BAB IV PERANCANGAN, PEMBUATAN, DAN PENGUJIAN APLIKASI

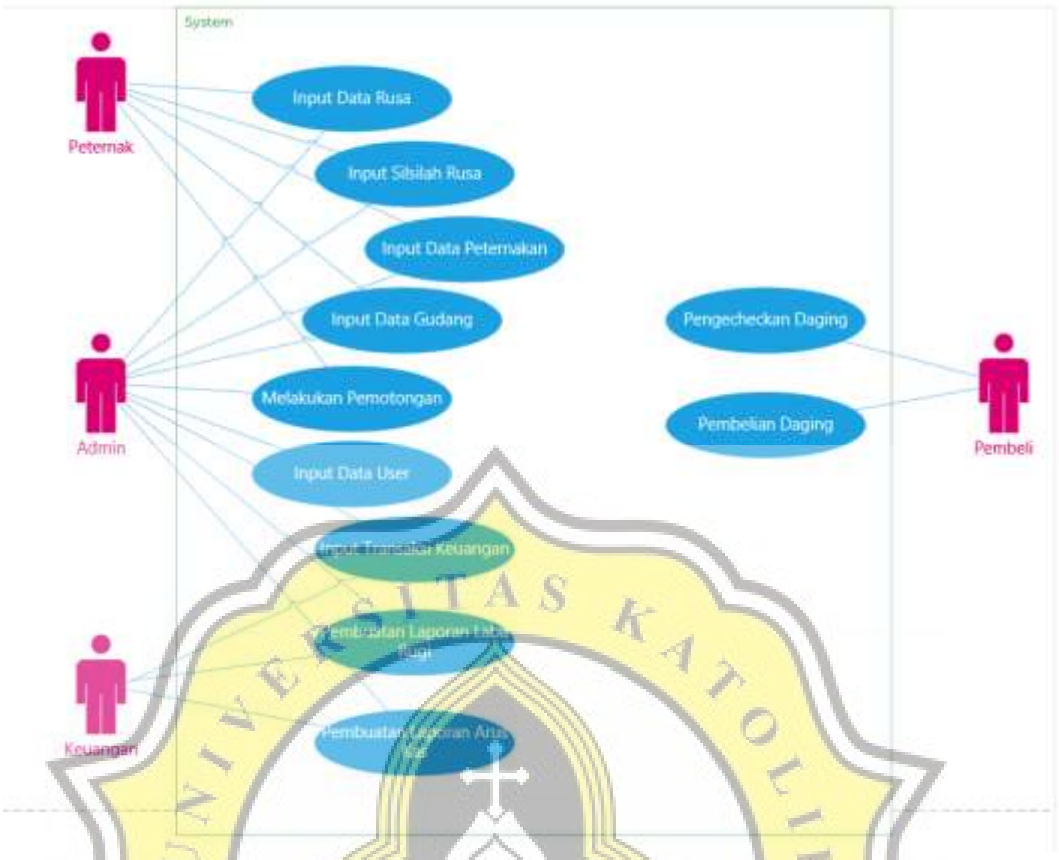
4.1. Perancangan Aplikasi

Aplikasi Sistem Recording Rusa memiliki tujuan pembuatan untuk pelacakan daging yang telah dijual kepada pembeli menggunakan impelementasi RFID (*Radio Frequency Identification)* dimana proses identifikasi daging dapat dipertanggung jawabkan asal usul daging tersebut dan identitas peranakan legal asal dari daging tersebut.

Aplikasi Sistem Recording Rusa sendiri berbasis web dan dapat diakses melalui domain yang telah disediakan. Dalam aplikasi Sistem Recording Rusa ini terdapat 2 Sistem implementasi RFID, yaitu terhadap rusa sebagai hewan penangkaran dan kepada daging sebagai pelacakan keaslian daging sendiri. Serta terdapat fitur pembelian, pengiriman daging serta laporan keuangan berupa arus kas dan laba rugi.

