancangan masukan menggambarkan setiap *input*/masukan yang terdapat pada sistem. Perancangan masukan didapatkan berdasarkan fungsi yang akan dibuat pada *website* laporan penjualan koperasi. Rancangan masukan tambah data barang terdapat masukan nama barang, stok, harga dan gambar yang dapat dilihat pada Gambar 9. Rancangan masukan tambah dan ubah akun *admin* terdapat masukan nama, *username*, *email*, *password*, gambar yang dapat dilihat pada Lampiran 33 dan Lampiran 34. Rancangan masukan tambah dan ubah data anggota terdapat masukan nama anggota, nomor anggota yang dapat dilihat pada Lampiran 35 dan Lampiran 36. Rancangan masukan ubah data barang terdapat nama barang, stok, harga dan gambar yang dapat dilihat pada Lampiran 37. Rancangan masukan tambah dan ubah data pembelian terdapat masukan pembeli, barang yang dapat dilihat pada Lampiran 38 dan Lampiran 39. Rancangan masukan *login* terdapat masukan *username* dan *password* yang dapat dilihat pada Lampiran 40. Tabel lampiran perancangan masukan dapat dilihat pada Tabel 10.

Tambah Barang

Gambar 9 Perancangan Masukan Tambah Data Barang

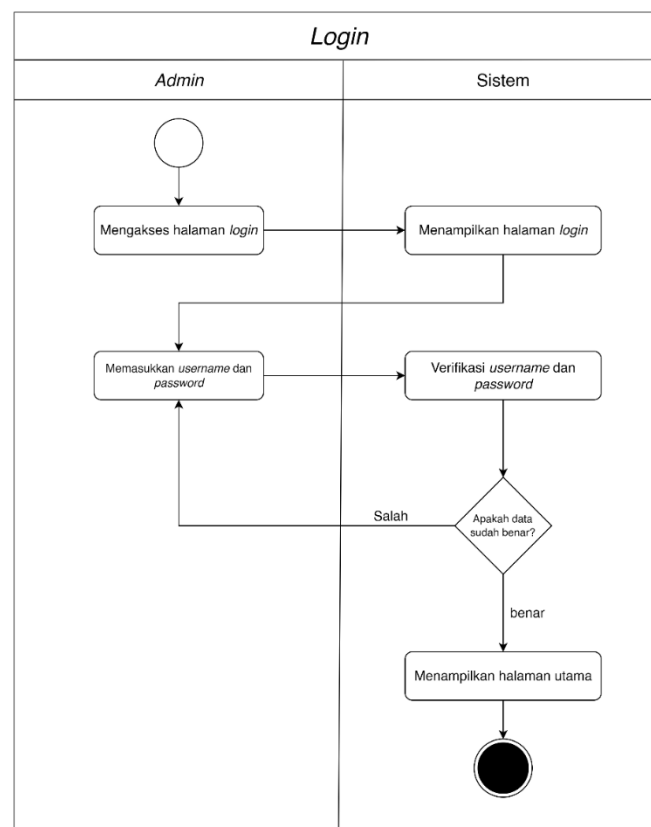
Tabel 10 Perancangan Masukan

No.	Perancangan Masukan	Deskripsi
1.	Tambah Akun <i>Admin</i>	Lampiran 33
2.	Ubah Akun <i>Admin</i>	Lampiran 34
3.	Tambah Data Anggota	Lampiran 35
4.	Ubah Data Anggota	Lampiran 36
5.	Ubah Data Barang	Lampiran 37
6.	Tambah Data Pembelian	Lampiran 38
7.	Ubah Data Pembelian	Lampiran 39

No.	Perancangan Masukan	Deskripsi
8.	<i>Login</i>	Lampiran 40

f) Perancangan Proses

Tahap perancangan proses dibuat menggunakan *activity diagram* yang berfungsi untuk memberikan gambaran alur proses sistem. Setiap *activity diagram* memberikan gambaran untuk setiap fungsi pada sistem. Aktor pada *activity diagram* yaitu *admin* sebagai pengelola sistem. Salah satu *activity diagram* pada sistem yaitu *login* yang dapat dilihat pada Gambar 10. Perancangan proses secara lengkap dapat dilihat pada Tabel 10.



Gambar 10 Activity Diagram Login

Tabel 11 Activity Diagram Iterasi Pertama



No.	Activity Diagram	Deskripsi
1.	Menambah Akun Admin	Lampiran 41
2.	Mengubah Akun Admin	Lampiran 42
3.	Menghapus Akun Admin	Lampiran 43
4.	Menambah Data Anggota	Lampiran 44
5.	Mengubah Data Anggota	Lampiran 45
6.	Menghapus Data Anggota	Lampiran 46
7.	Menambah Data Barang	Lampiran 47
8.	Mengubah Data Barang	Lampiran 48
9.	Menghapus Data Barang	Lampiran 49

No.	<i>Activity Diagram</i>	Deskripsi
10.	Menambah Data Pembelian	Lampiran 50
11.	Mengubah Data Pembelian	Lampiran 51
12.	Menghapus Data Pembelian	Lampiran 52
13.	Melihat Data Laporan	Lampiran 53
14	Mengunduh Data Laporan	Lampiran 54
15.	Melihat Data Akun <i>Admin</i>	Lampiran 55
16.	Melihat Data Anggota	Lampiran 56
17.	Melihat Data Barang	Lampiran 57
18.	Melihat Data Pembelian	Lampiran 58
19.	Melihat Grafik Penjualan	Lampiran 59
20.	Melihat Data Penjualan Barang	Lampiran 60
21.	Melihat Presentase Jenis Pembeli	Lampiran 61
22.	Melihat Detail Data Pembelian	Lampiran 62
23.	<i>Logout</i>	Lampiran 63

g) Perancangan Keluaran

Perancangan keluaran menggambarkan *output*/keluaran dari hasil *input*/masukan yang terdapat pada sistem. Setiap perancangan keluaran menampilkan respon dari masukan yang telah masuk kedalam sistem. Rancangan keluaran *input* barang terdapat tabel informasi barang yang telah dimasukkan kedalam dan siap jual yang dapat dilihat pada Gambar 11. Rancangan keluaran berhasil menambah data terdapat notifikasi data berhasil ditambah oleh sistem yang dapat dilihat pada Lampiran 64. Rancangan keluaran mengubah data terdapat notifikasi data berhasil diubah oleh sistem yang dapat dilihat pada Lampiran 65. Rancangan keluaran konfirmasi menghapus data terdapat peringatan saat melakukan hapus data yang dapat dilihat pada Lampiran 66. Rancangan keluaran berhasil menghapus data terdapat notifikasi data berhasil dihapus oleh sistem yang dapat dilihat pada Lampiran 67. Rancangan keluaran tabel data anggota terdapat tabel informasi anggota yang telah dimasukkan kedalam sistem yang dapat dilihat pada Lampiran 68. Rancangan keluaran tabel data barang terdapat informasi barang yang telah dimasukkan yang dapat dilihat pada Lampiran 69. Rancangan keluaran tabel data penjualan terdapat tabel informasi penjualan yang telah dimasukkan yang dapat dilihat pada Lampiran 70. Rancangan keluaran data laporan terdapat rekap data hasil penjualan tiap anggota yang dapat dilihat pada Lampiran 71. Tabel lampiran rancangan keluaran dapat dilihat pada Tabel 11.

Tabel Barang

Show	10	Entries	search	
Gambar	Nama	Harga	Stok	Aksi
				Ubah Hapus
				Ubah Hapus
Showing 1 to 10 of 2 Entries				previous 1 next

Gambar 11 Perancangan Keluaran Tabel Data Barang

Tabel 12 Perancangan Keluaran Iterasi Pertama

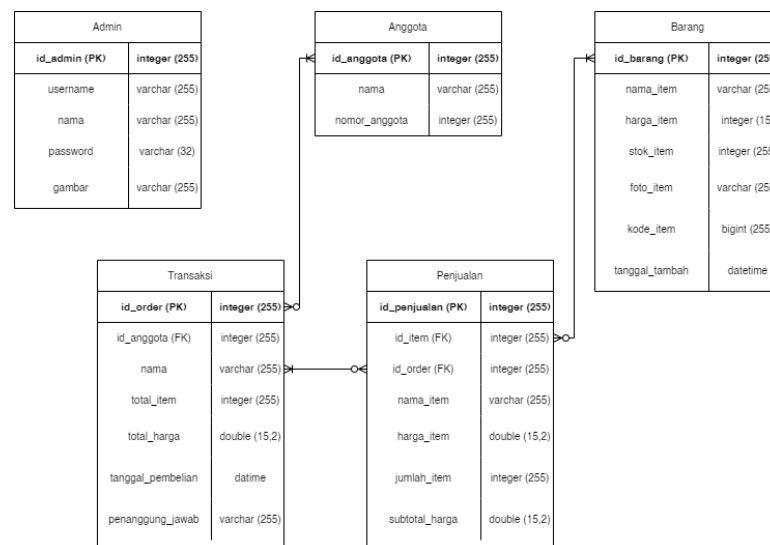
No.	Perancangan Keluaran	Deskripsi
1.	Berhasil Menambah Data	Lampiran 64
2.	Berhasil Mengubah Data	Lampiran 65
3.	Konfirmasi Menghapus Data	Lampiran 66
4.	Berhasil Menghapus Data	Lampiran 67
5.	Tabel Data Akun <i>Admin</i>	Lampiran 68
6.	Tabel Data Anggota	Lampiran 69
7.	Tabel Data Pembelian	Lampiran 70
8.	Tabel Data Laporan	Lampiran 71

4.1.5 Pembentukan *Prototype* (*Contruction of Prototype*)

Tahap pembentukan *prototype* dilakukan implementasi dari tahap sebelumnya yaitu pemodelan yang dibuat menjadi bentuk kode program.

a) Implementasi *Database*

Implementasi *database* dilakukan sesuai perencanaan yang telah dibuat pada tahap pemodelan perencanaan. *Database* sistem menggunakan *Database Management System* (DBMS) MySQL. *Database* pada iterasi pertama terdiri dari lima tabel yaitu *tbl_admin*, *tbl_barang*, *tbl_anggota*, *tbl_pembelian* dan *tbl_transaksi*. Kelima tabel tersebut diimplementasikan kedalam *database* *koperasi_db*. *Database* dibuat melalui phpMyAdmin bersama dengan tabel dan atributnya. Implementasi *Database* dapat dilihat pada Gambar 12.



Gambar 12 Implementasi Database

a) Implementasi Proses

Implementasi proses pada *website* laporan penjualan menggunakan *framework* CodeIgniter dengan menggunakan bahasa pemrograman PHP. CodeIgniter memiliki konsep yang bernama *Model-View-Controller* (MVC). *Model* merupakan komponen yang dapat memanipulasi database. Dalam penggunaannya, *Model* dapat menghubungkan dan mengambil data tabel dari *database*. *View* merupakan komponen yang menampilkan halaman *website*. *View* dapat dipanggil dengan komponen *Controller*. *View* dapat menerima dan merepresentasikan data yang telah dikirim dari *Controller*. *Controller* merupakan pengontrol aplikasi dan penghubung antara *Model* dan *View*. Implementasi proses halaman barang yang akan menampilkan *views* barang pada direktori Item/index yang dapat dilihat pada Gambar 13. Implementasi proses halaman beranda yang akan menampilkan *views home* pada direktori /beranda yang dapat dilihat pada Lampiran 72. Implementasi proses halaman profil *admin* yang akan menampilkan *views admin* pada direktori admin/viewAdmin yang dapat dilihat pada Lampiran 73. Implementasi proses data anggota yang akan menampilkan *views* anggota pada direktori buyer/viewBuyer yang dapat dilihat pada Lampiran 74. Implementasi proses data penjualan yang akan menampilkan *views* penjualan pada direktori order/orderView yang dapat dilihat pada Lampiran 75. Implementasi proses data laporan yang akan menampilkan *views* laporan pada direktori report/index yang dapat dilihat pada Lampiran 76. Tabel lampiran implementasi proses dapat dilihat pada Tabel 12.

```

//Halaman Data Barang
public function index()
{
    $this->m_item = new Model_item();

    $data = [
        'title' => 'Daftar Barang',
        'item' => $this->m_item->getAll(),
        'judul' => 'Koperasi | List Data Item',
    ];
    echo view('header', $data);
    echo view('Item/index', $data);
    echo view('footer');
}

//Fungsi menambah data barang
public function create()
{
    $validationRule = [
        'nama' => 'required',
        'harga' => 'required',
        'gambar' => [
            'mime_in[gambar,image/jpeg,image/gif,image/png]',
            'max_size[gambar,409600]',
        ],
    ];
    if (!$this->validate($validationRule)) {
        $data = ['errors' => implode($this->validator->getErrors())];
        session()->setFlashdata('peringatan', $data['errors']);
        return redirect()->to(base_url('Item'));
    } else {
        if ($this->request->getFile('gambar')->getName() == "") {
            $data = array(
                'nama_item' => $this->request->getPost('nama'),
                'harga_item' => $this->request->getPost('harga'),
                'stok_item' => $this->request->getPost('stok'),
                // 'kode_item' => $this->request->getPost('kode'),
            );
        } elseif (!$this->request->getFile('gambar')->getName() == "") {
            $foto = $this->request->getFile('gambar');
            $foto->move(WRITEPATH . '../public/assets/images/upload/');
            $data = array(
                'nama_item' => $this->request->getPost('nama'),
                'harga_item' => $this->request->getPost('harga'),
                'stok_item' => $this->request->getPost('stok'),
                // 'kode_item' => $this->request->getPost('kode'),
                'foto_item' => $foto->getName(),
            );
        }
        $query = $this->m_item->insert_model($data);
    }
    if ($query) {
        session()->setFlashdata('pesan', 'Data Berhasil ditambah');
        return redirect()->to(base_url('Item'));
    } else {
        echo json_encode(['code' => 0, 'msg' => 'Something went wrong']);
    }
}

```



```

//Halaman ubah data barang
public function viewEdit($id_item)
{
    $data = [
        'title' => 'Edit Barang',
        'key' => $id_item,
        'item' => $this->m_item->getByID($id_item),
    ];
    echo view('header');
    echo view('Item/viewEdit', $data);
    echo view('footer');
}

//Fungsi mengubah data barang
public function edit()
{
    $id = $this->request->getPost('id_item');

    if ($this->validation->setRules(
        [
            'nama_item' => 'required',
            'harga_item' => 'required',
            'gambar' => [
                'mime_in[gambar,image/jpg,image/jpeg,image/gif,image/png]',
                'max_size[gambar,4096]',
            ],
        ]
    )) {
        if ($this->request->getFile('gambar')->getName() == "") {
            $data = array(
                'nama_item' => $this->request->getPost('nama'),
                'harga_item' => $this->request->getPost('harga'),
                'stok_item' => $this->request->getPost('stok'),
            );
        } elseif (!$this->request->getFile('gambar')->getName() == "") {
            $foto = $this->request->getFile('gambar');
            $namabar = str_replace(' ', '-', $foto->getName());
            $foto->move(WRITEPATH . '../public/assets/images/upload/');
            $data = array(
                'nama_item' => $this->request->getPost('nama'),
                'harga_item' => $this->request->getPost('harga'),
                'stok_item' => $this->request->getPost('stok'),
                'foto_item' => $foto->getName(),
            );
        }
    }

    $this->builder_item->update($data, ['id_item' => $id]);
    session()->setFlashdata('pesan', 'Data Berhasil di perbaharui');
    return redirect()->to(base_url('Item'));
}

//Fungsi menghapus data barang
public function delete($id_item)
{
    $this->builder_item = $this->db->table('tbl_item');
    $this->builder_item->delete(['id_item' => $id_item]);
    session()->setFlashdata('pesan', 'Data Berhasil di hapus');
    return redirect()->to(base_url('Item'));
}
}

```

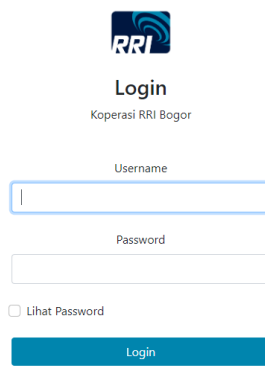
Gambar 13 Implementasi Proses Halaman Barang

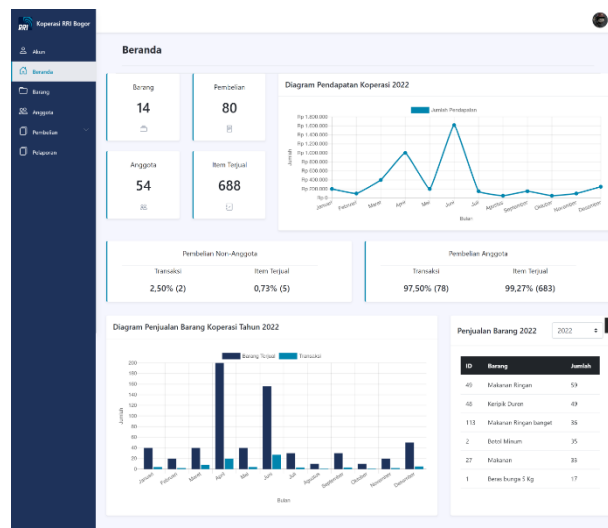
Tabel 13 Implementasi Proses Iterasi Pertama

No.	Implementasi Proses	Deskripsi
1.	Halaman Beranda	Lampiran 72
2.	Halaman Profil <i>Admin</i>	Lampiran 73
3.	Halaman Data Anggota	Lampiran 74
4.	Halaman Data Pembelian	Lampiran 75
5.	Halaman Laporan	Lampiran 76

b) Implementasi Antarmuka

Implementasi antarmuka dilakukan sesuai perencanaan yang telah dibuat pada tahap pemodelan perencanaan. Perencanaan antarmuka akan diimplementasikan kedalam bentuk kode program. Implementasi antarmuka halaman *login* sebagai halaman pertaman untuk masuk sistem yang dapat dilihat pada Gambar 13. Implementasi antarmuka halaman beranda terdapat informasi umum mengenai pembelian barang yang dapat dilihat pada Gambar 14. Implementasi antarmuka halaman profil *admin* terdapat terdapat informasi akun *admin* dan fitur mengelola akun *admin* yang terdapat pada Lampiran 72. Implementasi antarmuka halaman data anggota terdapat informasi data yang terdaftar sebagai anggota koperasi jual beli dan fungsi pengelolaan anggota yang dapat dilihat pada Lampiran 73. Implementasi antarmuka halaman data barang terdapat informasi barang yang tersimpan kedalam sistem untuk dijualbelikan dan fungsi pengelolaan barang yang terdapat pada Lampiran 74. Implementasi antarmuka halaman pembelian terdapat informasi transaksi barang yang terjual oleh pembeli di koperasi dan fungsi pengelolaan transaksi yang dapat dilihat pada Lampiran 75. Implementasi antarmuka halaman laporan terdapat informasi rekap pembelian barang untuk setiap anggota dan penyaringan data rekap yang dapat dilihat pada Lampiran 76. Tabel lampiran implementasi antarmuka dapat dilihat pada Tabel 14.