4.1.1. Perancangan Use Case Diagram

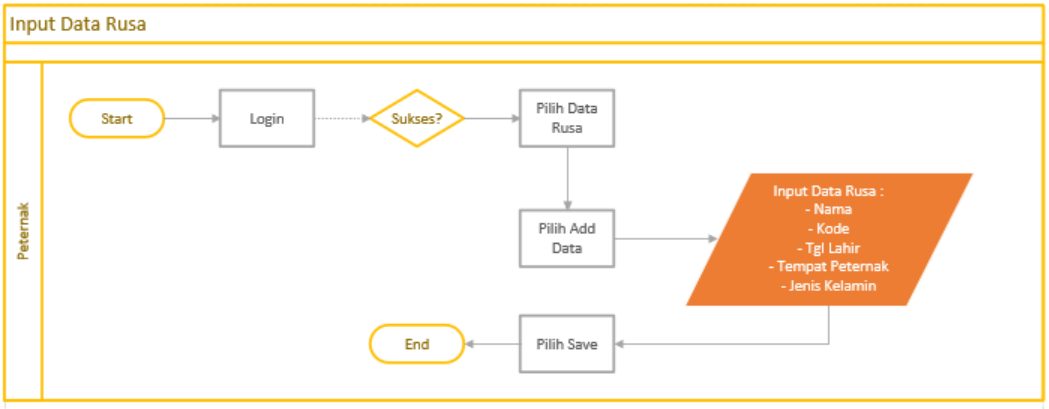
Pembuatan aplikasi dimulai dengan perancangan *use case diagram*. Supaya memberikan pemahaman akan tiap fungsi dan feature yang ada pada aplikasi, diagram dapat dilihat pada gambar 4.1. dibawah ini. Dari Use Case Diagram tersebut dapat dijelaskan bahwa penjual (peternak) mendapatkan fitur berupa input data peternakan, gudang, rusa, pemotongan rusa sementara administrasi keuangan mendapatkan fitur input transaksi, pembuatan laporan arus kas dan laba rugi dalam aplikasi serta Admin dimana dapat mengakses semua fitur tersebut ditambah pembuatan akun peternak, administrasi keuangan dan pembeli. Sementara disisi pembeli mendapatkan fitur pengecekan daging dan pembelian daging.



Gambar 4. 1. Use Case Diagram

4.1.2. Perancangan Flowchart

Setelah Pembuatan Use Case Diagram di ikuti dengan pembuatan flowchart dengan mengikuti case didalam Use Case Diagram. Gambar 4.2 berikut merupakan flowchart dalam proses input data rusa. Dari flowchart tersebut dapat dijelaskan bahwa proses diawali dengan peternak melakukan login, lalu memilih data rusa dan dilanjutkan memilih menambahkan data. Dari tahap tersebut, peternak diminta untuk memasukan data rusa berupa nam, kode, tanggal lahir, tempat peternakan serta jenis kelaminnya, selanjutnya memilih save maka data akan tersimpan dan proses selesai.