Gambar 13 Implementasi Antarmuka *Login*



Gambar 14 Implementasi Antarmuka Beranda

Tabel 14 Implementasi Antarmuka Iterasi Pertama

No.	Perancangan Masukan	Deskripsi
1.	Halaman Profil <i>Admin</i>	Lampiran 77
2.	Halaman Data Anggota	Lampiran 78
3.	Halaman Data Barang	Lampiran 79
4.	Halaman Data Pembelian	Lampiran 80
5.	Halaman Laporan	Lampiran 81

c) Implementasi Masukan

Implementasi masukan merupakan hasil dari pemodelan perancangan masukan pada tahap sebelumnya. Implementasi masukan tambah data barang terdapat masukan input barang, nama barang, stok, harga dan gambar yang dapat dilihat pada Gambar 16. Implementasi masukan tambah dan ubah akun *admin* terdapat masukan nama, *username*, *email*, *password* dan gambar yang dapat dilihat pada Lampiran 82 dan Lampiran 83. Implementasi masukan tambah dan ubah data anggota terdapat masukan nama anggota dan nomor anggota yang dapat dilihat pada Lampiran 84 dan Lampiran 85. Implementasi masukan ubah data barang terdapat masukan nama barang, stok, harga dan gambar yang dapat dilihat pada Lampiran 86. Implementasi masukan tambah dan ubah data pembelian terdapat masukan nama pembeli dan barang yang dapat dilihat pada Lampiran 87 dan Lampiran 88. Tabel lampiran implementasi masukan dapat dilihat pada Tabel 14.

Gambar 15 Implementasi Masukan Tambah Data Barang

Tabel 15 Implementasi Masukan Iterasi Pertama

No.	Perancangan Masukan	Deskripsi
1.	Tambah Akun <i>Admin</i>	Lampiran 82
2.	Ubah Profil <i>Admin</i>	Lampiran 83
3.	Tambah Data Anggota	Lampiran 84
4.	Ubah Data Anggota	Lampiran 85
5.	Ubah Data Barang	Lampiran 86
6.	Tambah Data Pembelian	Lampiran 87
7.	Ubah Data Pembelian	Lampiran 88

d) Implementasi Keluaran

Implementasi keluaran merupakan hasil dari pemodelan perancangan keluaran pada tahap sebelumnya. Implementasi keluaran data barang menampilkan keluaran tabel barang terbaru setelah perubahan terakhir yang dapat dilihat pada Gambar 16. Implementasi keluaran berhasil menambah data menampilkan notifikasi bahwa data berhasil ditambahkan oleh sistem yang dapat dilihat pada Lampiran 89. Implementasi keluaran berhasil mengubah data menampilkan notifikasi bahwa data berhasil diubah oleh sistem yang dapat dilihat pada Lampiran 90. Implementasi keluaran konfirmasi menghapus data menampilkan peringatan hapus data setelah menekan tombol hapus yang dapat dilihat pada Lampiran 91. Implementasi keluaran berhasil menghapus data menampilkan notifikasi bahwa data telah dihapus oleh sistem yang dapat dilihat pada Lampiran 92. Implementasi keluaran tabel data akun *admin* menampilkan tabel *admin* terbaru setelah perubahan terakhir yang dapat dilihat pada Lampiran 93. Implementasi keluaran tabel data anggota menampilkan tabel anggota terbaru setelah perubahan terakhir yang dapat dilihat pada Lampiran 94. Implementasi keluaran data pembelian menampilkan tabel pembelian terbaru setelah perubahan terakhir yang dapat dilihat pada Lampiran 95. Implementasi keluaran data laporan menampilkan tabel laporan terbaru setelah perubahan terakhir yang dapat dilihat pada Lampiran 96. Tabel implementasi masukan dapat dilihat pada Tabel 16.

No.	Nama	Gambar	Harga	Stok	Aksi
0	Makanan Ringan		Rp. 15.000,00	56	
1	Keripik Nangka		Rp. 7.500,00	2	
2	Keripik		Rp. 5.000,00	5	
3	Minuman		Rp. 5.000,00	2	
4	Beras		Rp. 50.000,00	12	

Gambar 16 Implementasi Keluaran Tabel Data Barang

Tabel 16 Implementasi Keluaran Iterasi Pertama

No.	Perancangan Keluaran	Deskripsi
1.	Berhasil Menambah Data	Lampiran 89
2.	Berhasil Mengubah Data	Lampiran 90
3.	Konfirmasi Menghapus Data	Lampiran 91
4.	Berhasil Menghapus Data	Lampiran 92
5.	Tabel Data Akun <i>Admin</i>	Lampiran 93
6.	Tabel Data Anggota	Lampiran 94
7.	Tabel Data Pembelian	Lampiran 95
8.	Tabel Data Laporan	Lampiran 96

4.1.5 Penyerahan dan Umpan Balik (*Deployment Delivery and Feedback*)

a) Pengujian

Tahap terakhir yaitu penyerahan sistem yang telah dibuat dan umpan balik dari klien. Sistem diserahkan kepada pembimbing lapang untuk dievaluasi. Pengujian dilakukan dengan presentasi hasil sistem yang dibuat kepada pembimbing lapang. Hasil presentasi diberikan komentar terkait sistem yang dibuat. Umpan balik berupa komentar dijadikan bahan evaluasi untuk melengkapi kebutuhan klien yang masih belum tercapai.

Selanjutnya dilakukan pengujian *black box testing*. Pengujian ini bertujuan untuk memeriksa fungsional sistem saat digunakan serta menemukan kesalahan. Hasil yang didapat dari pengujian *black box testing* pada iterasi pertama, *website* laporan penjualan telah berjalan dengan baik sesuai dengan kebutuhan klien. Hasil pengujian iterasi pertama pada *website* laporan penjualan koperasi dapat dilihat pada Tabel 16.

Tabel 17 Hasil Pengujian *Login*

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
<i>Login</i>	Benar	Mengisi <i>form</i> dengan data yang benar.	Menampilkan halaman utama.	Sukses
	Salah	Mengisi <i>form</i> dengan data yang salah.	1. Menampilkan <i>alert</i> “ <i>username</i> atau <i>password</i> salah”.	Sukses

Tabel 18 Hasil Pengujian Iterasi Pertama

No.	Hasil Pengujian	Deskripsi
1.	Menambah Akun <i>Admin</i>	Lampiran 97
2.	Mengubah Akun <i>Admin</i>	Lampiran 98
3.	Menghapus Akun <i>Admin</i>	Lampiran 99
4.	Menambah Data Anggota	Lampiran 100
5.	Mengubah Data Anggota	Lampiran 101
6.	Menghapus Data Anggota	Lampiran 102
7.	Menambah Data Barang	Lampiran 103
8.	Mengubah Data Barang	Lampiran 104
9.	Menghapus Data Barang	Lampiran 105
10.	Menambah Data Pembelian	Lampiran 106
11.	Mengubah Data Pembelian	Lampiran 107
12.	Menghapus Data Pembelian	Lampiran 108
13.	Melihat Data Laporan	Lampiran 109
14.	Mengunduh Laporan	Lampiran 110
15.	Melihat Data Akun <i>Admin</i>	Lampiran 111
16.	Melihat Data Anggota	Lampiran 112
17.	Melihat Data Barang	Lampiran 113
18.	Melihat Data Pembelian	Lampiran 114
19.	Melihat Grafik Penjualan	Lampiran 115
20.	Melihat Data Penjualan Barang Perbulan	Lampiran 116
21.	Melihat Presentase Jenis Pembeli	Lampiran 117
22.	Melihat Detil Data Pembelian	Lampiran 118
23.	<i>Logout</i>	Lampiran 119

Tahap pengujian telah dilakukan, pembimbing lapang meminta adanya fitur untuk dapat memilih laporan perbulan maupun pertahun. Laporan perbulan berguna untuk melihat data pembelian perbulan untuk mendata pengeluaran dari setiap anggota dalam transaksi. Fitur ini menyaring data sesuai dengan tahun dan bulan. Nantinya pengeluaran tersebut berguna untuk memotong gaji setiap anggota karyawan. Laporan pertahun berguna untuk melihat data pembelian pertahun untuk mendata pengeluaran dari setiap anggota dalam transaksi. Nantinya pengeluaran tersebut berguna untuk menghitung hasil akhir sisa hasil usaha anggota koperasi.

4.2 *Prototype* Iterasi Kedua

Prototype iterasi pertama sudah selesai dilakukan dan telah dievaluasi oleh pembimbing lapang. Setelah dilakukannya pengujian dan evaluasi, terdapat tambahan fitur. Iterasi kedua ini akan menjabarkan lebih lanjut terkait fitur-fitur tambahan yang diberikan. Fitur yang telah dijelaskan pada iterasi pertama tidak dijelaskan kembali pada iterasi kedua.

4.2.1 Komunikasi (*Communication*)

Tahap komunikasi pada iterasi kedua dilakukan diskusi ulang terkait penambahan fitur yang diinginkan klien. Hasil diskusi pada tahap komunikasi iterasi kedua ini didapat kebutuhan tambahan untuk sistem. Dari hasil pengujian dan evaluasi, terdapat penambahan fitur sortir data laporan berdasarkan bulan dan tahun. Sebelumnya pada fitur sortir hanya memberikan fitur sortir data berdasarkan jarak tanggal.

4.2.2 Perencanaan Cepat (*Quick Plan*)

Tahap perencanaan cepat iterasi kedua mengidentifikasi masalah sesuai dengan masalah dan melakukan analisis kebutuhan tambahan yang telah diberikan pada tahap sebelumnya. Identifikasi masalah pada iterasi kedua ini dilakukan untuk beberapa fitur tambahan pada tahap komunikasi berupa pembuatan fitur sortir data laporan perbulan dan pertahun.

a) Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional iterasi kedua dilakukan dari hasil diskusi pada tahap komunikasi. Tambahan beberapa kebutuhan fungsional yang dibuat kedalam sistem dapat dilihat pada Tabel 18.

Tabel 19 Analisis Kebutuhan Fungsional Iterasi Kedua

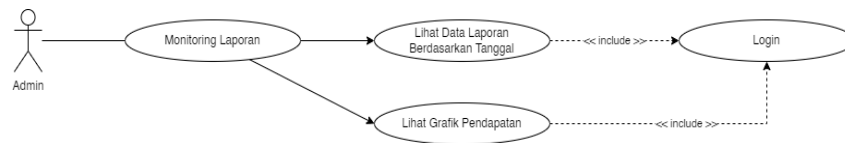
Kode Fungsi	Nama Fungsi	Deskripsi
KPR-023	Melihat Data Laporan Berdasarkan Tanggal	Melihat data laporan dengan memasukkan <i>range</i> tanggal
KPR-024	Melihat Grafik Pendapatan	Melihat grafik pendapatan berdasarkan jumlah harga yang terjual

4.2.3 Pemodelan Perancangan Cepat (*Modelling Quick Design*)

Tahap pemodelan perancangan cepat iterasi kedua terdapat beberapa rancangan meliputi perancangan *use case diagram*, *class diagram*, proses, antarmuka, masukan, keluaran. Perancangan pada iterasi kedua dilakukan hanya untuk penambahan fungsi baru.

a) *Use Case Diagram*

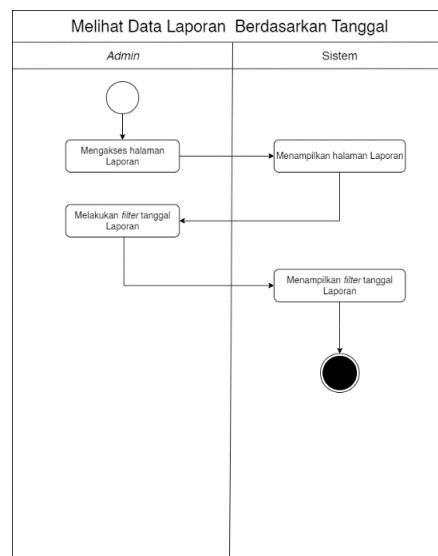
Perancangan *use case diagram* pada tahap iterasi kedua menambah beberapa fungsi. Fungsi tersebut meliputi melihat data laporan penjualan berdasarkan tanggal dan grafik pendapatan. *Use case diagram* iterasi kedua ini hanya menampilkan fungsi tambahan dari kebutuhan fungsional yang dapat dilihat pada Gambar 17.



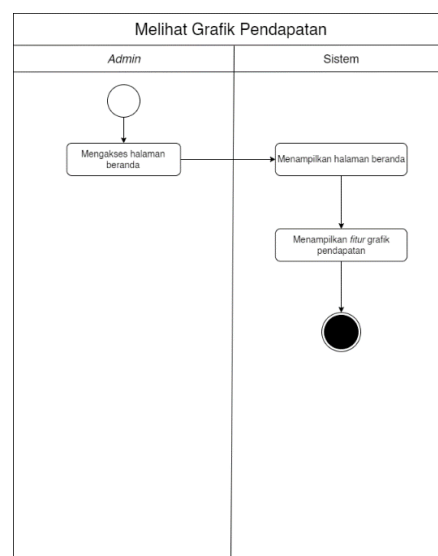
Gambar 17 Use Case Diagram Iterasi Kedua

b) Perancangan Proses

Perancangan proses pada tahap iterasi kedua terdapat fungsi tambahan yang dibuat. Rancangan proses fungsi laporan penjualan berdasarkan tanggal dan grafik pendapatan dapat dilihat pada Gambar 18 dan Gambar 19.



Gambar 18 Perancangan Proses Melihat Data Laporan Berdasarkan Tanggal



Gambar 19 Perancangan Proses Melihat Grafik Pendapatan

c) Perancangan Antarmuka

Tahap perancangan antarmuka iterasi kedua memiliki perubahan dan penambahan pada tampilan. Fungsi data laporan penjualan perbulan dan pertahun akan merubah tampilan pada halaman laporan yang akan menampilkan beberapa tabel. Rancangan antarmuka yang dibuat dapat dilihat pada Gambar 20 dan Gambar 21.

Pembeli	Total transaksi	Total Item	Total Harga	Aksi
				Ubah Hapus
				Ubah Hapus

Gambar 20 Perancangan Antarmuka Melihat Data Laporan Berdasarkan Tanggal

Gambar 21 Perancangan Antarmuka Melihat Grafik Pendapatan

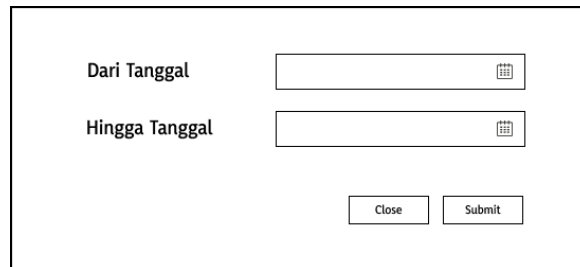
d) Perancangan Masukan

Perancangan masukan pada iterasi kedua terdapat penambahan masukan. Fungsi tambahan yang dibuat akan memiliki masukan dengan memilih bulan atau tahun tertentu untuk menampilkan data tertentu. Rancangan masukan iterasi kedua dapat dilihat pada Gambar 22.

Pembeli	Total transaksi	Total Item	Total Harga	Aksi
				Ubah Hapus
				Ubah Hapus

Gambar 22 Perancangan Masukan Melihat Data Laporan Berdasarkan Tanggal Perancangan Keluaran

Perancangan keluaran pada iterasi kedua memiliki tambahan keluaran. Fungsi tambahan data laporan penjualan perbulan dan pertahun ditampilkan berdasarkan masukan. Hasil yang didapat setelah adanya masukan, halaman akan menampilkan keluaran sesuai dengan data masukan. Rancangan keluaran iterasi kedua dapat dilihat pada Gambar 23.



The image shows a web form for selecting a date range. It contains two labels, 'Dari Tanggal' and 'Hingga Tanggal', each followed by a text input field with a calendar icon on the right. Below these fields are two buttons: 'Close' and 'Submit'.

Gambar 23 Perancangan Keluaran Melihat Data Laporan Berdasarkan Tanggal

4.2.4 Pembentukan *Prototype* (*Construction of Prototype*)

Pembentukan prototype pada tahap iterasi kedua dilakukan implementasi terhadap perancangan model dari tambahan fungsi yang telah sesuai dengan kebutuhan fungsional. Implementasi yang dilakukan tahap iterasi kedua hanya menampilkan tambahan fungsi yang sesuai dengan kebutuhan sistem. Implementasi pada tahap ini terdiri atas implementasi *database*, proses, antarmuka, masukan, dan keluaran.

a) Implementasi Proses

Implementasi proses pada tahap iterasi kedua dilakukan penambahan tabel untuk fitur laporan perbulan dan pertahun. Penambahan tabel tersebut dilakukan memanggil data dari tabel yang sama sehingga tidak memerlukan model baru. Implementasi proses iterasi kedua dapat dilihat pada Gambar 24 dan Gambar 25.

```

//Fungsi menampilkan data pertahun
public function pertahun()
{
    if (!$this->request->getPost('datefrom') or !$this->request->getPost('dateto')) {
        $result = $this->orderModel->findAll();
    } else {
        $datefrom = date("Y-m-d", strtotime($this->request->getPost('datefrom')));
        $dateto = date("Y-m-d", strtotime($this->request->getPost('dateto')));

        $result = $this->orderModel->getfilterdata($datefrom, $dateto);
    }
    if (isset($_GET['month'])) {
        $selected_month = $_GET['month'];
        $result = $this->orderModel->filterByMonth($selected_month);
    }
    foreach ($this->pembeliModel->findAll() as $key => $row) {
        $totalItem = 0;
        $totalHarga = 0;
        $totalTransaksi = 0;
        foreach ($result as $keyorder => $roworder) {
            if ($row['id_pembeli'] == $roworder['id_pembeli']) {
                $totalItem += $roworder['total_item'];
                $totalHarga += $roworder['total_harga'];
                $totalTransaksi++;
            }
        }

        $dataReport[] = array(
            'id_pembeli' => $row['id_pembeli'],
            'nama_pembeli' => $row['nama_pembeli'],
            'total_transaksi' => $totalTransaksi,
            'total_item' => $totalItem,
            'total_harga' => $totalHarga,
        );
    }

    $data = [
        'title' => 'Daftar Hasil Pembelian',
        'dataReport' => $dataReport,
        'orders' => $this->orderModel->findAll(),
        'datefrom' => '',
        'dateto' => '',
    ];
    echo view('header');
    echo view('Report/pertahun', $data);
    echo view('footer');
}

//Fungsi mensortir data berdasarkan tanggal
public function filterdata()
{
    $datefrom = date("Y-m-d", strtotime($this->request->getPost('datefrom')));
    $dateto = date("Y-m-d", strtotime($this->request->getPost('dateto')));

    $result = $this->orderModel->getfilterdata($datefrom, $dateto);

    foreach ($this->pembeliModel->findAll() as $key => $row) {
        $totalItem = 0;
        $totalHarga = 0;
        $totalTransaksi = 0;
        foreach ($result as $keyorder => $roworder) {
            if ($row['id_pembeli'] == $roworder['id_pembeli']) {
                $totalItem += $roworder['total_item'];
            }
        }
    }
}

```

Gambar 24 Implementasi Proses Melihat Data Laporan Berdasarkan Tanggal

```

public function index()
{
    $db = db_connect();
    $itemjualsql = $db->query("SELECT id_order, nama_item, jumlah_item FROM tbl_jual
                                ORDER BY id_jual DESC LIMIT 7 ");
    $itemjualresult = $itemjualsql->getResultArray();
    foreach ($itemjualresult as $key => $val) {
        $arr_jual[] = $val;
    }
    $data = [
        'title' => 'Beranda',
        'admin' => $this->m_admin->findAll(),
        'logged_in' => session()->get(),
        'itemPerMonth' => $this->m_order->itemsPerMonth(),
        'totalperitem' => $this->m_jual->countPerItem(),
        'tot_item' => $this->m_item->get()->getNumRows(),
        'tot_pembeli' => $this->m_pembeli->get()->getNumRows(),
        'tot_pembelian' => $this->m_order->get()->getNumRows(),
        'tot_sold' => $this->m_order->countItemSold(),
        'tot_jual' => $this->m_jual->countItemSold(),
        'itemjual' => $this->m_jual->findAll(),
        'item' => $this->m_item->findAll(),
        'tabel_jual' => $arr_jual,
        'transaksiperbulan' => $this->m_order->transactionPerMonth(),
        'pemasukanperbulan' => $this->m_order->incomePerMonth(),
        'pembelian' => $this->m_order->findAll(),
    ];
    if (isset($_GET['month'])) {
        if (bulan($_GET['month']) == date("Y")) {
            unset($_GET['month']);
            return redirect()->to(base_url('Home'));
        } else {
            $data['totalperitem'] = $this->m_jual->countPerItemByMonths($_GET['month']);
        }
    }
    echo view('header');
    echo view('beranda', $data);
    echo view('footer');
}

```

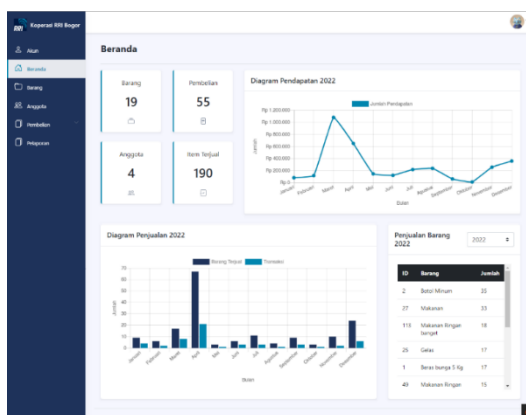
Gambar 25 Implementasi Proses Melihat Grafik Pendapatan

b) Implementasi Antarmuka

Implementasi antarmuka iterasi kedua dilakukan berdasarkan perancangan antarmuka dari penambahan fitur yang telah sesuai dengan tahap sebelumnya. Antarmuka yang diimplementasikan terdapat pada halaman laporan dengan tambahan tabel untuk fungsi baru. Implementasi antarmuka iterasi kedua dapat dilihat pada Gambar 26 dan Gambar 27.