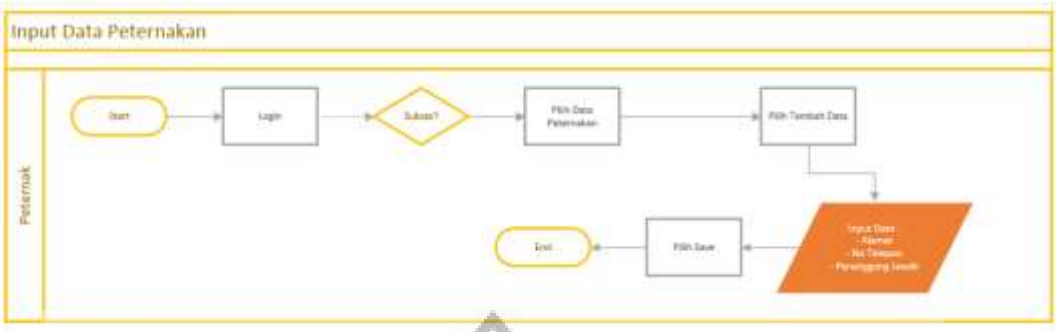
Gambar 4.2 Flowchart Input Data Rusa

Gambar 4.3 dibawah merupakan flowchart dalam proses input data silsilah bibit rusa. Dari flowchart tersebut dapat dijelaskan bahwa proses diawali dengan peternak melakukan login, lalu memilih data rusa dan dilanjutkan memilih rusa yang akan di input silsilahnya. Setelah memilih rusa, maka dilanjutkan dengan memilih tambah hubungan, selanjutnya peternak diminta untuk menginput data silsilah berupa perkawinan dan turunan/peranakan dilanjutkan dengan memilih save maka data akan tersimpan dan proses selesai.

Gambar 4.3 Flowchart Input Data Silsilah Rusa

Gambar 4.4 dibawah merupakan flowchart dalam proses input data peternakan. Dari flowchart tersebut dapat dijelaskan bahwa proses diawali dengan peternak melakukan login, lalu memilih data peternakan. Selanjutnya memilih tambah data, setelah itu peternak diminta untuk menginput data peternakan berupa alamat, no telepon dan penanggung

jawab. Setelah itu memilih save maka data akan tersimpan dan proses selesai.

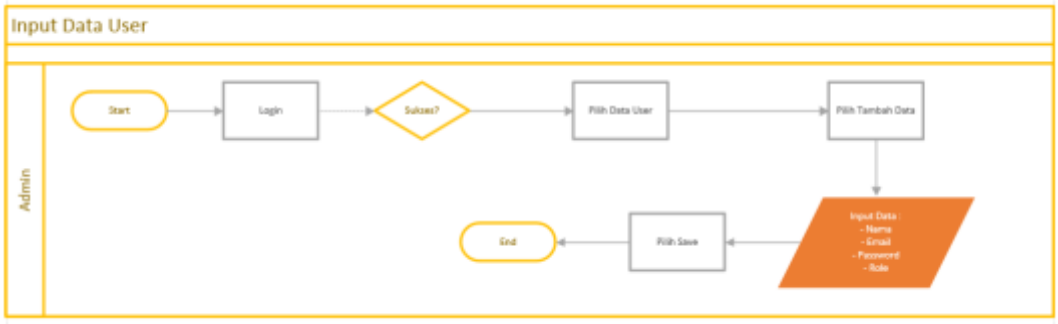


Gambar 4.4 Flowchart Input data Peternakan

Gambar 4.5 dibawah merupakan flowchart dalam proses input data gudang. Dari flowchart tersebut dapat dijelaskan bahwa proses diawali dengan peternak melakukan login, lalu memilih data gudang. Selanjutnya memilih tambah data, setelah itu peternak diminta untuk menginput data gudang berupa nama dan dimana peternakan berada. Setelah itu memilih save maka data akan tersimpan dan proses selesai.