Nama Pembeli	Total Transaksi	Total Item Dibeli	Total Nominal Item
Rahmania	21	68	Rp. 960.000,00
naufal	2	6	Rp. 210.000,00
Ribai	9	38	Rp. 510.000,00
Aqiq	23	77	Rp. 1.690.616,00

Gambar 26 Implementasi Antarmuka Melihat Data Laporan Berdasarkan Tanggal



Gambar 27 Implementasi Antarmuka Melihat Grafik Pendapatan

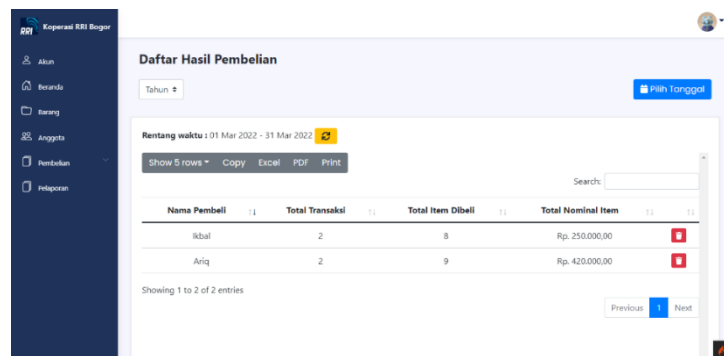
c) Implementasi Masukan

Implementasi masukan tahap iterasi kedua dilakukan berdasarkan dari rancangan antarmuka sebelumnya. Implementasi masukan laporan berdasarkan tanggal dengan memilih bulan dan tahun yang diinginkan. Implementasi masukan iterasi kedua dapat dilihat pada Gambar 28.

Gambar 28 Implementasi Masukan Iterasi Kedua

d) Implementasi Keluaran

Implementasi keluaran tahap iterasi kedua dilakukan sesuai dengan rancangan antarmuka sebelumnya. Implementasi keluaran berhasil mengubah data laporan penjualan perbulan atau pertahun sesuai dengan masukan. Implementasi keluaran iterasi kedua dapat dilihat pada Gambar 29.



Gambar 29 Implementasi Keluaran Iterasi Kedua

4.2.5 Penyerahan dan Umpan Balik (*Deployment Delivery and Feedback*)

Tahap terakhir yaitu sistem telah dibuat lalu sistem dipresentasikan ke pembimbing lapang dengan mensimulasikan sistemnya. Pengujian dilakukan menggunakan metode *black box testing* untuk setiap fungsinya. Hasil pengujian yang dilakukan pada iterasi kedua dapat dilihat pada Tabel 20 dan Tabel 21.

Tabel 20 Hasil Pengujian Melihat Data Laporan Berdasarkan Tanggal

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Melihat Data Laporan Berdasarkan Tanggal	Benar	Mengisi <i>form</i> dengan data yang benar.	Menampilkan data laporan yang sesuai tanggal	Sukses
	Salah	Mengisi <i>form</i> dengan data yang salah.	1. Menampilkan <i>alert</i> "The tanggal awal field is require". 2. Menampilkan <i>alert</i> "The tanggal akhir field is require".	Sukses

Tabel 21 Hasil Pengujian Melihat Grafik Pendapatan

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Melihat Grafik Pendapatan	Benar	Memilih tombol menu beranda	Menampilkan grafik penjualan yang berdasarkan data penjualan	Sukses

V SIMPULAN DAN SARAN

5.1 Simpulan

Setelah melakukan pembuatan dan pengujian aplikasi *Marketing Kit* RRI Bogor dengan menggunakan metodologi *prototyping*, didapatkan beberapa kesimpulan:

- a. *Website* Laporan Penjualan Koperasi RRI Bogor ini telah dapat digunakan untuk merekap hasil penjualan koperasi yaitu melihat data laporan penjualan, mengunduh laporan penjualan, melihat data anggota, melihat data barang, melihat data penjualan, melihat grafik penjualan, melihat data penjualan barang dan melihat presentase pembelian anggota dan non-anggota. Dengan adanya *Website* Laporan Penjualan Koperasi RRI Bogor ini, pengguna dapat melihat hasil rekap penjualan koperasi.
- b. *Website* Laporan Penjualan Koperasi RRI Bogor telah dapat digunakan untuk mengelola barang dan penjualan koperasi yaitu menambah akun admin, mengubah akun admin, menghapus akun admin, menambah data anggota, mengubah data anggota, menghapus data anggota, menambah data barang, mengubah data barang, menghapus data barang, menambah data penjualan, mengubah data penjualan, menghapus data penjualan. Dengan adanya *Website* Laporan Penjualan Koperasi RRI Bogor ini, pengguna dapat mengelola barang dan penjualan dengan baik.

5.2 Saran

Website laporan penjualan koperasi RRI Bogor dapat dikembangkan menjadi lebih baik lagi, seperti menambahkan fitur pengelolaan stok barang, kategori barang, dan menambah data keranjang tanpa *load* ulang halaman.

DAFTAR PUSTAKA

- Aditya R, Pranatawijaya VH, Putra PB. 2021. Rancang Bangun Aplikasi Monitoring Kegiatan Menggunakan Metode Prototype. *JOINTECOMS*. 7(2):49–50.
- Anwar B, Jaya H, Kusuma IP. 2014. IMPLEMENTASI LOCATION BASED SERVICE BERBASIS ANDROID UNTUK MENGETAHUI POSISI USER. *Jurnal SAINTIKOM*. 13(2):127–130.
- Febrianto E, Yulianto, Lestari NHF. 2018. Penerapan Viewboard Rooster Berbasis Bootstrap Sebagai Penunjang Pelayanan iDuhelp! Pada Perguruan Tinggi. *Jurnal Ilmiah Teknologi Informasi Asia*. 12(2):125–126.
- Firmansyah Y, Maulana R, Arivianti D. 2019. PROTOTIPE SISTEM INFORMASI PELELANGAN BARANG BERBASIS WEB SEBAGAI MEDIA PENGOLAH INFORMASI DATA PELELANGAN. *Jurnal Khatulistiwa Informatika*. 7(2):135–136.
- Hutahaean J. 2014. *Konsep Sistem Informasi*. Ed ke-1. Volume ke-1. Yogyakarta: Deepublish.
- Ishari QA, Wibowo AT, Milad M K. 2020. Sistem Informasi Aset Intelektual Knowledge Management System. *MATICS*. 12(1):16.
- Mandolang A, Deo YY, Rindengan, Steven R, Sentinuwo. 2017. Rancang Bangun Aplikasi Agenda Kegiatan Pimpinan (Studi Kasus: Rektorat Unsrat). *E-Journal Teknik Informatika*. 11(1):1–5.
- Manumutu MA, Wattimena J. 2019. Perancangan Sistem Informasi Konsultasi Akademik Berbasis Website. *Jurnal Sistem Informasi Bisnis*. *Jurnal Sistem Informasi Bisnis*. 9(2):151–154.
- Prasetyo E. 2015. Rancang Bangun Sistem Informasi Sekolah Tinggi Ilmu Ekonomi Rahmanyah Kabupaten Musi Banyuasin Berbasis Website. *Jurnal Informatika*. 1(2):19–20.
- Prinhandoyo MT. 2018. Unified Modeling Language (UML) Model Untuk Pengembangan Sistem Informasi Akademik Berbasis Web. *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*. 3(1):127–128.
- Puspitasari D. 2015. Rancang Bangun Sistem Informasi Koperasi Simpan Pinjam Karyawan Berbasis Web. *Jurnal Pilar Nusa Mandiri*. 11(2):187.
- Putra AD, Andriyanto F, Karisman, Harti TD, Puspita W. 2020. PENGUJIAN APLIKASI POINT OF SALES BERBASIS WEB MENGGUNAKAN BLACK BOX TESTING. *Jurnal Bina Komputer*. 2(1):75.
- Sianipar RH. 2015. Pemrograman Javascript: Teori Dan Implementasi. Ed ke-1 hlm 1–2.
- Syaddad H. 2017. RANCANG BANGUN DIGITAL ARCHIVING DI PERGURUAN TINGGI MENGGUNAKAN METODE PROTOTYPE MODEL STUDI KASUS: UNIVERSITAS SURYAKANCANA. *Media Jurnal Informatika*. 9(1):49–51.
- Usada E, Yuniarsyah Y, Rifani N. 2012. RANCANG BANGUN SISTEM INFORMASI JADWAL PERKULIAHAN BERBASIS JQUERY MOBILE DENGAN MENGGUNAKAN PHP DAN MySQL. *Jurnal Infotel*. 4(2):40–41.

LAMPIRAN

Lampiran 1 *Use Case Scenario* Menambah Akun

No.	Use Case	Deskripsi
1.	Nama Deskripsi Prekondisi Aliran Utama Aliran Alternatif Kondisi Akhir	Menambah Akun Menambah akun admin. Aktor mengakses halaman utama. 1. Membuka halaman akun. 2. Klik tombol tambah. 3. Isi data untuk menambah akun. 4. Pilih tombol <i>submit</i> . Jika nama, <i>username</i> , <i>email</i> , <i>password</i> tidak diisi, maka tidak dapat menambah akun. Aplikasi menyimpan data masukan dan menampilkan halaman akun dengan status “Data berhasil ditambah”.

Lampiran 2 *Use Case Scenario* Mengubah Akun

No.	Use Case	Deskripsi
1.	Nama Deskripsi Prekondisi Aliran Utama Aliran Alternatif Kondisi Akhir	Mengubah Akun Mengubah akun admin. Aktor mengakses halaman utama. 1. Membuka halaman akun. 2. Klik tombol <i>edit</i> . 3. Isi data yang ingin diubah. 4. Pilih tombol simpan. Jika nama, <i>email</i> tidak diisi, maka tidak dapat menyimpan perubahan. Aplikasi mengubah perubahan dan menampilkan halaman akun dengan status “Data berhasil diubah”.

Lampiran 3 *Use Case Scenario* Menghapus Akun

No.	Use Case	Deskripsi
1.	Nama Deskripsi Prekondisi Aliran Utama	Menghapus Akun Menghapus akun admin. Aktor mengakses halaman utama. 1. Membuka halaman akun.

No.	Use Case	Deskripsi
		2. Klik tombol hapus. 3. Klik OK pada konfirmasi aksi.
	Aliran Alternatif	Jika mengklik <i>cancel</i> pada konfirmasi aksi maka hapus akun dibatalkan.
	Kondisi Akhir	Aplikasi menghapus data akun dan menampilkan halaman akun dengan status “Data berhasil dihapus”.

Lampiran 4 *Use Case Scenario* Menambah Data Anggota

No.	Use Case	Deskripsi
1.	Nama	Menambah Data Anggota
	Deskripsi	Menambah data anggota koperasi.
	Prekondisi	Aktor mengakses halaman utama.
	Aliran Utama	1. Membuka halaman anggota. 2. Isi nama anggota, nomor anggota. 3. Klik tombol <i>input data</i> .
	Aliran Alternatif	Jika nama anggota, nomor anggota tidak diisi, maka tidak dapat menyimpan data.
	Kondisi Akhir	Aplikasi menyimpan data anggota dan menampilkan halaman anggota dengan status “Data berhasil ditambahkan”.

Lampiran 5 *Use Case Scenario* Mengubah Data Anggota

No.	Use Case	Deskripsi
1.	Nama	Mengubah Data Anggota
	Deskripsi	Mengubah data anggota koperasi.
	Prekondisi	Aktor mengakses halaman utama.
	Aliran Utama	1. Membuka halaman anggota. 2. Klik tombol <i>edit</i> . 3. Isi data yang ingin diubah. 4. Pilih tombol simpan.

No.	Use Case	Deskripsi
	Aliran Alternatif	Jika nama anggota, nomor anggota tidak diisi, maka tidak dapat mengubah data.
	Kondisi Akhir	Aplikasi mengubah data anggota dan menampilkan halaman anggota dengan status “Data berhasil diubah”.

Lampiran 6 Use Case Scenario Menghapus Data Anggota

No.	Use Case	Deskripsi
1.	Nama	Menghapus Data Anggota
	Deskripsi	Menghapus data anggota koperasi.
	Prekondisi	Aktor mengakses halaman utama.
	Aliran Utama	<ol style="list-style-type: none"> 1. Membuka halaman anggota. 2. Klik tombol hapus. 3. Klik OK pada konfirmasi aksi.
	Aliran Alternatif	Jika mengklik <i>cancel</i> pada konfirmasi aksi maka hapus anggota dibatalkan.
	Kondisi Akhir	Aplikasi menghapus data anggota dan menampilkan halaman anggota dengan status “Data berhasil dihapus”.

Lampiran 7 Use Case Scenario Menambah Data Barang

No.	Use Case	Deskripsi
1.	Nama	Menambah Data Barang
	Deskripsi	Menambah data barang koperasi.
	Prekondisi	Aktor mengakses halaman utama.
	Aliran Utama	<ol style="list-style-type: none"> 1. Membuka halaman barang. 2. Klik tombol <i>input data</i>. 3. Isi nama barang, kode barang, stok, harga, gambar. 4. Klik tombol <i>submit</i>.

No.	Use Case	Deskripsi
	Aliran Alternatif	Jika nama barang, stok, harga tidak diisi, maka tidak dapat menyimpan data.
	Kondisi Akhir	Aplikasi menyimpan data anggota dan menampilkan halaman barang dengan status “Data berhasil ditambahkan”.

Lampiran 8 Use Case Scenario Mengubah Data Barang

No.	Use Case	Deskripsi
1.	Nama	Mengubah Data Barang
	Deskripsi	Mengubah data barang koperasi.
	Prekondisi	Aktor mengakses halaman utama.
	Aliran Utama	1. Membuka halaman barang. 2. Klik tombol <i>edit</i> . 3. Isi data yang ingin diubah. 4. Pilih tombol simpan.
	Aliran Alternatif	Jika nama barang, harga tidak diisi, maka tidak dapat mengubah data.
	Kondisi Akhir	Aplikasi mengubah data anggota dan menampilkan halaman barang dengan status “Data berhasil diubah”.

Lampiran 9 Use Case Scenario Menghapus Data Barang

No.	Use Case	Deskripsi
1.	Nama	Menghapus Data Barang
	Deskripsi	Menghapus data barang koperasi.
	Prekondisi	Aktor mengakses halaman utama.
	Aliran Utama	1. Membuka halaman anggota. 2. Klik tombol hapus. 3. Klik OK pada konfirmasi aksi.
	Aliran Alternatif	Jika mengklik <i>cancel</i> pada konfirmasi aksi maka hapus barang dibatalkan.

No.	Use Case	Deskripsi
	Kondisi Akhir	Aplikasi menghapus data anggota dan menampilkan halaman anggota dengan status “Data berhasil dihapus”.

Lampiran 10 Use Case Scenario Menambah Data Pembelian

No.	Use Case	Deskripsi
1.	Nama	Menambah Data Pembelian
	Deskripsi	Menambah data pembelian koperasi.
	Prekondisi	Aktor mengakses halaman utama.
	Aliran Utama	<ol style="list-style-type: none"> 1. Membuka halaman <i>input</i> pembelian. 2. Isi nama anggota, barang. 3. Klik tombol tambah. 4. Klik tombol simpan.
	Aliran Alternatif	Jika nama anggota, barang kosong atau tidak diisi, maka tidak dapat menyimpan data.
	Kondisi Akhir	Aplikasi menyimpan data pembelian dan menampilkan halaman <i>input</i> pembelian dengan status “Data berhasil ditambahkan”.

Lampiran 11 Use Case Scenario Mengubah Data Pembelian

No.	Use Case	Deskripsi
1.	Nama	Mengubah Data Pembelian
	Deskripsi	Mengubah data pembelian koperasi.
	Prekondisi	Aktor mengakses halaman utama.
	Aliran Utama	<ol style="list-style-type: none"> 1. Membuka halaman daftar pembelian. 2. Klik tombol <i>edit</i>. 3. Isi data yang ingin diubah. 4. Pilih tombol simpan.
	Aliran Alternatif	Jika nama anggota, barang kosong atau tidak diisi, maka tidak dapat mengubah data.
	Kondisi Akhir	Aplikasi mengubah data pembelian dan menampilkan halaman daftar pembelian

No.	Use Case	Deskripsi
		dengan status “Data berhasil diubah”.

Lampiran 12 *Use Case Scenario* Menghapus Data Pembelian

No.	Use Case	Deskripsi
1.	Nama Deskripsi Prekondisi Aliran Utama Aliran Alternatif Kondisi Akhir	Menghapus Data Pembelian Menghapus data pembelian koperasi. Aktor mengakses halaman utama. 1. Membuka halaman daftar pembelian. 2. Klik tombol hapus. 3. Klik OK pada konfirmasi aksi. Jika mengklik <i>cancel</i> pada konfirmasi aksi maka hapus pembelian dibatalkan. Aplikasi menghapus data pembelian dan menampilkan halaman anggota dengan status “Data berhasil dihapus”.

Lampiran 133 *Use Case Scenario* Melihat Data Laporan

No.	Use Case	Deskripsi
1.	Nama Deskripsi Prekondisi Aliran Utama Aliran Alternatif Kondisi Akhir	Melihat Data Laporan Melihat data laporan koperasi. Aktor mengakses halaman utama. 1. Membuka halaman pelaporan. - Aplikasi menampilkan laporan koperasi.

Lampiran 144 *Use Case Scenario* Mengunduh Data Laporan

No.	Use Case	Deskripsi
1.	Nama Deskripsi Prekondisi	Mengunduh Data Laporan Mengunduh data laporan penjualan. Aktor mengakses halaman utama.

No.	Use Case	Deskripsi
	Aliran Utama	1. Membuka halaman pelaporan. 2. Klik tombol <i>excel</i> atau PDF.
	Aliran Alternatif Kondisi Akhir	- Aplikasi mengunduh data laporan dan otomatis tersimpan kedalam <i>device</i> .

Lampiran 155 *Use Case Scenario* Melihat Data Akun Admin

No.	Use Case	Deskripsi
1.	Nama Deskripsi Prekondisi	Melihat Data Akun Admin Melihat data akun <i>admin</i> . Aktor mengakses halaman utama.
	Aliran Utama	1. Membuka halaman akun.
	Aliran Alternatif Kondisi Akhir	- Aplikasi menampilkan akun admin.

Lampiran 166 *Use Case Scenario* Melihat Data Anggota

No.	Use Case	Deskripsi
1.	Nama Deskripsi Prekondisi	Melihat Data Anggota Melihat data anggota koperasi. Aktor mengakses halaman utama.
	Aliran Utama	1. Membuka halaman anggota.
	Aliran Alternatif Kondisi Akhir	- Aplikasi menampilkan anggota koperasi.

Lampiran 177 *Use Case Scenario* Melihat Data Barang

No.	Use Case	Deskripsi
1.	Nama Deskripsi Prekondisi	Melihat Data Barang Melihat data barang koperasi. Aktor mengakses halaman utama.
	Aliran Utama	1. Membuka halaman barang.
	Aliran Alternatif	-

No.	Use Case	Deskripsi
	Kondisi Akhir	Aplikasi menampilkan barang koperasi.

Lampiran 188 *Use Case Scenario* Melihat Data Pembelian

No.	Use Case	Deskripsi
1.	Nama Deskripsi Prekondisi Aliran Utama Aliran Alternatif Kondisi Akhir	Melihat Data Pembelian Melihat data pembelian koperasi. Aktor mengakses halaman utama. 1. Membuka halaman daftar pembelian. - Aplikasi menampilkan daftar pembelian koperasi.

Lampiran 199 *Use Case Scenario* Melihat Grafik Penjualan

No.	Use Case	Deskripsi
1.	Nama Deskripsi Prekondisi Aliran Utama Aliran Alternatif Kondisi Akhir	Melihat Grafik Penjualan Melihat grafik penjualan koperasi. Aktor melakukan <i>login</i> . 1. Membuka halaman utama. Jika mengklik <i>cancel</i> pada konfirmasi aksi maka <i>logout</i> dibatalkan. Aplikasi menampilkan grafik penjualan koperasi.

Lampiran 200 *Use Case Scenario* Melihat Penjualan Barang Perbulan

No.	Use Case	Deskripsi
1.	Nama Deskripsi Prekondisi Aliran Utama Aliran Alternatif Kondisi Akhir	Melihat Penjualan Barang Perbulan Melihat tabel penjualan barang koperasi perbulan. Aktor melakukan <i>login</i> . 1. Membuka halaman utama. 2. Memilih data berdasarkan bulan. - Aplikasi menampilkan tabel penjualan barang koperasi.

Lampiran 211 *Use Case Scenario* Melihat Presentase Jenis Pembeli

No.	Use Case	Deskripsi
1.	Nama	Melihat Presentase Jenis Pembeli
	Deskripsi	Melihat presentase penjualan barang anggota dan non-anggota.
	Prekondisi	Aktor melakukan <i>login</i> .
	Aliran Utama	1. Membuka halaman utama.
	Aliran Alternatif	-
	Kondisi Akhir	Aplikasi menampilkan presentase jenis pembeli.

Lampiran 222 *Use Case Scenario* Melihat *Detail* Data Pembelian

No.	Use Case	Deskripsi
1.	Nama	Melihat Detail Data Pembelian
	Deskripsi	Melihat detail data pembelian koperasi.
	Prekondisi	Aktor mengakses halaman utama.
	Aliran Utama	1. Membuka halaman pembelian 2. Klik tombol <i>detail</i> .
	Aliran Alternatif	-
	Kondisi Akhir	Aplikasi menampilkan detail data pembelian koperasi.

Lampiran 233 *Use Case Scenario* Logout

No.	Use Case	Deskripsi
1.	Nama	<i>Logout</i>
	Deskripsi	Keluar sebagai akun dari aplikasi.
	Prekondisi	Aktor melakukan <i>login</i> .
	Aliran Utama	1. Membuka halaman akun. 2. Klik tombol <i>logout</i> . 3. Klik OK pada konfirmasi aksi.
	Aliran Alternatif	Jika mengklik <i>cancel</i> pada konfirmasi aksi maka hapus pembelian dibatalkan.

No.	<i>Use Case</i>	Deskripsi
	Kondisi Akhir	Aplikasi melakukan <i>logout</i> dan menampilkan halaman <i>login</i> .

Lampiran 244 Deskripsi Tabel tbl_barang

No.	Nama <i>Field</i>	Tipe (<i>length</i>)	Null	Keterangan	Deskripsi
1.	id_item	int (255)	No	<i>Primary Key</i>	Berisi angka unik dengan <i>increment</i> untuk setiap data
2.	nama_item	varchar (255)	No	-	Berisi nama untuk setiap data barang
3.	harga_item	int (15)	No	-	Berisi harga untuk setiap data barang
4.	stok_item	int (255)	No	-	Berisi stok untuk setiap data barang
5.	foto_item	varchar (255)	Yes	-	Berisi foto untuk setiap data barang
6.	kode_item	bigint (255)	Yes	-	Berisi kode barang untuk setiap data barang
7.	tanggal_tambah	datetime	Yes	-	Berisi tanggal <i>input</i> barang

Lampiran 255 Deskripsi Tabel tbl_anggota

No.	Nama <i>Field</i>	Tipe (<i>length</i>)	Null	Keterangan	Deskripsi
1.	id_pembeli	int (255)	No	<i>Primary Key</i>	Berisi angka unik dengan <i>increment</i> untuk setiap data
2.	nama_pembeli	varchar (255)	No	-	Berisi nama anggota untuk setiap data anggota
3.	nomor_anggota	varchar (255)	No	-	Berisi kode anggota untuk setiap data anggota

Lampiran 266 Deskripsi Tabel tbl_pembelian




No.	Nama <i>Field</i>	Tipe (<i>length</i>)	Null	Keterangan	Deskripsi
1.	id_pembelian	int (255)	No	Primary Key	Berisi angka unik dengan <i>increment</i> untuk setiap data
2.	id_pembeli	int (255)	No	Foreign Key	Berisi id pembeli unik yang terdapat pada tabel tbl_anggota
3.	nama_pembeli	varchar (255)	No	-	Berisi nama anggota yang sesuai dengan id_pembeli
4.	total_item	int (255)	No	-	Berisi jumlah barang yang dibeli untuk setiap data pembelian
5.	total_harga	double (15,2)	No	-	Berisi total harga dari barang yang dibeli untuk setiap data pembelian
6.	tanggal_pembelian	datetime	Yes	-	Berisi tanggal pembelian yang terjadi untuk setiap data pembelian
8.	penanggungjawab	varchar (255)	No	-	Berisi nama akun admin yang bertanggung jawab untuk setiap data pembelian

Lampiran 277 Deskripsi Tabel tbl_transaksi

No.	Nama <i>Field</i>	Tipe (<i>length</i>)	Null	Keterangan	Deskripsi
1.	id_jual	int (255)	No	Primary Key	Berisi angka unik dengan <i>increment</i> untuk setiap data
2.	id_item	int (255)	No	-	Berisi id barang unik yang terdapat pada tabel tbl_barang

No.	Nama <i>Field</i>	Tipe (<i>length</i>)	Null	Keterangan	Deskripsi
3.	id_order	int (255)	No	-	Berisi id pembelian unik yang terdapat pada tabel tbl_pembelian
4.	nama_item	varchar (255)	No	-	Berisi nama barang yang dibeli untuk setiap data transaksi
5.	harga_item	double (15,2)	No	-	Berisi harga barang yang dibeli untuk setiap data transaksi
6.	jumlah_item	int (255)	No	-	Berisi jumlah barang yang dibeli untuk setiap data transaksi
7.	subtotal_item	double (15,2)	No	-	Berisi total harga yang dibeli dari perhitungan harga dan jumlah barang

Lampiran 288 Perancangan Antarmuka Halaman Profil *Admin*

Menu Sidebar	Header				
	<div>  <div>Logout Edit</div> </div>				
	Gambar	Nama	Username	Password	Aksi
					
					

Lampiran 299 Perancangan Antarmuka Halaman Data Anggota

Menu Sidebar	Header				
	Anggota				
	No	Nama	Kode Anggota	Aksi	
				<div>Ubah Hapus</div>	
			<div>Ubah Hapus</div>		

Nama

Kode Anggota

Input

Lampiran 300 Perancangan Antarmuka Halaman Data Barang

Header																
Menu Sidebar	<p>Barang</p> <p>Input</p> <table border="1"> <thead> <tr> <th>Gambar</th> <th>Nama</th> <th>Harga</th> <th>Stok</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td>Detail Ubah Hapus</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td>Detail Ubah Hapus</td> </tr> </tbody> </table>	Gambar	Nama	Harga	Stok	Aksi					Detail Ubah Hapus					Detail Ubah Hapus
	Gambar	Nama	Harga	Stok	Aksi											
					Detail Ubah Hapus											
					Detail Ubah Hapus											

Lampiran 311 Perancangan Antarmuka Halaman Data Pembelian

Header																
Menu Sidebar	<p>Pembelian</p> <p>Input</p> <table border="1"> <thead> <tr> <th>Pembeli</th> <th>Jumlah</th> <th>Harga</th> <th>Tanggal</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td>Detail Ubah Hapus</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td>Detail Ubah Hapus</td> </tr> </tbody> </table>	Pembeli	Jumlah	Harga	Tanggal	Aksi					Detail Ubah Hapus					Detail Ubah Hapus
	Pembeli	Jumlah	Harga	Tanggal	Aksi											
					Detail Ubah Hapus											
					Detail Ubah Hapus											

Lampiran 322 Perancangan Antarmuka Halaman Laporan

Header																
Menu Sidebar	<p>Laporan tahun</p> <p>PDF Excel</p> <table border="1"> <thead> <tr> <th>Pembeli</th> <th>Total transaksi</th> <th>Total Item</th> <th>Total Harga</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td>Ubah Hapus</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td>Ubah Hapus</td> </tr> </tbody> </table>	Pembeli	Total transaksi	Total Item	Total Harga	Aksi					Ubah Hapus					Ubah Hapus
	Pembeli	Total transaksi	Total Item	Total Harga	Aksi											
					Ubah Hapus											
					Ubah Hapus											

Lampiran 333 Perancangan Masukkan Tambah Akun *Admin*

Header	
Menu Sidebar	<p>Tambah Akun</p> <div> <p>Username <input type="text"/></p> <p>Nama <input type="text"/></p> <p>Password <input type="password"/></p> <p>Gambar <input type="text"/></p> <p>Input</p> <p>Preview Gambar</p> <p>Simpan</p> </div>

Lampiran 344 Perancangan Masukkan Ubah Akun *Admin*

Menu Sidebar

Header

Ubah Profil

Preview Gambar

Nama

Input

Password

Input

Gambar

Input

Simpan

Lampiran 355 Perancangan Masukkan Tambah Data Anggota

Menu Sidebar

Header

Tambah Anggota

Nama

Input

Nomor Anggota

Input

Simpan

Lampiran 366 Perancangan Masukkan Ubah Data Anggota

Nama

Input

Nomor Anggota

Input

Simpan

Lampiran 377 Perancangan Masukkan Ubah Data Barang

Ubah Barang

Nama Barang

Makanan

Stok

15

Harga

50000

Gambar

Input

Preview Gambar

Simpan

Lampiran 388 Perancangan Masukkan Tambah Data Pembelian

Tambah Pembelian

Jenis Pembeli

Nama Pembeli

Cari Barang

Keranjang

Nama	Jumlah	harga	Aksi
makanan	5	50000	<div>Hapus</div>

Update

Hapus

Simpan

Lampiran 399 Perancangan Masukkan Ubah Data Pembelian

Ubah Pembelian

Jenis Pembeli

Nama Pembeli

Cari Barang

Keranjang

Nama	Jumlah	harga	Aksi
makanan	5	50000	<div>Hapus</div>

Update

Hapus

Simpan

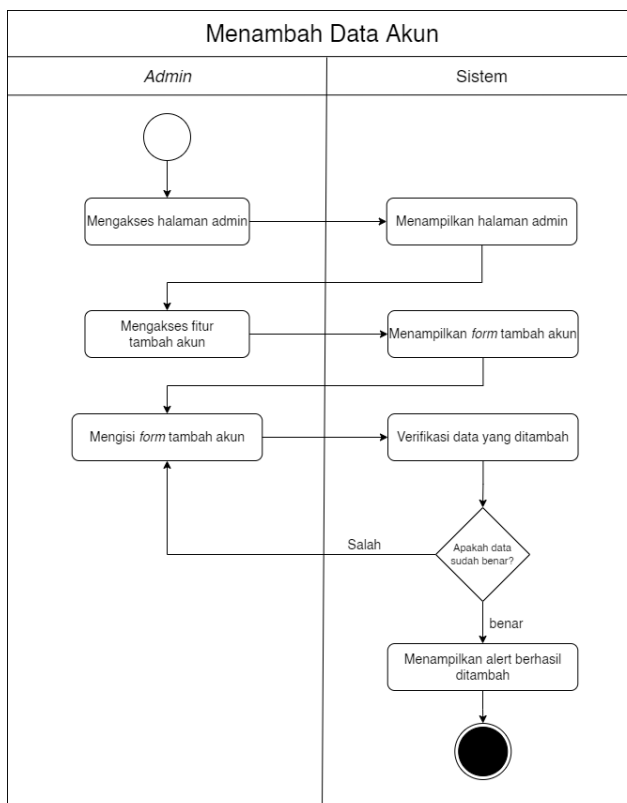
Lampiran 400 Perancangan Masukkan *Login*

Username

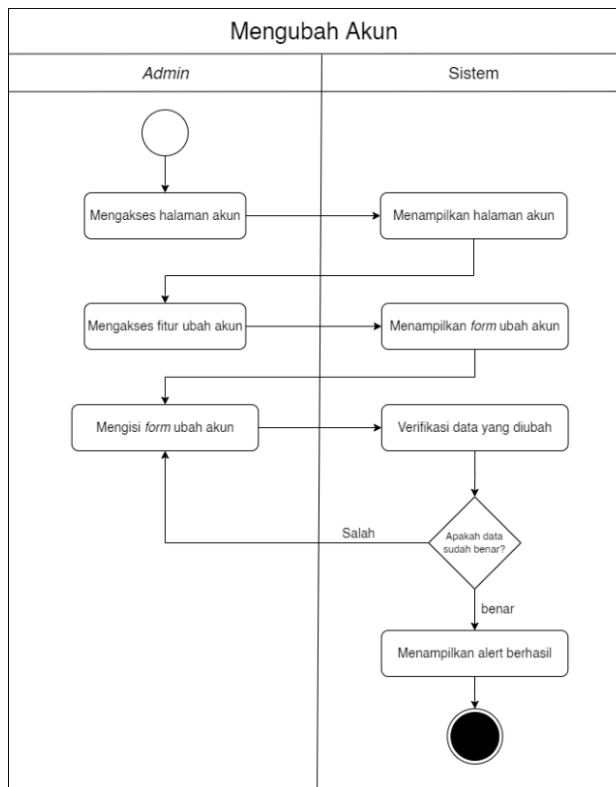
Password

Login

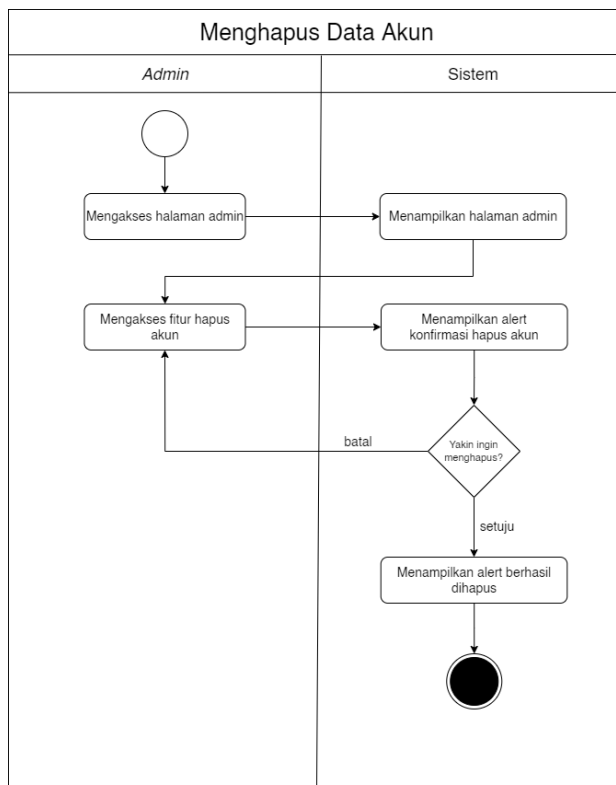
Lampiran 411 Activity Diagram Menambah Akun Admin



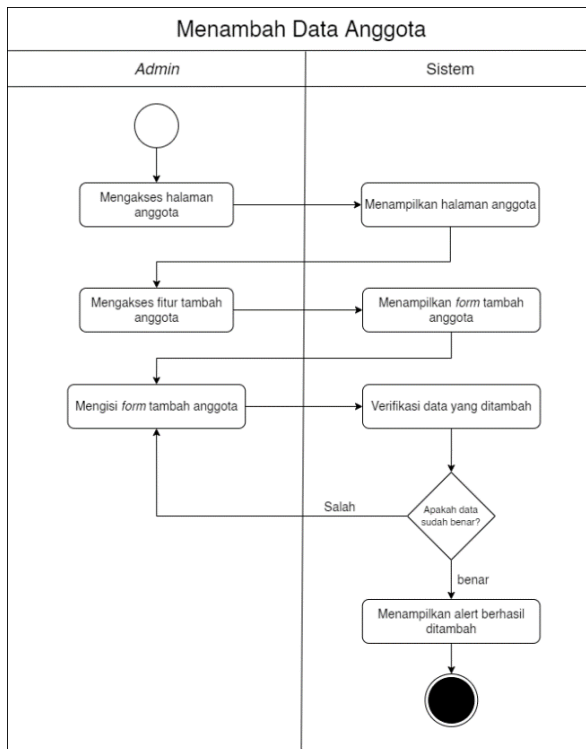
Lampiran 422 Activity Diagram Mengubah Akun Admin



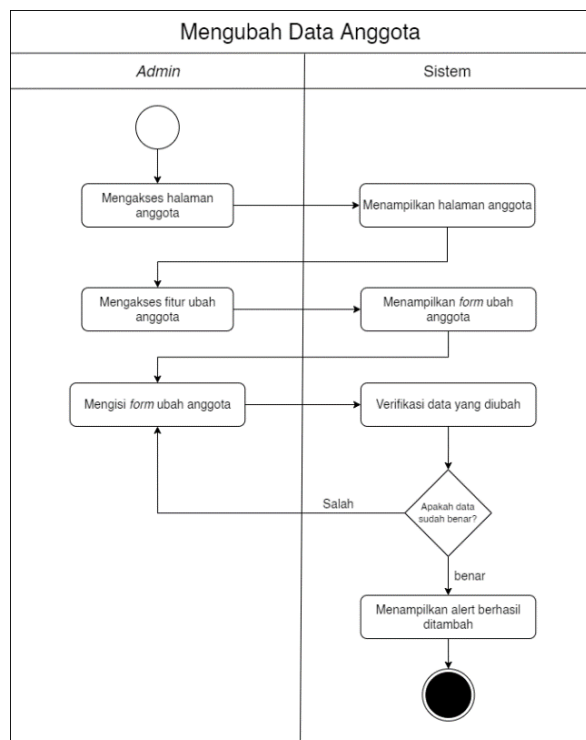
Lampiran 433 Activity Diagram Menghapus Akun Admin