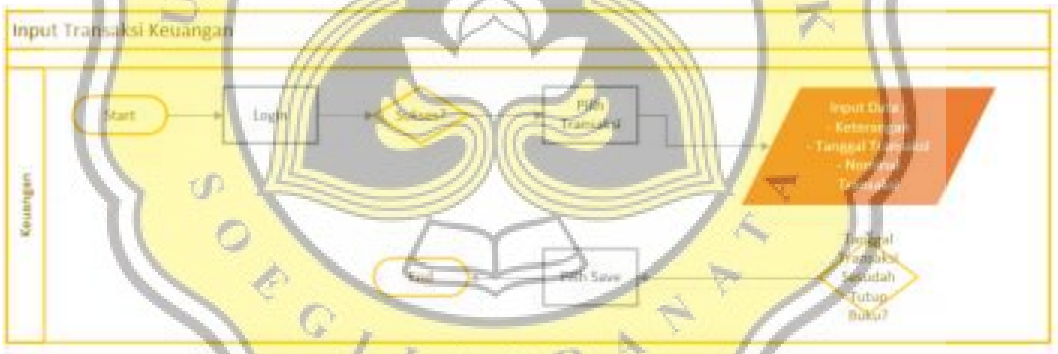
Gambar 4.5 Flowchart input data Gudang

Gambar 4.6 dibawah merupakan flowchart dalam proses input data user. Dari flowchart tersebut dapat dijelaskan bahwa proses diawali dengan admin melakukan login, lalu memilih data user. Selanjutnya memilih tambah data, setelah itu admin diminta untuk menginput data user berupa nama, email, password dan role dari user tersebut. Setelah itu memilih save maka data akan tersimpan dan proses selesai.



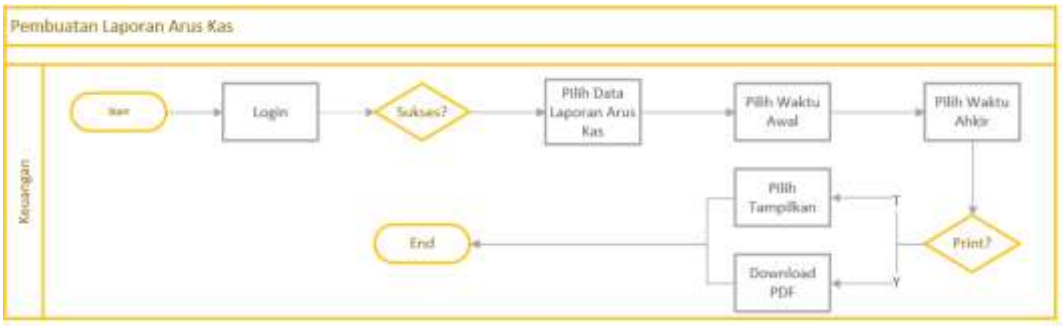
Gambar 4.6 Flowchart input data user

Gambar 4.7 dibawah merupakan flowchart dalam proses input data transaksi. Dari flowchart tersebut dapat dijelaskan bahwa proses diawali dengan keuangan melakukan login, lalu memilih transaksi. Selanjutnya keuangan diminta untuk menginput data transaksi berupa keterangan, tanggal transaksi dan nominal transaksi. Lalu akan dilakukan pengecheckan apakah tanggal transaksi sesudah tutup buku, jika sukses maka data akan tersimpan dan proses selesai.



Gambar 4.7 Flowchart input data transaksi

Gambar 4.8 dibawah merupakan flowchart dalam proses pembuatan laporan arus kas. Dari flowchart tersebut dapat dijelaskan bahwa proses diawali dengan keuangan melakukan login, lalu memilih laporan arus kas, selanjutnya memilih waktu awal laporan dilanjut dengan waktu ahkir laporan. Selanjutnya jika memilih print maka data akan terdownload dengan format pdf, jika tidak maka data akan ditampilkan saja.