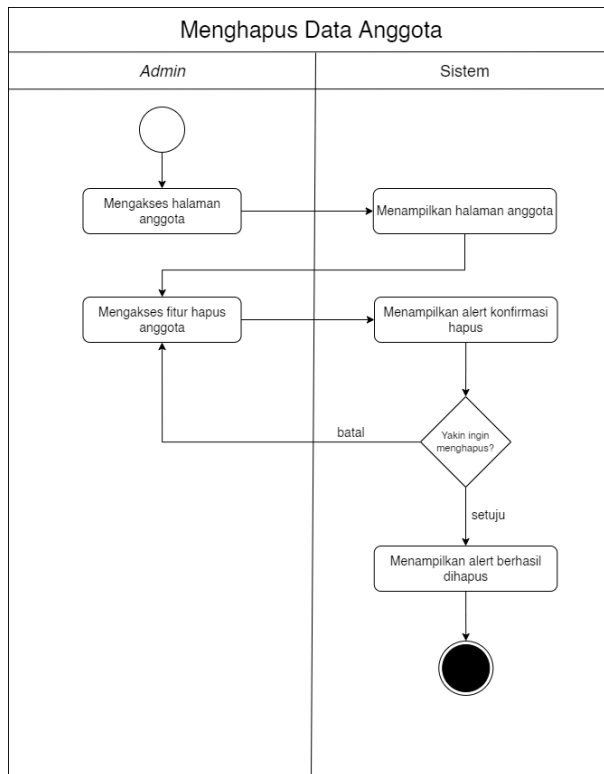
Lampiran 444 Activity Diagram Menambah Data Anggota



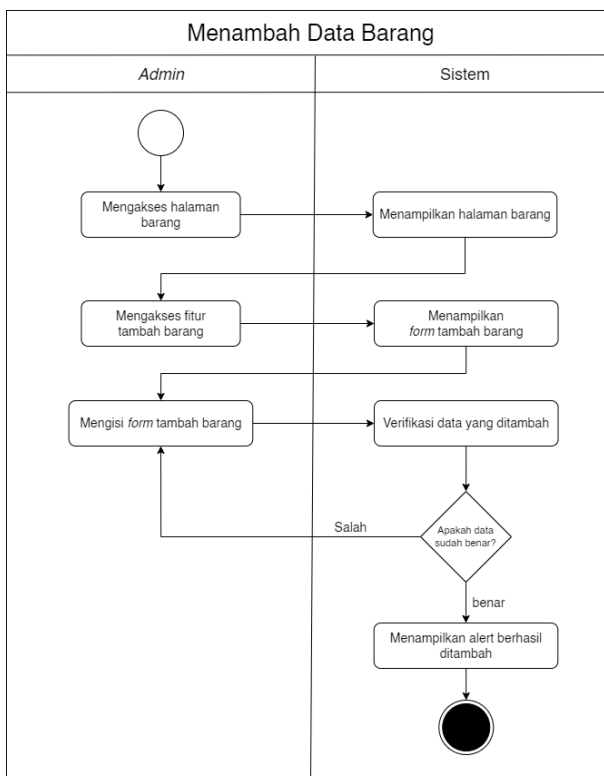
Lampiran 455 Activity Diagram Mengubah Data Anggota



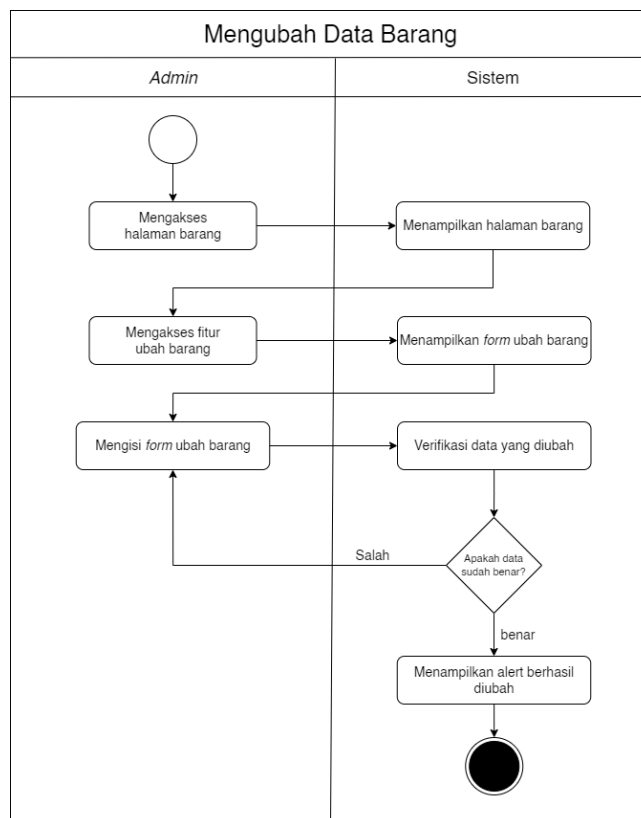
Lampiran 466 Activity Diagram Menghapus Data Anggota



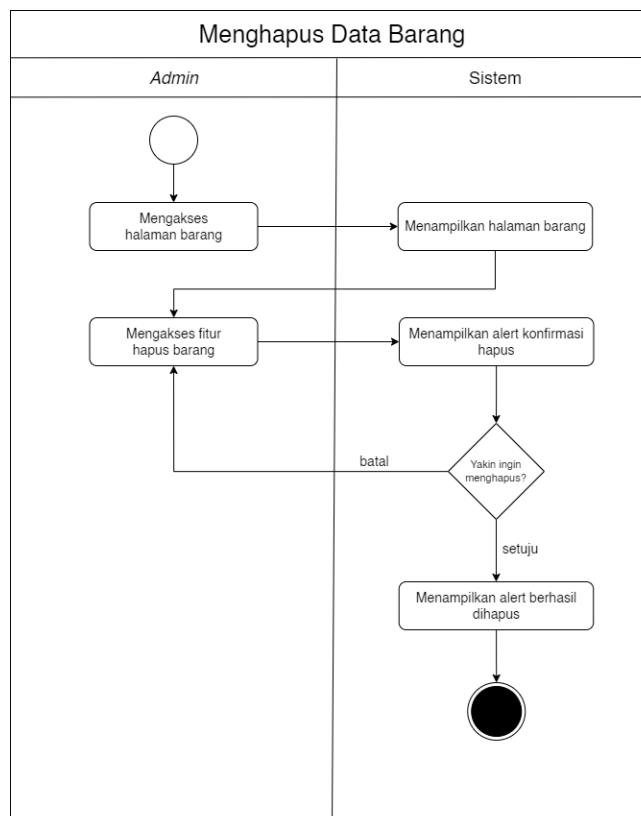
Lampiran 477 Activity Diagram Menambah Data Barang



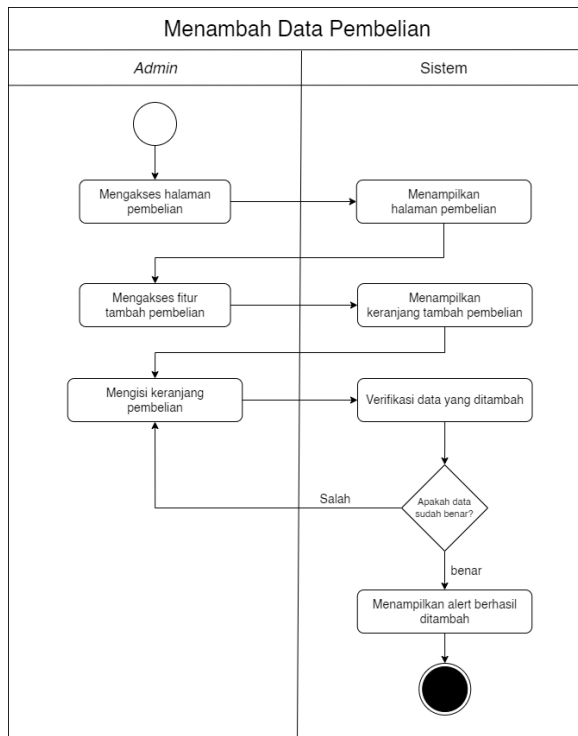
Lampiran 488 Activity Diagram Mengubah Data Barang



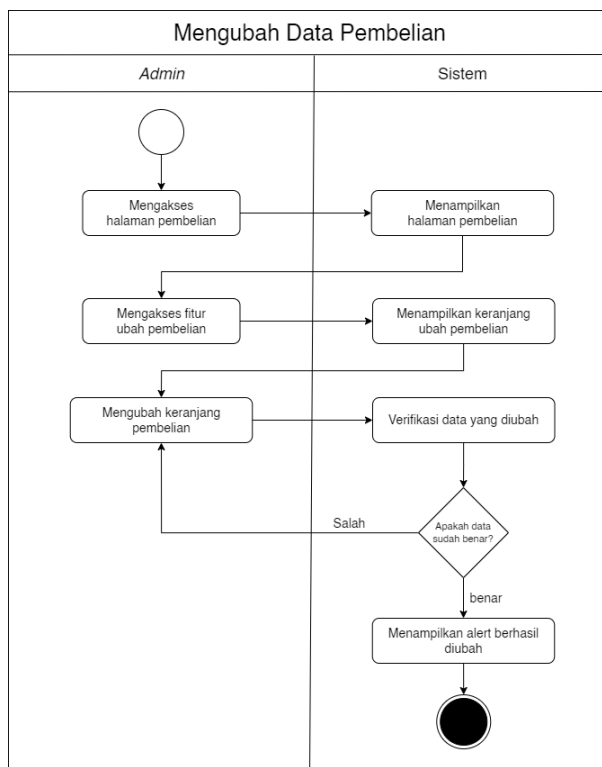
Lampiran 499 Activity Diagram Menghapus Data Barang



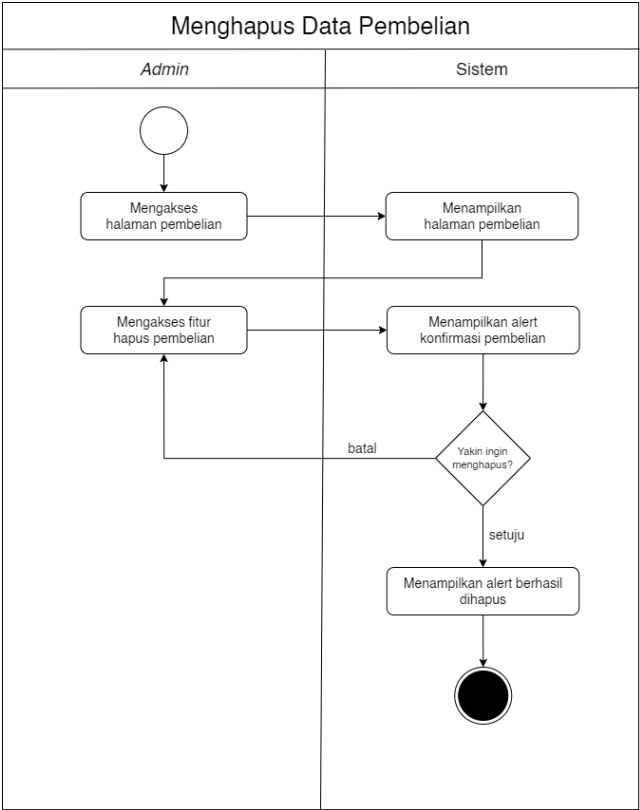
Lampiran 500 Activity Diagram Menambah Data Pembelian



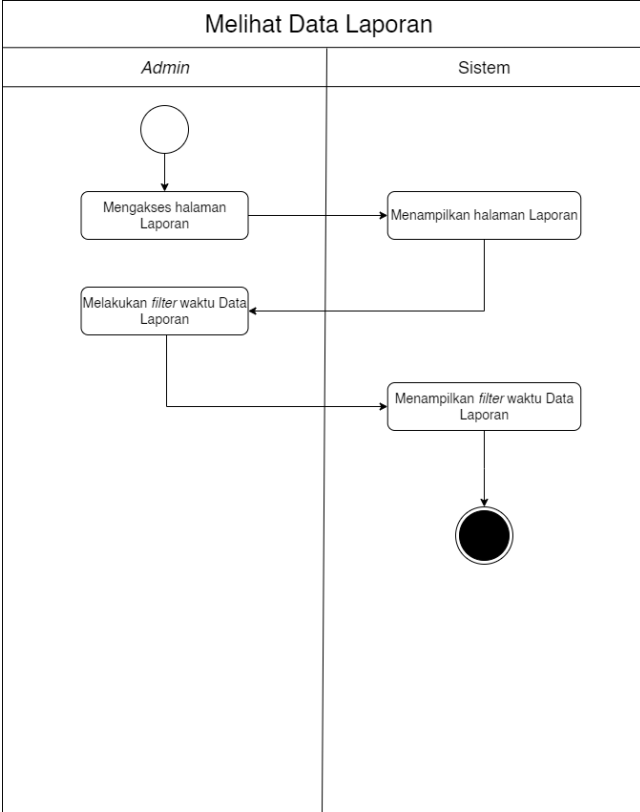
Lampiran 511 Activity Diagram Mengubah Data Pembelian



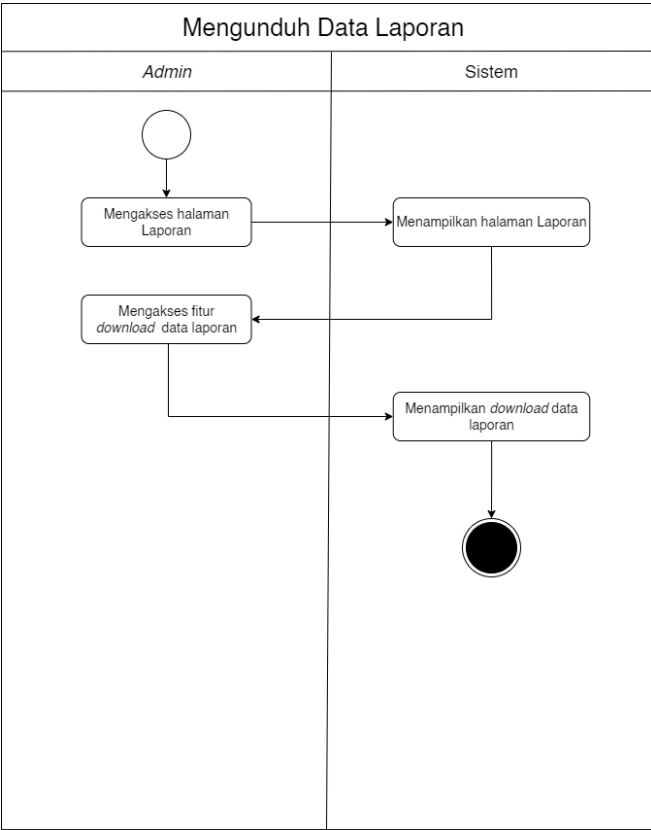
Lampiran 522 Activity Diagram Menghapus Data Pembelian



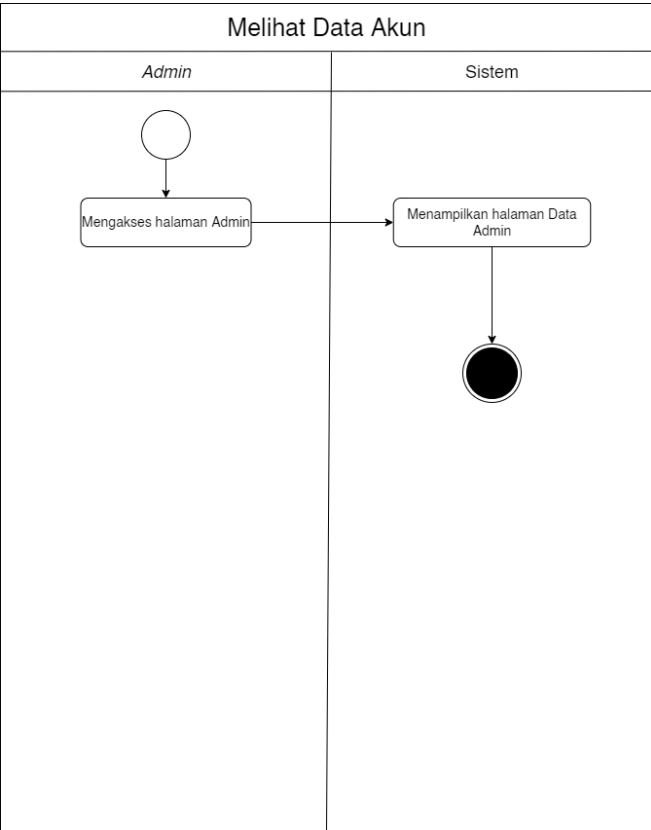
Lampiran 533 Activity Diagram Melihat Data Laporan



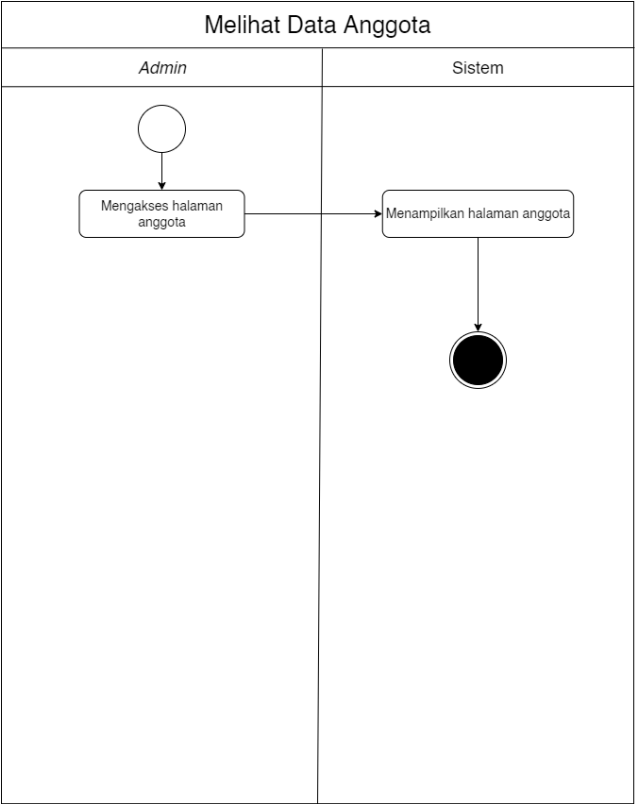
Lampiran 544 Activity Diagram Mengunduh Data Laporan