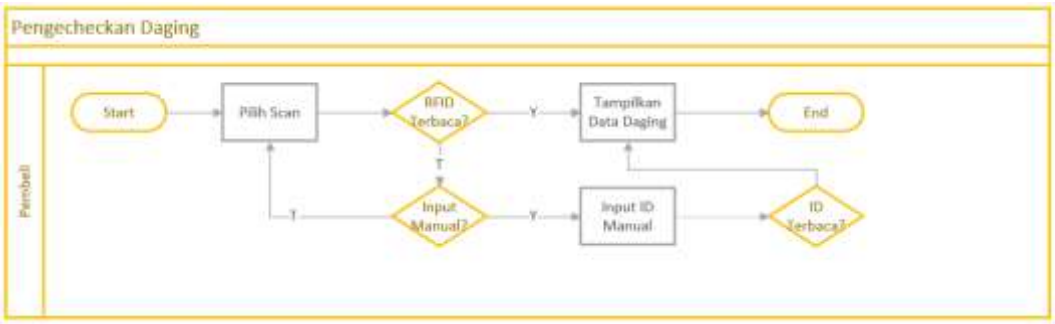


Gambar 4.8 Flowchart Pembuatan Laporan Arus Kas

Gambar 4.9 dibawah merupakan flowchart dalam proses pembuatan laporan laba rugi. Dari flowchart tersebut dapat dijelaskan bahwa proses diawali dengan keuangan melakukan login, lalu memilih laporan laba rugi, selanjutnya memilih waktu awal laporan dilanjut dengan waktu ahkir laporan. Selanjutnya jika memilih print maka data akan terdownload dengan format pdf, jika tidak maka data akan ditampilkan saja.

Gambar 4.9 Flowchart Pembuatan Laporan Laba Rugi

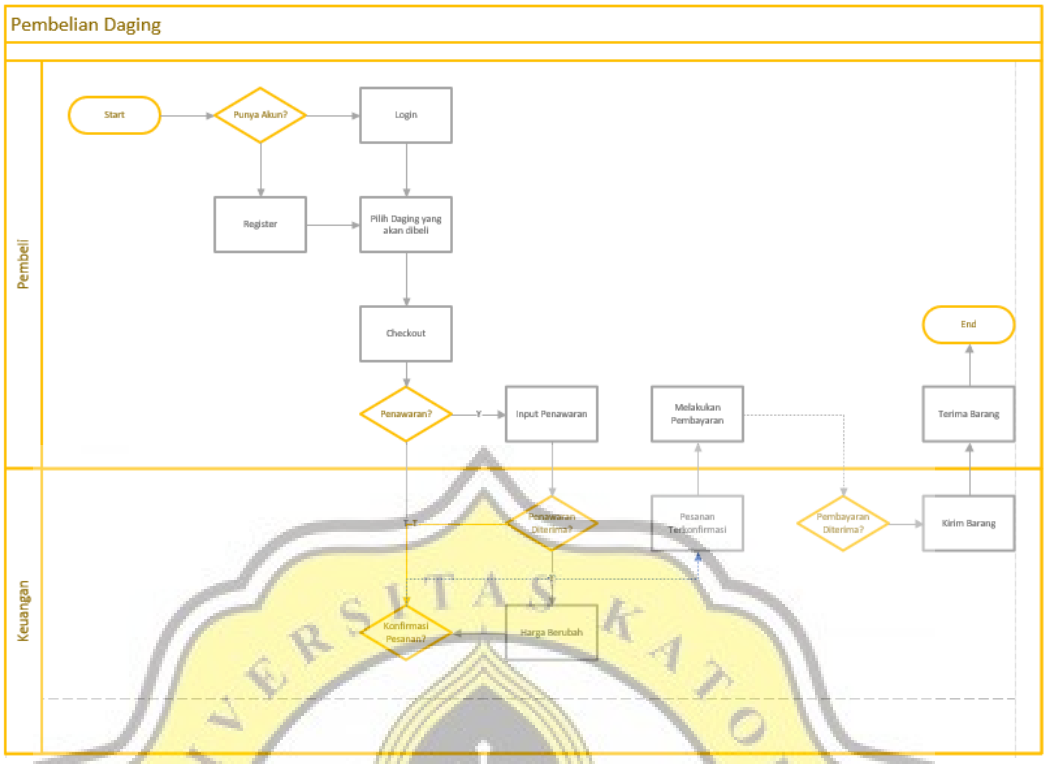
Gambar 4.10 dibawah merupakan flowchart dalam proses pengecheckan daging. Dari flowchart tersebut dapat dijelaskan bahwa proses diawali dengan pembeli memilih scan, jika scan rfid terbaca maka akan menampilkan data daging, sebaliknya jika scan tidak terbaca pembeli akan memilih apakah ingin menginput manual atau tidak, jika pembeli memilih input manual maka akan dilanjutkan dengan penginputan id secara manual serta melakukan pengecheckan kembali dari id tersebut hingga id terbaca dengan benar dan menampilkan data daging.