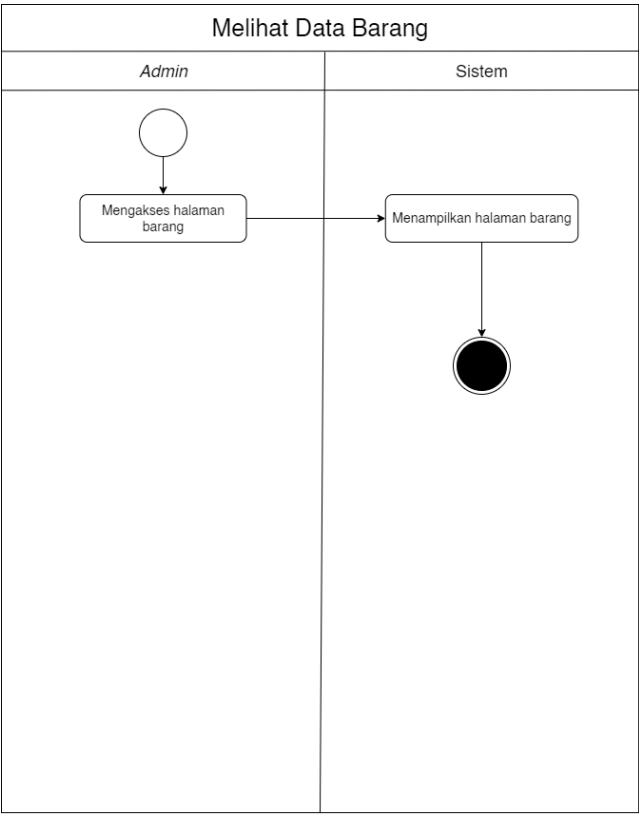
Lampiran 555 Activity Diagram Melihat Data Akun Admin



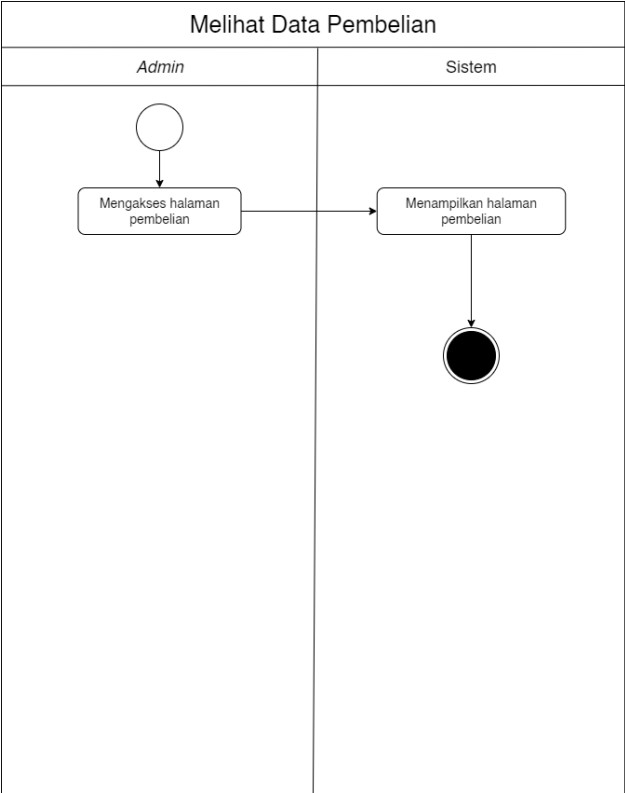
Lampiran 566 Activity Diagram Melihat Data Anggota



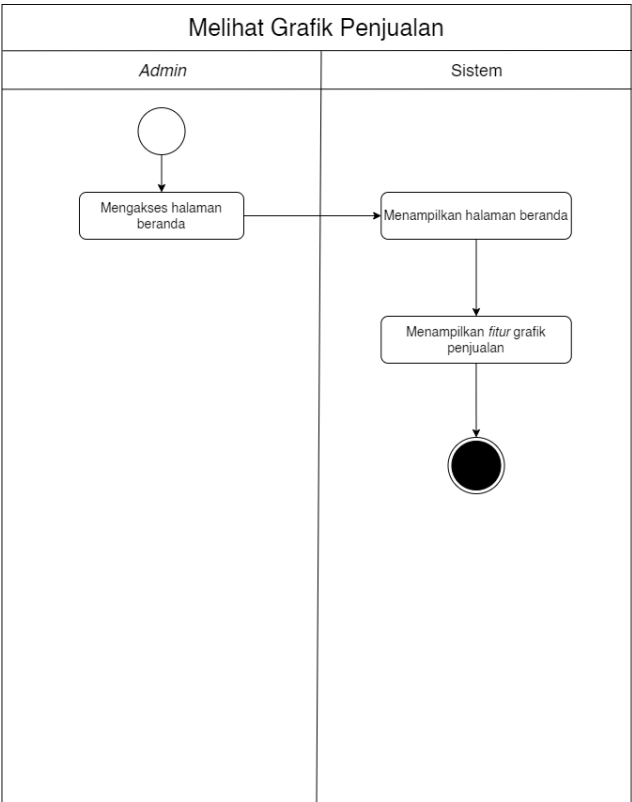
Lampiran 577 Activity Diagram Melihat Data Barang



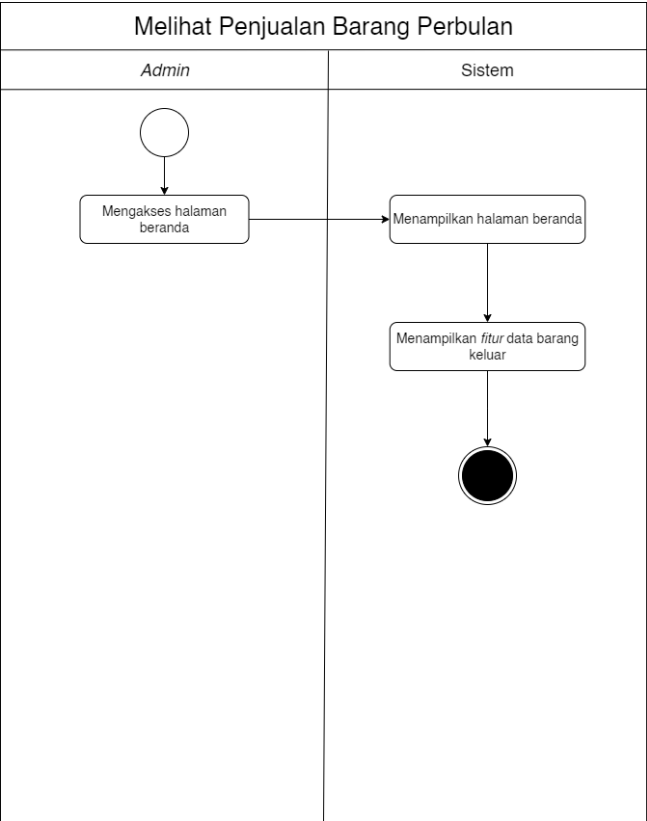
Lampiran 588 Activity Diagram Melihat Data Pembelian



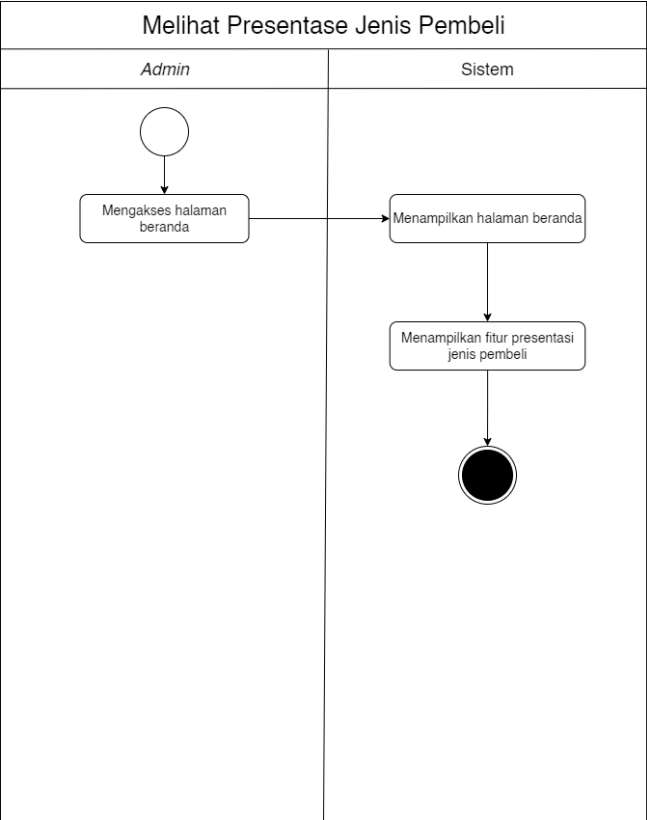
Lampiran 599 Activity Diagram Melihat Grafik Penjualan



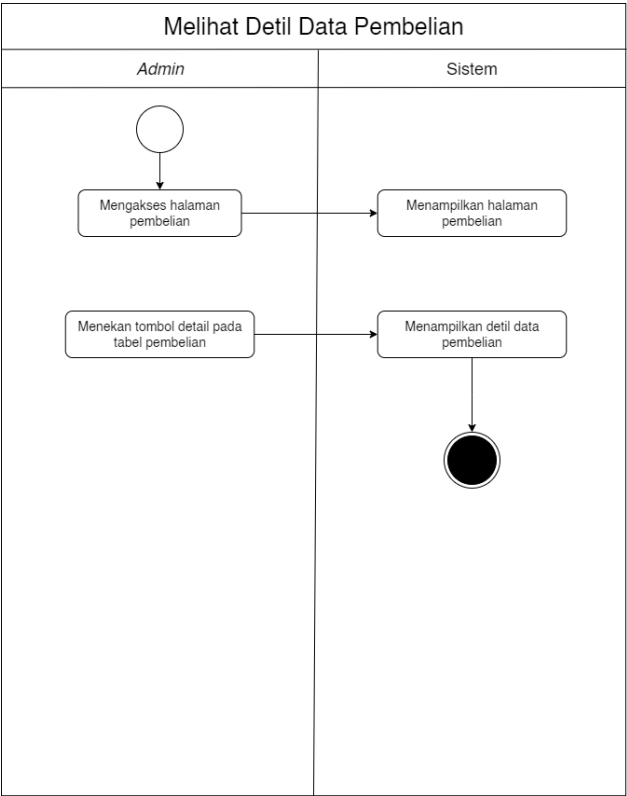
Lampiran 600 Activity Diagram Melihat Data Penjualan Barang Perbulan



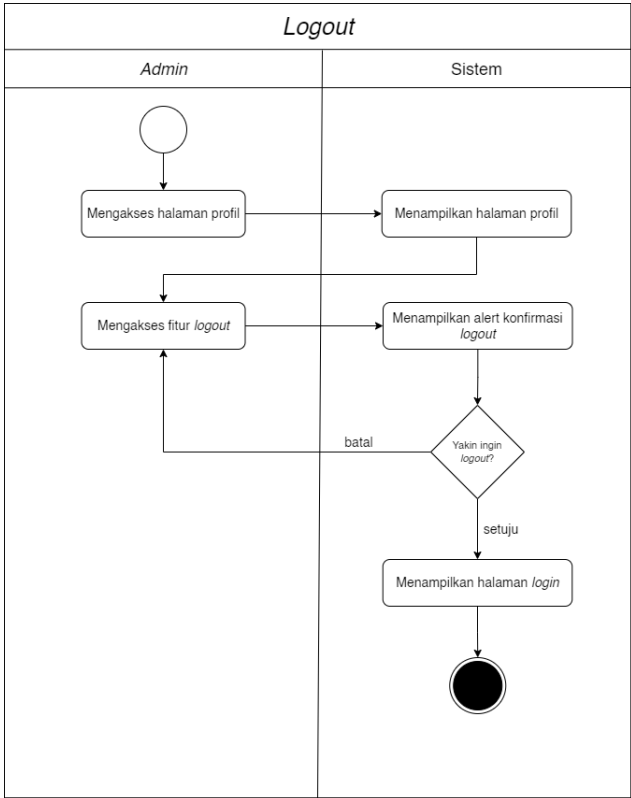
Lampiran 611 Activity Diagram Presentasi Jenis Pembeli



Lampiran 622 *Activity Diagram* Melihat Detil Data Pembelian



Lampiran 633 *Activity Diagram* Logout



Lampiran 644 Perancangan Keluaran Berhasil Menambah Data

Data berhasil ditambah !

Lampiran 655 Perancangan Keluaran Berhasil Mengubah Data

Data berhasil diubah !

Lampiran 666 Perancangan Keluaran Konfirmasi Menghapus Data

Apakah anda yakin ingin menghapus data?




OK

Cancel

Lampiran 677 Perancangan Keluaran Berhasil Menghapus Data

Data berhasil dihapus !

Lampiran 6868 Perancangan Keluaran Tabel Data Akun *Admin*

Header					
Menu Sidebar	<div>  <div>Logout Edit</div> </div>				
	Gambar	Nama	Username	Password	Aksi
					
					

Lampiran 6969 Perancangan Keluaran Tabel Data Anggota

Header				
Menu Sidebar	Anggota			
	No	Nama	Kode Anggota	Aksi
				<div>Ubah Hapus</div>
			<div>Ubah Hapus</div>	

Nama

Kode Anggota

Input

Lampiran 700 Perancangan Keluaran Tabel Data Pembelian

Tabel Pembelian

Show	10	Entries	search	
Pembeli	jumlah	Harga	Tanggal	Aksi
				Detail Ubah Hapus
				Detail Ubah Hapus
Showing 1 to 10 of 2 Entries				previous 1 next

Lampiran 711 Perancangan Keluaran Tabel Data Laporan

Menu Sidebar	Header			
	Laporan PDF Excel			
	Pembeli	Total transaksi	Total Item	Total Harga

Lampiran 722 Implementasi Proses Halaman Beranda

```

public function index()
{
    $db = db_connect();
    $itemjualsql = $db->query("SELECT id_order, nama_item, jumlah_item FROM tbl_jual
                             ORDER BY id_jual DESC LIMIT 7 ");
    $itemjualresult = $itemjualsql->getResultArray();
    foreach ($itemjualresult as $key => $val) {
        $arr_jual[] = $val;
    }
    $data = [
        'title' => 'Beranda',
        'admin' => $this->m_admin->findAll(),
        'logged_in' => session()->get(),
        'itemPerMonth' => $this->m_order->itemsPerMonth(),
        'totalperitem' => $this->m_jual->countPerItem(),
        'tot_item' => $this->m_item->get()->getNumRows(),
        'tot_pembeli' => $this->m_pembeli->get()->getNumRows(),
        'tot_pembelian' => $this->m_order->get()->getNumRows(),
        'tot_sold' => $this->m_order->countItemSold(),
        'tot_jual' => $this->m_jual->countItemSold(),
        'itemjual' => $this->m_jual->findAll(),
        'item' => $this->m_item->findAll(),
        'tabel_jual' => $arr_jual,
        'transaksiperbulan' => $this->m_order->transactionPerMonth(),
        'pembelian' => $this->m_order->findAll(),
    ];
    if (isset($_GET['month'])) {
        if (bulan($_GET['month']) == date("Y")) {
            unset($_GET['month']);
            return redirect()->to(base_url('Home'));
        } else {

```

```

        $data['totalperitem'] = $this->m_jual->countPerItemByMonths($_GET['month']);
    }
    }
    echo view('header');
    echo view('beranda', $data);
    echo view('footer');
}

public function tabel_item_ajax()
{
    if ($this->request->isAJAX()) {
        $postData = $_POST['month'];
        return print_r($postData);
    }
}
}

```

Lampiran 733 Implementasi Proses Halaman Profil *Admin*

```

//Halaman Admin
public function index()
{
    $data = [
        'title' => 'Akun Admin',
        'admin' => $this->adminModel->findAll(),
        'logged_in' => session()->get(),
        'imageDefault' => base_url('assets/images/user_default.png'),
    ];
    echo view('header');
    echo view('Admin/viewAdmin', $data);
    echo view('footer');
}

//Halaman menambah akun admin
public function add()
{
    $data = [
        'title' => 'Tambah Admin',
        'admin' => $this->adminModel->findAll(),
    ];

    echo view('header');
    echo view('Admin/addAdmin', $data);
    echo view('footer');
}

```

```

//Fungsi menambah akun admin
public function create()
{
    if ($this->request->getMethod() === 'post') {
        if (!$this->validate(
            [
                'nama' => 'required',
                'username' => 'required|is_unique[tbl_admin.username]',
                'password' => 'required|min_length[6]|max_length[40]',
                'email' => 'required|valid_emails|valid_email|is_unique[tbl_admin.email]',
                'gambar' => [
                    'mime_in[gambar,image/jpg,image/jpeg,image/gif,image/png]',
                    'max_size[gambar,4096]',
                ],
            ]
        )) {
            $data = ['errors' => implode($this->validator->getErrors())];
            session()->setFlashdata('peringatan', $data['errors']);
            return redirect()->back()->withInput();
        } else {
            if (!$this->request->getFile('gambar')->getName()) {
                $data = array(
                    'username' => $this->request->getPost('username'),
                    'nama' => $this->request->getPost('nama'),
                    'email' => $this->request->getPost('email'),
                    'password' => sha1($this->request->getPost('password')),
                );
            } elseif ($this->request->getFile('gambar')->getName()) {
                $foto = $this->request->getFile('gambar');
                $namabaru = str_replace(' ', '-', $foto->getName());
                $foto->move(WRITEPATH . '../public/assets/images/upload/');
                $data = array(
                    'username' => $this->request->getPost('username'),
                    'nama' => $this->request->getPost('nama'),
                    'email' => $this->request->getPost('email'),
                    'password' => sha1($this->request->getPost('password')),
                    'foto' => $foto->getName(),
                );
            }
        }
        $this->builderAdmin->insert($data);
        session()->setFlashdata('pesan', 'Data Admin Berhasil di tambah');
        return redirect()->to(base_url('Admin'));
    }
}

//fungsi mengubah akun admin
public function edit($id)
{
    $adminData = $this->adminModel->getDataByID($id);

    if ($this->request->getMethod() === 'post') {
        if (!$this->validate(
            [
                'nama' => 'required',
                'email' => 'required|valid_email|valid_emails',
                // 'password' => 'min_length[6]|max_length[40]',
                'gambar' => [

```



```

        'mime_in[gambar,image/jpg,image/jpeg,image/gif,image/png]',
        'max_size[gambar,4096]',
    ],
    )) {
        $data = ['errors' => implode($this->validator->getErrors())];
        session()->setFlashdata('peringatan', $data['errors']);
        return redirect()->back()->withInput();
    } else {
        $email = $this->request->getPost('email');
        $getAll = $this->adminModel->getAllExceptOne($adminData['id_admin']);
        foreach ($getAll as $val) {
            if ($email == $val['email']) {
                session()->setFlashdata('peringatan', 'Email sudah terdaftar, silakan masukkan email lain');
                return redirect()->back()->withInput();
            } else {
                $updateEmail = TRUE;
            }
        }
        if ($email == $adminData['email'] || $updateEmail == TRUE) {
            if (!$this->request->getFile('gambar')->getName() == '' &&
                !$this->request->getPost('password') == '') {
                $foto = $this->request->getFile('gambar');
                $namabaru = str_replace(' ', '-', $foto->getName());
                $foto->move(WRITEPATH . '../public/assets/images/upload/');
                $data = array(
                    'nama' => $this->request->getPost('nama'),
                    'email' => $this->request->getPost('email'),
                    'password' => sha1($this->request->getPost('password')),
                    'foto' => $foto->getName(),
                );
            } elseif ($this->request->getFile('gambar')->getName() == '' &&
                $this->request->getPost('password') == '') {
                $data = array(
                    'nama' => $this->request->getPost('nama'),
                    'email' => $this->request->getPost('email'),
                    // 'password' => sha1($this->request->getPost('password')),
                );
            }
            $this->builderAdmin->update($data, ['id_admin' => $id]);
            session()->setFlashdata('pesan', 'Data Admin Berhasil di perbaharui');
            return redirect()->to(base_url('Admin'));
        }
    }
}

}

$data = [
    'title' => 'Edit Admin',
    'key' => $id,
    'admin' => $adminData,
    'imageDefault' => base_url('assets/images/user_default.png'),
];

echo view('header');
echo view('Admin/editAdmin', $data);
echo view('footer');
}

//fungsi menghapus akun admin
public function delete($id)
{
    $this->builderAdmin->delete(['id_admin' => $id]);
    session()->setFlashdata('pesan', 'Akun admin Berhasil dihapus');
    return redirect()->to(base_url('Admin'));
}
}

```

Lampiran 744 Implementasi Proses Halaman Data Anggota

```

//Halaman Data Anggota
public function index()
{
    $data = [
        'title' => 'Daftar Anggota',
        'pembeli' => $this->pembeliModel->findAll(),
    ];

    echo view('header');
    echo view('Buyer/viewBuyer', $data);
    echo view('footer');
}

//Fungsi menambah data anggota
public function create()
{
    foreach ($this->pembeliModel->findAll() as $key => $value) {
        if (!$this->request->getPost('nama')) {
            session()->setFlashdata('peringatan', 'Nama anggota harus diisi');
            return redirect()->to(base_url('Buyer'));
        }
        if (!$this->request->getPost('nomor_anggota')) {
            session()->setFlashdata('peringatan', 'Nomor anggota harus diisi');
            return redirect()->to(base_url('Buyer'));
        }
        elseif($value['nomor_anggota'] == $this->request->getPost('nomor_anggota')) {
            session()->setFlashdata('peringatan', 'Nomor anggota tidak boleh sama');
            return redirect()->to(base_url('Buyer'));
        }
    }

    $data = array(
        'nama_pembeli' => $this->request->getPost('nama'),
        'nomor_anggota' => $this->request->getPost('nomor_anggota'),
    );

    $this->table->insert($data);

    session()->setFlashdata('pesan', 'Data berhasil ditambah');
    return redirect()->to(base_url('Buyer'));
}

//Halaman ubah data anggota
public function viewEdit($id)
{
    $data = [
        'title' => 'Edit Anggota',
        'pembeli' => $this->pembeliModel->findAll(),
        'id' => $id,
    ];

    echo view('header');
    echo view('Buyer/viewEdit', $data);
    echo view('footer');
}

//Fungsi ubah data anggota
public function edit()
{
    $id = $this->request->getPost('id_pembeli');

    foreach ($this->pembeliModel->findAll() as $key => $value) {

```

```

        $i = 0;
        if ($value['id_pembeli'] == $id) {
            $row = $i;
            break;
        }
        $i++;
    };

    if (!$this->request->getPost('nama')) {
        session()->setFlashdata('peringatan', 'Gagal memperbaharui, nama anggota tidak boleh kosong');
        return redirect()->to(base_url('Buyer/viewEdit/' . $row));
    } elseif (!$this->request->getPost('nomor_anggota')) {
        session()->setFlashdata('peringatan', 'Gagal memperbaharui, Harap nomor anggota
        tidak boleh kosong');
        return redirect()->to(base_url('Buyer/viewEdit/' . $row));
    }

    $data = array(
        'id_pembeli' => $id,
        'nama_pembeli' => $this->request->getPost('nama'),
        'nomor_anggota' => $this->request->getPost('nomor_anggota'),
    );

    $this->table->update($data, ['id_pembeli' => $id]);
    session()->setFlashdata('pesan', 'Data Berhasil di perbaharui');
    return redirect()->to(base_url('Buyer'));
}

//Fungsi menghapus data anggota
public function delete($id)
{
    $this->table->delete(['id_pembeli' => $id]);

    session()->setFlashdata('pesan', 'Data Anggota berhasil dihapus');
    return redirect()->to(base_url('Buyer'));
}
}

```

Lampiran 755 Implementasi Proses Halaman Data Pembelian

```

//Halaman Data Pembelian
public function index()
{
    $data = [
        'title' => 'Daftar Pembelian',
        'orders' => $this->orderModel->get_item_list('id_pembelian', 'DESC'),
        'sort_by' => 'id_pembelian',
        'sort_order' => 'DESC',
        'item' => $this->itemModel->getAll(),
    ];
    echo view('header');
    echo view('Orders/orderView', $data);
    echo view('footer');
}

//Halaman menambah data pembelian
public function input()
{
    $cart = \Config\Services::cart();
    // $cart->destroy();
    $data = [
        'title' => 'Input Pembelian',
        'item' => $this->itemModel->getAll(),
        'pembeli' => $this->pembeliModel->getBarang(),
        'cart' => \Config\Services::cart(),
    ];
    echo view('header');
    echo view('orders/input', $data);
    echo view('footer');
}

//Halaman detail data pembelian
public function viewDetail($id_pembelian)
{
    $data = [
        'title' => 'Detail Pembelian',
        'orders' => $this->orderModel->findAll(),
        'itemJual' => $this->itemJualModel->getAll(),
        'id' => $id_pembelian,
    ];

    echo view('header');
    echo view('Orders/viewDetail', $data);
    echo view('footer');
}

//Fungsi menambah data pembelian
public function add($id)
{
    $cart = \Config\Services::cart();
    $m_item = new Model_item();

    $item = $m_item->getByID($id);

    if ($item) {
        $cart->insert(array(
            'id_item' => $item[0]['id_item'],
            'jumlah_item' => 1,
            'harga_item' => $item[0]['harga_item'],
            'nama_item' => $item[0]['nama_item'],
        ));
    }
}

```

```

        session()->setFlashdata('pesan', 'Barang berhasil ditambahkan !');
        return redirect()->back()->withInput();
    }

    //Fungsi membersihkan keranjang pembelian
    public function clear()
    {
        $cart = \Config\Services::cart();
        $cart->destroy();
        return redirect()->to(base_url('/Orders/input'));
    }

    public function clearPerItem($index)
    {
        $cart = \Config\Services::cart();
        $cart->remove($index);
        return redirect()->back()->withInput();
    }

    //Fungsi menghapus data pembelian
    public function deleteOrder($id_pembelian)
    {
        $this->builderPembelian->delete(['id_pembelian' => $id_pembelian]);
        $this->builderJual->delete(['id_order' => $id_pembelian]);
        session()->setFlashdata('pesan', 'Data Berhasil di Hapus !');
        return redirect()->to(base_url('Orders'));
    }

    //Fungsi mengubah data pembelian
    public function editOrder($id_pembelian)
    {
        $data = [
            'title' => 'Edit Pembelian',
            'item' => $this->itemModel->getAll(),
            'pembeli' => $this->pembeliModel->getBarang(),
            'order' => $this->orderModel->getById($id_pembelian),
            'itemJual' => $this->itemJualModel->getAll(),
            'cart' => \Config\Services::cart(),
            'id' => $id_pembelian,
        ];
        echo view('header');
        echo view('Orders/edit', $data);
        echo view('footer');
    }

    //Fungsi menambah barang ke keranjang
    public function itemToCart($id_pembelian)
    {
        $cart = \Config\Services::cart();
        $cart->destroy();
        $itemJual = $this->itemJualModel->getAll();
        foreach ($itemJual as $key => $row) :
            if ($itemJual[$key]['id_order'] == $id_pembelian) {
                $rowbyorder = $itemJual[$key];
                $cart->insert(array(
                    'id_item' => $rowbyorder['id_item'],
                    'jumlah_item' => $rowbyorder['jumlah_item'],
                    'harga_item' => $rowbyorder['harga_item'],
                    'nama_item' => $rowbyorder['nama_item'],
                ));
            }
        endforeach;
    }

```

```

if (isset($pembeli2)) {
    $dataOrder = array(
        'nama_pembeli' => $pembeli2,
        'total_item' => $cart->totalItems(),
        'total_harga' => $cart->total(),
        'penanggung_jawab' => session()->get()['username'],
    );
    $this->db->table('tbl_pembelian')->insert($dataOrder);
    // return print_r('oke masbroww');
} else {

    $pembeli = explode('|', $this->request->getPost('pembeli'));

    $dataOrder = array(
        'id_pembeli' => $pembeli[0],
        'nama_pembeli' => $pembeli[1],
        'total_item' => $cart->totalItems(),
        'total_harga' => $cart->total(),
        'penanggung_jawab' => session()->get()['username'],
    );
    $this->db->table('tbl_pembelian')->insert($dataOrder);
}

$id_order = $this->db->insertID();

$i = 0;
foreach ($cart->contents() as $key => $value) {
    $dataItem[] = array(
        'id_item' => $value['id_item'],
        'id_order' => $id_order,
        'nama_item' => $value['nama_item'],
        'harga_item' => $value['harga_item'],
        'jumlah_item' => $value['jumlah_item'],
        'subtotal_harga' => $value['harga_item'] * $value['jumlah_item'],
    );
    $i++;
}
$this->builderJual->insertBatch($dataItem);
$model = $this->itemModel->findAll();
foreach ($model as $itemkey => $itemvalue) {

    if ($itemvalue['stok_item'] != 0) {
        $items[] = $itemvalue;
        $itemkey = $itemkey;
    }
}

$j = 0;
foreach ($dataItem as $key => $value) {

    foreach ($items as $keyItem => $item) {
        if ($item['id_item'] == $value['id_item']) {
            $updateStok = $item['stok_item'] - $value['jumlah_item'];
            $this->builderItem->where('id_item', $item['id_item']);
            $this->builderItem->update([
                'stok_item' => $updateStok,
            ]);
        }
        $j++;
    }
}

$cart->destroy();
session()->setFlashdata('sukses', 'Data Keranjang Berhasil di Simpan !');
return redirect()->to(base_url('Orders/input'));

```

```

case 'update':
    $cart = \Config\Services::cart();
    $i = 1;
    $j = 0;
    $model = $this->itemModel->findAll();

    foreach ($model as $itemkey => $itemvalue) {

        if ($itemvalue['stok_item'] != 0) {
            $items[] = $itemvalue;
            $itemkey = $itemkey;
        }
    }

    foreach ($cart->contents() as $key => $value) {

        foreach ($items as $keyItem => $item) {
            if ($item['id_item'] == $value['id_item']) {
                if ($item['stok_item'] >= $this->request->getPost('jumlah_item' . $i)) {
                    $cart->update(array(
                        'rowid' => $value['rowid'],
                        'jumlah_item' => $this->request->getPost('jumlah_item' . $i),
                    ));
                } elseif ($item['stok_item'] < $this->request->getPost('jumlah_item' . $i)) {
                    session()->setFlashdata('peringatan',
                        'Jumlah Barang <b>' . $item['nama_item'] . '</b> Melebihi Stok !');
                    return redirect()->back()->withInput();
                }
            } else $j++;
        }
        $i++;
    }

    session()->setFlashdata('pesan', 'Data Keranjang Berhasil di Perbaharui !');
    return redirect()->back()->withInput();

case 'saveupdate':
    $cart = \Config\Services::cart();

    $id = $this->request->getPost('id_order');
    $id_pembeli = $this->request->getPost('pembeli');
    $pembeli2 = $this->request->getPost('nonanggota');
    // print_r($pembeli);
    if (isset($pembeli) && empty($pembeli2)) {
        session()->setFlashdata('peringatan', 'Harap pembeli diisi terlebih dahulu');
        return redirect()->back()->withInput();
        break;
    } elseif (!empty($pembeli) && !empty($pembeli2)) {
        session()->setFlashdata('peringatan', 'Harap pilih jenis pembeli terlebih dahulu');
        return redirect()->back()->withInput();
        break;
    } elseif (empty($cart->contents())) {
        session()->setFlashdata('peringatan', 'Harap barang diisi terlebih dahulu');
        return redirect()->back()->withInput();
        break;
    }

    if ($id_pembeli == 0) {
        $dataOrder = array(
            'id_pembelian' => $id,
            'nama_pembeli' => $pembeli2,
            'total_item' => $cart->totalItems(),
            'total_harga' => $cart->total(),
            'penanggung_jawab' => session()->get()['username'],
        );
    }

```

```

        $this->orderModel->edit($dataOrder);
        // return print_r('oke masbroww');
    } else {
        $pembeli = $this->pembeliModel->getById($id_pembeli);

        $dataOrder = array(
            'id_pembelian' => $id,
            'id_pembeli' => $pembeli[0]['id_pembeli'],
            'nama_pembeli' => $pembeli[0]['nama_pembeli'],
            'total_item' => $cart->totalItems(),
            'total_harga' => $cart->total(),
            'penanggung_jawab' => session()->get()['username'],
        );
        // return print_r($dataOrder);
        $this->orderModel->edit($dataOrder);
    }
    $id_order = $id;

    $i = 0;
    foreach ($cart->contents() as $value) {
        $dataItem[] = array(
            'id_item' => $value['id_item'],
            'id_order' => $id_order,
            'nama_item' => $value['nama_item'],
            'harga_item' => $value['harga_item'],
            'jumlah_item' => $value['jumlah_item'],
            'subtotal_harga' => $value['harga_item'] * $value['jumlah_item'],
        );
        $i++;
    }
    // return print_r($dataItem);
    $model = $this->itemModel->findAll();
    foreach ($model as $itemkey => $itemvalue) {

        if ($itemvalue['stok_item'] != 0) {
            $items[] = $itemvalue;
            $itemkey = $itemkey;
        }
    }
    $orderan = $this->itemJualModel->getByOrder($id_order);
    $j = 0;
    foreach ($dataItem as $itemchange) {

        foreach ($orderan as $perorder) {
            foreach ($items as $item) {
                $selisih = 0;
                if (($item['id_item'] == $itemchange['id_item']) && ($itemchange['id_item'] ==
                    $perorder['id_item'])) {
                    // return print_r($perorder['jumlah_item']);
                    if ($perorder['jumlah_item'] > $itemchange['jumlah_item']) {
                        $selisih = $perorder['jumlah_item'] - $itemchange['jumlah_item'];
                        $updateStok = $item['stok_item'] + $selisih;
                        $this->builderItem->where('id_item', $item['id_item']);
                        $this->builderItem->update([
                            'stok_item' => $updateStok,
                        ]);
                        // return print_r('ngurang');
                        break;
                    } elseif ($perorder['jumlah_item'] < $itemchange['jumlah_item']) {
                        $selisih = $itemchange['jumlah_item'] - $perorder['jumlah_item'];
                        $updateStok = $item['stok_item'] - $selisih;
                        $this->builderItem->where('id_item', $item['id_item']);

```



```

        $this->builderItem->where('id_item', $item['id_item']);
        $this->builderItem->update([
            'stok_item' => $updateStok,
        ]);
        // return print_r('nambah');
        break;
    } else {
        $updateStok = $item['stok_item'] - $itemchange['jumlah_item'];
        $this->builderItem->where('id_item', $item['id_item']);
        $this->builderItem->update([
            'stok_item' => $updateStok,
        ]);
        // return print_r($selisih);
    }
}
}
break;
$j++;
}
}

$this->builderJual->delete(['id_order' => $id_order]);
$this->builderJual->insertBatch($dataItem);
// $this->builderJual->updateBatch($dataItem);

$cart->destroy();

session()->setFlashdata('pesan', 'Data Keranjang Berhasil di Simpan !');
return redirect()->to(base_url('Orders'));

```

Lampiran 766 Implementasi Proses Halaman Laporan

```

//Halaman data laporan
public function index()
{
    echo view('header');
    $this->viewData();
    echo view('footer');
}

//Fungsi menampilkan data laporan
public function viewData()
{
    $nama_bulan = '';
    if (!$this->request->getPost('datefrom') or !$this->request->getPost('dateto')) {
        $result = $this->orderModel->findAll();
    } else {
        $datefrom = date("Y-m-d", strtotime($this->request->getPost('datefrom')));
        $dateto = date("Y-m-d", strtotime($this->request->getPost('dateto')));

        $result = $this->orderModel->getfilterdata($datefrom, $dateto);
    }

    if (isset($_GET['year'])) {
        $daftar pembeli = $this->pembeliModel->findAll();
        $dataperyear = $this->orderModel->getByYear($_GET);

        foreach ($daftar pembeli as $pembeli) {
            $totalItem = 0;
            $totalHarga = 0;
            $totalTransaksi = 0;
            foreach ($dataperyear as $peryear) {

```

```

        if ($pyear['id_pembeli'] == $pembeli['id_pembeli']) {
            $totalItem += $pyear['total_item'];
            $totalHarga += $pyear['total_harga'];
            $totalTransaksi++;
        }
    }
    $dataReport[] = array(
        'id_pembeli' => $pembeli['id_pembeli'],
        'nama_pembeli' => $pembeli['nama_pembeli'],
        'total_transaksi' => $totalTransaksi,
        'total_item' => $totalItem,
        'total_harga' => $totalHarga,
    );
}
} else {

    foreach ($this->pembeliModel->findAll() as $key => $row) {
        $totalItem = 0;
        $totalHarga = 0;
        $totalTransaksi = 0;
        foreach ($result as $keyorder => $roworder) {
            if ($row['id_pembeli'] == $roworder['id_pembeli']) {
                $totalItem += $roworder['total_item'];
                $totalHarga += $roworder['total_harga'];
                $totalTransaksi++;
            }
        }
        $tahun = $this->orderModel->distinctYear();
        foreach ($tahun as $year) {
            $listTahun[] = $year['YEAR(tanggal_pembelian)'];
            sort($listTahun);
        }

        $dataReport[] = array(
            'id_pembeli' => $row['id_pembeli'],
            'nama_pembeli' => $row['nama_pembeli'],
            'total_transaksi' => $totalTransaksi,
            'total_item' => $totalItem,
            'total_harga' => $totalHarga,
        );
    }
}

}
}

$data = [
    'title' => 'Daftar Hasil Pembelian',
    'dataReport' => $dataReport,
    'orders' => $this->orderModel->findAll(),
    'datefrom' => '',
    'dateto' => '',
    'bulan' => $nama_bulan,
    'tahun' => $this->orderModel->distinctYear(),
];
echo view('Report/index', $data);
}

//Fungsi menampilkan data pertahun
public function pertahun()
{
    if (!$this->request->getPost('datefrom') or !$this->request->getPost('dateto')) {
        $result = $this->orderModel->findAll();
    } else {
        $datefrom = date("Y-m-d", strtotime($this->request->getPost('datefrom')));
        $dateto = date("Y-m-d", strtotime($this->request->getPost('dateto')));

        $result = $this->orderModel->getfilterdata($datefrom, $dateto);
    }
}

```

```

if (isset($_GET['month'])) {
    $selected_month = $_GET['month'];
    $result = $this->orderModel->filterByMonth($selected_month);
}

foreach ($this->pembeliModel->findAll() as $key => $row) {
    $totalItem = 0;
    $totalHarga = 0;
    $totalTransaksi = 0;
    foreach ($result as $keyorder => $roworder) {
        if ($row['id_pembeli'] == $roworder['id_pembeli']) {
            $totalItem += $roworder['total_item'];
            $totalHarga += $roworder['total_harga'];
            $totalTransaksi++;
        }
    }

    $dataReport[] = array(
        'id_pembeli' => $row['id_pembeli'],
        'nama_pembeli' => $row['nama_pembeli'],
        'total_transaksi' => $totalTransaksi,
        'total_item' => $totalItem,
        'total_harga' => $totalHarga,
    );
}

$data = [
    'title' => 'Daftar Hasil Pembelian',
    'dataReport' => $dataReport,
    'orders' => $this->orderModel->findAll(),
    'datefrom' => '',
    'dateto' => '',
];
echo view('header');
echo view('Report/pertahun', $data);
echo view('footer');
}

//Fungsi mensortir data berdasarkan tanggal
public function filterdata()
{
    $datefrom = date("Y-m-d", strtotime($this->request->getPost('datefrom')));
    $dateto = date("Y-m-d", strtotime($this->request->getPost('dateto')));

    $result = $this->orderModel->getfilterdata($datefrom, $dateto);

    foreach ($this->pembeliModel->findAll() as $key => $row) {
        $totalItem = 0;
        $totalHarga = 0;
        $totalTransaksi = 0;
        foreach ($result as $keyorder => $roworder) {
            if ($row['id_pembeli'] == $roworder['id_pembeli']) {
                $totalItem += $roworder['total_item'];
                $totalHarga += $roworder['total_harga'];
                $totalTransaksi++;
            }
        }
    }
    $this->builderPembelian->countAllResults();

    $dataReport[] = array(
        'id_pembeli' => $row['id_pembeli'],
        'nama_pembeli' => $row['nama_pembeli'],
        'total_transaksi' => $totalTransaksi,
        'total_item' => $totalItem,
        'total_harga' => $totalHarga,
    );
}