Gambar 4.10 Flowchart Proses Pengecheckan Daging

Gambar 4.11 dibawah merupakan flowchart dalam proses pembelian daging. Dari flowchart tersebut dapat dijelaskan bahwa proses pembelian daging diawali dengan pembeli melakukan login jika mempunyai akun atau register jika belum mempunyai akun. Dilanjutkan dengan pembeli memilih daging yang akan dibeli, setelahnya pembeli melakukan checkout. Jika pembeli melakukan penawaran maka dilanjutkan dengan menginput penawaran dan menunggu keuangan mengkonfirmasi penawaran tersebut, jika tidak maka akan langsung mengkonfirmasi pesanan tanpa perubahan harga, sementara jika penawaran diterima dan dilanjut ke perubahan harga dan konfirmasi pesanan. Jika pembeli tidak memilih penawaran maka akan berlanjut langsung pada konfirmasi pesanan oleh keuangan.

Setelah pesanan terkonfirmasi maka dilanjutkan dengan pembeli melakukan pembayaran serta berlanjut pada konfirmasi pembayaran oleh keuangan, jika telah terkonfirmasi maka dilanjut dengan pengiriman barang sampai barang diterima oleh pembeli.



Gambar 4.11 Flowchart Proses Pembelian Daging

4.1.3. Normalisasi Data

Setelah flowchart terpenuhi, maka dilanjutkan dengan normalisasi data yang ada, sehingga menghasilkan struktur table yang normal atau yang baik. Dibawah berikut merupakan gambar 4.12 yang berisi data awal yang belum terstruktur atau tidak normal. Dapat dilihat dari gambar tersebut, bentuk data masih belum terstruktur dan belum normal.

Data Rusa

Kode	Nama Rusa	Peternakan	Tgl Lahir	Kelamin	Status Kaw	Hubungan	Kode	Nama	Lahir	Kelamin	Keteranga
KD-6281	Lani	Kudus	27/02/2017	Betina	Sudah	Kawin	KD-1729	Breno	12/11/2016	Jantan	Telah kaw
						Anak	KS-1248	Ayu	1/10/2017	Betina	Sebagai Ai
						Anak	KS-3480	Marcelo	5/8/2018	Jantan	Sebagai Ai
KD-1729	Breno	Kudus	12/11/2016	Jantan	Sudah	Kawin	KD-6281	Lani	27/02/2017	Betina	Telah kaw
						Anak	KS-1248	Ayu	1/10/2017	Betina	Sebagai Ai
						Anak	KS-3480	Marcelo	5/8/2018	Jantan	Sebagai Ai
						Kawin	KS-1248	Aty	1/10/2017	Betina	Telah kaw

Data Transaksi

Tanggal	Keterangan	Nominal	KodePos	Pos	Daging	Berat	NominalDaging	Customer	Alamat	Kota	Tipe Trans	Dari
					Flank	5340 gr	250000					
10/10/2020	n Daging oleh bapa	400000	PJL	jualan Dag	Ribs	2421 gr	200000	Pandawa	ndung Raya	Bandung	Masuk	Bank BCA
					Neck	3520 gr	250000					
					Ribs	2421 gr	200000					
8/10/2020	n Daging oleh bapa	400000	PJL	jualan Dag	Leg	520 gr	50000	amsudew	Semarang	Semarang	Masuk	s Sekretar

Dari kedua data tersebut dilakukan normalisasi data menuju ke level 1nf. Bentuk 1nf pada data tersebut dapat dilihat pada gambar 4.13 berikut. Dimana dari gambar tersebut dapat dilihat bahwa setiap atribut dari data tersebut hanya memiliki nilai tunggal dalam satu baris, sehingga memenuhi persyaratan level 1nf.