```

```

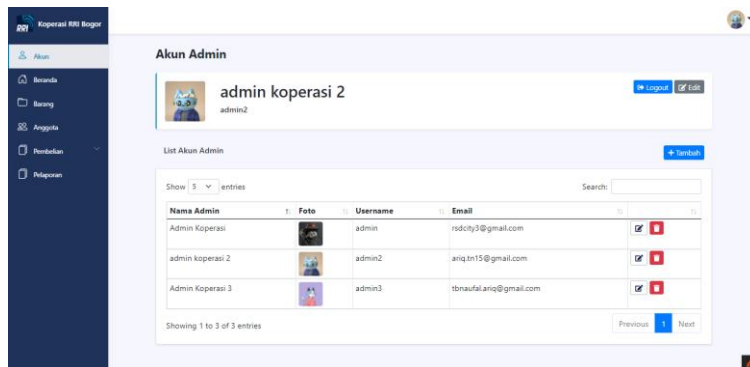
    );
}

$data = [
    'title' => 'Daftar Hasil Pembelian',
    'dataReport' => $dataReport,
    'orders' => $this->orderModel->findAll(),
    'datefrom' => $datefrom,
    'dateto' => $dateto,
    'tahun' => $this->orderModel->distinctYear(),
];
echo view('header');
echo view('Report/index', $data);
echo view('footer');
}

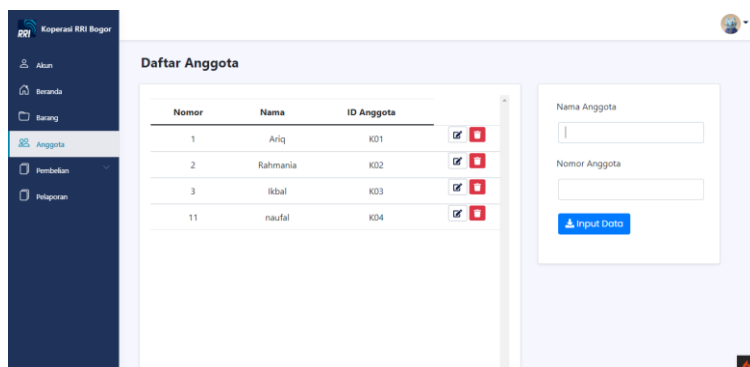
public function deleteAll($id)
{
    foreach ($this->orderModel->findAll() as $key => $value) {
        if ($value['id_pembeli'] == $id) {
            foreach ($this->itemJualModel->findAll() as $k => $val) {
                if ($value['id_pembelian'] == $val['id_order']) {
                    $this->builderJual->delete(['id_order' => $value['id_pembelian']]);
                }
            }
            $this->builderPembelian->delete(['id_pembeli' => $id]);
        }
    }
    session()->setFlashdata('pesan', 'Data berhasil dihapus');
    return redirect()->to(base_url('report'));
}
}

```

Lampiran 777 Implementasi Antarmuka Profil *Admin*



Lampiran 788 Implementasi Antarmuka Halaman Data Anggota



Lampiran 799 Implementasi Antarmuka Halaman Data Barang

Daftar Barang

Input Data

Show 5 entries

No.	Nama	Kode	Gambar	Harga	Stok	Aksi
5	Makanan Ringan	0		Rp. 5.000,00	40	
6	Keripik Duren	0		Rp. 10.000,00	29	
7	Keripik Nangka	0		Rp. 7.500,00	15	
8	Keripik	0		Rp. 5.000,00	42	
9	Minuman	0		Rp. 5.000,00	40	

Showing 6 to 10 of 19 entries

Lampiran 800 Implementasi Antarmuka Halaman Data Pembelian

Daftar Pembelian

Data Riwayat

Show 5 rows Copy Excel PDF Print

Search:

Order ID	Nama Pembeli	Jumlah Barang	Total Harga	Tanggal	Aksi
139	Rahmania	5	Rp. 45.000,00	22 Apr 2022	
138	Ariq	2	Rp. 10.000,00	22 Apr 2022	
137	Rahmania	5	Rp. 25.000,00	21 Apr 2022	
136	Rahmania	4	Rp. 20.000,00	21 Apr 2022	
135	Ariq	1	Rp. 5.000,00	21 Apr 2022	

Showing 1 to 5 of 55 entries

Lampiran 811 Implementasi Antarmuka Halaman Laporan

Daftar Hasil Pembelian

Tahun

Pilih Tanggal

Show 5 rows Copy Excel PDF Print

Search:

Nama Pembeli	Total Transaksi	Total Item Dibeli	Total Nominal Item	Aksi
Rahmania	21	68	Rp. 960.000,00	
naufal	2	6	Rp. 210.000,00	
ikbal	9	38	Rp. 510.000,00	
Ariq	23	77	Rp. 1.690.616,00	

Showing 1 to 4 of 4 entries

Lampiran 822 Implementasi Masukkan Tambah Akun Admin

Nama

Username

Email

Password


Gambar

Choose File No file chosen

Submit

preview

Lampiran 833 Implementasi Masukkan Ubah Profil *Admin*




Nama

Username

Email

Password

Gambar



preview

Lampiran 844 Implementasi Masukkan Tambah Data Anggota

Nama Anggota

Nomor Anggota

Lampiran 855 Implementasi Masukkan Ubah Data Anggota

Nama Anggota

Nomor Anggota

Lampiran 866 Implementasi Masukkan Ubah Data Barang


Nama Barang


Stok

Harga

Gambar

Gambar





Lampiran 877 Implementasi Masukkan Tambah Data Pembelian

Pembeli

Nama	jumlah	Total
Total Barang		0
Total		IDR 0

Lampiran 888 Implementasi Masukkan Ubah Data Pembelian

Nama Anggota

Nama	jumlah	Total
Minuman	<input type="text" value="5"/>	IDR 25,000
Beras	<input type="text" value="1"/>	IDR 50,000
Total Barang	6	
Total		IDR 75,000

Lampiran 899 Implementasi Keluaran Berhasil Menambah Data

Data Berhasil di Simpan

Lampiran 9090 Implementasi Keluaran Berhasil Mengubah Data

Data Berhasil di perbaharui


Lampiran 9191 Implementasi Keluaran Konfirmasi Menghapus Data

Apakah anda yakin ingin menghapus data?

Lampiran 9292 Implementasi Keluaran Berhasil Menghapus Data





Data Berhasil di hapus

Lampiran 9393 Implementasi Keluaran Tabel Data Akun *Admin*

 **Hadi Irawan**
admin [Logout](#) [Edit](#)

List Akun Admin [+ Tambah](#)

Show 5 entries Search:

Nama Admin	Foto	Username	Email	
ariq		admin5	tubagus_ariq@apps.ipb.ac.id	Edit Delete
Hadi Irawan		admin	rsdcity3@gmail.com	Edit Delete
Halimah		admin3	tbnauful.ariq@gmail.com	Edit Delete
Septian		admin2	ariq.tn15@gmail.com	Edit Delete

Showing 1 to 4 of 4 entries [Previous](#) [1](#) [Next](#)

Lampiran 9494 Implementasi Keluaran Tabel Data Anggota

Nomor	Nama	ID Anggota	
1	Ariq	K01	Edit Delete
2	Rahmania	K02	Edit Delete
3	Ikbal	K03	Edit Delete
11	naufal	K04	Edit Delete

Lampiran 9595 Implementasi Keluaran Tabel Data Pembelian

Show 5 rows [Copy](#) [Excel](#) [PDF](#) [Print](#) Search:

Order ID	Nama Pembeli	Jumlah Barang	Total Harga	Tanggal	Penanggung Jawab	
141	Ikbal	4	Rp. 20.000,00	08 May 2022	admin2	Edit Delete
140	Ikbal	4	Rp. 20.000,00	08 May 2022	admin2	Edit Delete
139	Rahmania	5	Rp. 45.000,00	22 Apr 2022	admin2	Edit Delete
138	Ariq	2	Rp. 10.000,00	22 Apr 2022	admin2	Edit Delete
137	Rahmania	5	Rp. 25.000,00	21 Apr 2022	admin2	Edit Delete

Showing 1 to 5 of 57 entries [Previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [...](#) [12](#) [Next](#)

Lampiran 9696 Implementasi Keluaran Tabel Data Laporan

Show 5 rows [Copy](#) [Excel](#) [PDF](#) [Print](#) Search:

Nama Pembeli	Total Transaksi	Total Item Dibeli	Total Nominal Item	
Rahmania	21	68	Rp. 960.000,00	Delete
naufal	2	6	Rp. 210.000,00	Delete
Ikbal	11	46	Rp. 550.000,00	Delete
Ariq	23	77	Rp. 1.690.616,00	Delete

Showing 1 to 4 of 4 entries [Previous](#) [1](#) [Next](#)

Lampiran 9797 Hasil Pengujian Menambah Akun *Admin*

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Menambah Akun	Benar	Mengisi <i>form</i> dengan data yang benar.	1. Menampilkan <i>alert</i> "Data Admin berhasil ditambah" 2. Menampilkan halaman admin.	Sukses
	Salah	Mengisi <i>form</i> dengan data yang salah.	1. Menampilkan <i>alert</i> "The password field must be at least 6 character length". 2. Menampilkan <i>alert</i> "The name field is required" 3. Menampilkan <i>alert</i> "The username field is required." 4. Menampilkan <i>alert</i> "The password field is required." 5. Menampilkan <i>alert</i> "The email field is required." 6. Menampilkan <i>alert</i> "The username must contain a unique value." 7. Menampilkan <i>alert</i> "The email must contain unique value."	Sukses

Lampiran 9898 Hasil Pengujian Mengubah Akun *Admin*

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Mengubah Akun	Benar	Mengisi <i>form</i> dengan data yang benar.	1. Menampilkan <i>alert</i> "Data Admin berhasil diubah" 2. Menampilkan halaman admin.	Sukses
	Salah	Mengisi <i>form</i> dengan data yang salah.	1. Menampilkan <i>alert</i> "The password field must be at least 6 character length".	Sukses

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
			2. Menampilkan <i>alert</i> “The name field is required”	
			3. Menampilkan <i>alert</i> “The username field is required.”	
			4. Menampilkan <i>alert</i> “The password field is required.”	
			5. Menampilkan <i>alert</i> “The email field is required.”	
			6. Menampilkan <i>alert</i> “The username must contain a unique value.”	
			7. Menampilkan <i>alert</i> “The email must contain unique value.”	

Lampiran 9999 Hasil Pengujian Menghapus Akun

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Menhapus Akun	Benar	Memilih tombol hapus pada akun admin	1. Menampilkan <i>alert</i> “Apakah anda yakin ingin menghapus akun ini?” 2. Menampilkan <i>alert</i> “Akun admin berhasil dihapus”	Sukses

Lampiran 1000 Hasil Pengujian Menambah Data Anggota

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Menambah Data Anggota	Benar	Mengisi <i>form</i> dengan data yang benar.	Menampilkan <i>alert</i> “Data Anggota berhasil ditambah”	Sukses
	Salah	Mengisi <i>form</i> dengan data yang salah.	1. Menampilkan <i>alert</i> “Nama anggota perlu diisi” 2. Menampilkan <i>alert</i> “Nomor anggota perlu diisi”	Sukses

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
			3. Menampilkan <i>alert</i> “Nomor anggota tidak boleh sama”	

Lampiran 1011 Hasil Pengujian Mengubah Data Anggota

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Meng ubah Data	Benar	Mengisi <i>form</i> dengan data yang benar.	Menampilkan <i>alert</i> “Data Anggota berhasil diubah”	Sukses
Anggota	Salah	Mengisi <i>form</i> dengan data yang salah.	1. Menampilkan <i>alert</i> “Nama anggota harus diisi” 2. Menampilkan <i>alert</i> “Nomor anggota harus diisi” 3. Menampilkan <i>alert</i> “Nomor anggota tidak boleh sama”	Sukses

Lampiran 1022 Hasil Pengujian Menghapus Data Anggota

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Menhapus Anggota	Benar	Memilih tombol hapus pada data anggota	1. Menampilkan <i>alert</i> “Apakah anda yakin ingin menghapus data ini?” 2. Menampilkan <i>alert</i> “Data anggota berhasil dihapus”	Sukses

Lampiran 1033 Hasil Pengujian Menambah Data Barang

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Menambah Data	Benar	Mengisi <i>form</i> dengan data yang benar.	Menampilkan <i>alert</i> “Data Barang berhasil ditambah”	Sukses

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Barang	Salah	Mengisi <i>form</i> dengan data yang salah.	<ol style="list-style-type: none"> 1. Menampilkan <i>alert</i> “The nama <i>field</i> is <i>require</i>” 2. Menampilkan <i>alert</i> “The harga <i>field</i> is <i>require</i>” 3. Menampilkan <i>alert</i> “The kode barang <i>must contain unique value</i>” 	Sukses

Lampiran 1044 Hasil Pengujian Mengubah Data Barang

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Mengubah Data Barang	Benar	Mengisi <i>form</i> dengan data yang benar.	Menampilkan <i>alert</i> “Data barang berhasil diubah”	Sukses
g	Salah	Mengisi <i>form</i> dengan data yang salah.	<ol style="list-style-type: none"> 1. Menampilkan <i>alert</i> “The nama <i>field</i> is <i>require</i>” 2. Menampilkan <i>alert</i> “The harga <i>field</i> is <i>require</i>” 3. Menampilkan <i>alert</i> “The kode barang <i>must contain unique value</i>” 	Sukses

Lampiran 1055 Hasil Pengujian Menghapus Data Barang

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Menhapus Barang	Benar	Memilih tombol hapus pada data barang	<ol style="list-style-type: none"> 1. Menampilkan <i>alert</i> “Apakah anda yakin ingin menghapus data ini?” 2. Menampilkan <i>alert</i> “Data barang berhasil dihapus” 	Sukses

Lampiran 1066 Hasil Pengujian Menambah Data Pembelian

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Menambah Data Barang	Benar	Mengisi <i>form</i> dengan data yang benar.	Menampilkan <i>alert</i> “Data Barang berhasil ditambah”	Sukses
	Salah	Mengisi <i>form</i> dengan data yang salah.	<ol style="list-style-type: none"> 1. Menampilkan <i>alert</i> “The nama field is require” 2. Menampilkan <i>alert</i> “The harga field is require” 3. Menampilkan <i>alert</i> “The kode barang must contain unique value” 	Sukses

Lampiran 10707 Hasil Pengujian Mengubah Data Pembelian

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Mengubah Data Pembelian	Benar	Mengisi <i>form</i> dengan data yang benar.	Menampilkan <i>alert</i> “Data berhasil diubah”	Sukses
	Salah	Mengisi <i>form</i> dengan data yang salah.	<ol style="list-style-type: none"> 1. Menampilkan <i>alert</i> “The nama field is require” 2. Menampilkan <i>alert</i> “The barang field is require” 	Sukses

Lampiran 10808 Hasil Pengujian Menghapus Data Pembelian

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Menghapus Data Pembelian	Benar	Memilih tombol hapus pada data pembelian	<ol style="list-style-type: none"> 1. Menampilkan <i>alert</i> “Apakah anda yakin ingin menghapus data ini?” 2. Menampilkan <i>alert</i> “Data berhasil dihapus” 	Sukses

Lampiran 10909 Hasil Pengujian Melihat Data Laporan

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Melihat Data Laporan	Benar	Memilih tombol menu pelaporan	Menampilkan tabel data laporan yang telah tersimpan	Sukses

Lampiran 1100 Hasil Pengujian Mengunduh Data Laporan

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Mengunduh Data Laporan	Benar	Memilih tombol PDF atau Excel pada halaman laporan	Mengunduh data laporan sesuai format <i>file</i>	Sukses

Lampiran 1111 Hasil Pengujian Melihat Data Akun *Admin*

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Melihat Data Akun <i>Admin</i>	Benar	Memilih tombol menu akun	Menampilkan tabel data <i>admin</i> yang telah tersimpan	Sukses

Lampiran 1122 Hasil Pengujian Melihat Data Anggota

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Melihat Data Anggota	Benar	Memilih tombol menu anggota	Menampilkan tabel data anggota yang telah tersimpan	Sukses

Lampiran 1133 Hasil Pengujian Melihat Data Barang

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Melihat Data Barang	Benar	Memilih tombol menu barang	Menampilkan tabel data barang yang telah tersimpan	Sukses

Lampiran 1144 Hasil Pengujian Melihat Data Pembelian

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Melihat Data Pembelian	Benar	Memilih tombol menu pembelian	Menampilkan tabel data pembelian yang telah tersimpan	Sukses

Lampiran 1155 Hasil Pengujian Melihat Grafik Penjualan

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Melihat Grafik Penjualan	Benar	Memilih tombol menu beranda	Menampilkan grafik penjualan yang berdasarkan data penjualan	Sukses

Lampiran 11616 Hasil Pengujian Melihat Data Penjualan Barang Perbulan

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Melihat Data Penjualan Barang Perbulan	Benar	Memilih tombol menu beranda	Menampilkan data penjualan barang yang berdasarkan bulan yang dipilih	Sukses

Lampiran 11717 Hasil Pengujian Melihat Presentase Jenis Pembeli

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Melihat Presentasi Jenis Pembeli	Benar	Memilih tombol menu beranda	Menampilkan presentase penjualan barang anggota dan non-anggota.	Sukses

Lampiran 11818 Hasil Pengujian Melihat Detil Data Pembelian

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Melihat Detil Data Pembelian	Benar	Memilih tombol detail pada tabel data pembelian	Menampilkan nama pembeli, tanggal dan tiap barang yang dibeli beserta jumlahnya	Sukses

Lampiran 1199 Hasil Pengujian *Logout*

Fungsi	Masukan	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
<i>Logout</i>	Benar	Memilih tombol <i>logout</i> pada menu admin	Menampilkan halaman login	Sukses

RIWAYAT HIDUP

Penulis dilahirkan di kota Bogor pada tanggal 20 Oktober 2001 sebagai anak kedua dari Bapak Akhsani dan Ibu Yati. Pendidikan sekolah menengah atas (SMA) menempuh di MAN 2 Kota Bogor dan lulus pada tahun 2019. Penulis melanjutkan pendidikan sebagai mahasiswa program diploma (D-3) di Program Studi Manajemen Informatika Sekolah Vokasi IPB.

Semasa kuliah, penulis mengikuti kegiatan organisasi Badan Eksekutif Mahasiswa (BEM) Sekolah Vokasi IPB periode 2019-2020 menjadi bagian dari staff di Departemen Komunikasi dan Informasi. Penulis juga menjadi bagian dari tim *startup* Ambivers.id menjabat sebagai penanggung jawab divisi *graphic design* di tahun 2022. Penulis mendapatkan juara 3 pada perlombaan PITCOM Fest 2.0 kategori media pembelajaran pada tahun 2021.