Kode	Nama Rus	Peternaka	Tgl Lahir	Kelamin	Status Kav	Hubungan	Kode	Nama	Lahir	Kelamin	Keteranga
KD-6281	Lani	Kudus	27/02/2017	Betina	Sudah	Kawin	KD-1729	Breno	12/11/2016	Jantan	Telah kaw
KD-6282	Lani	Kudus	27/02/2018	Betina	Sudah	Anak	KS-1248	Ayu	1/10/2017	Betina	Sebagai Ai
KD-6283	Lani	Kudus	27/02/2019	Betina	Sudah	Anak	KS-3480	Marcelo	5/8/2018	Jantan	Sebagai Ai
KD-1729	Breno	Kudus	12/11/2016	Jantan	Sudah	Kawin	KD-6281	Lani	27/02/2017	Betina	Telah kaw
KD-1730	Breno	Kudus	12/12/2016	Jantan	Sudah	Anak	KS-1248	Ayu	1/10/2017	Betina	Sebagai Ai
KD-1731	Breno	Kudus	12/13/2016	Jantan	Sudah	Anak	KS-3480	Marcelo	5/8/2018	Jantan	Sebagai Ai
KD-1732	Breno	Kudus	12/14/2016	Jantan	Sudah	Kawin	KS-1248	Aty	1/10/2017	Betina	Telah kaw

Tanggal	Keteranga	Nominal	KodePos	Pos	Daging	Berat	NominalD	Customer	Alamat	Kota	Tipe Trans	Dari
10/10/2020	Pemesana	400000	PJL	Penjualan	Flank	5340 gr	250000	Pandawa	Jln Bandung R	Bandung	Masuk	Bank BCA
10/10/2020	Pemesana	400000	PJL	Penjualan	Ribs	2421 gr	200000	Pandawa	Jln Bandung R	Bandung	Masuk	Bank BCA
8/10/2020	Pemesana	500000	PJL	Penjualan	Neck	3520 gr	250000	Samsudev	Semarang	Semarang	Masuk	Kas Sekret
8/10/2020	Pemesana	500000	PJL	Penjualan	Ribs	2421 gr	200000	Samsudev	Semarang	Semarang	Masuk	Kas Sekret
8/10/2020	Pemesana	500000	PJL	Penjualan	Leg	520 gr	50000	Samsudev	Semarang	Semarang	Masuk	Kas Sekret

Gambar 4.13 Normalisasi Level 1nf

Dari tabel data tersebut masih dapat dilakukan normalisasi data menuju ke level 2nf. Bentuk 2nf pada data tersebut dapat dilihat pada gambar 4.14 berikut. Dimana dari gambar tersebut dapat dilihat bahwa sudah tidak adanya partial "functional dependency" kepada primary key dalam sebuah tabel, sehingga memenuhi persyaratan level 2nf.

rusa

Id	Kode	Nama Rusa	peternakan_i	Tgl Lahir	Kelamin
1	KD-6281	Lani	1	27/02/2017	Betina
2	KD-1729	Breno	1	12/11/2016	Jantan
3	KS-1248	Ayu	1	1/10/2017	Betina
4	KS-3480	Marcelo	1	5/8/2018	Jantan

peternakan

id	nama
1	kudus

hubungan_rusa				
id	rusa_id	rusa_id_to	hubungan	keterangan_hubu
1	1	2	Kawin	Telah Kawin
2	1	3	Anak	Mempunyai Anak
3	1	4	Anak	Mempunyai Anak
4	2	1	Kawin	Telah Kawin
5	2	3	Kawin	Telah Kawin
6	2	3	Anak	Mempunyai Anak
7	2	4	Anak	Mempunyai Anak

id	Tanggal	Keterangan	Nominal	KodePos	Pos	Tipe Trans	balance_i
1	10/10/2020	Pemesanan Da	400000	PJL	Penjualan	Masuk	1
2	8/10/2020	Pemesanan Da	500000	PJL	Penjualan	Masuk	2
3	10/10/2020	Pengeluaran B	100000	BBM	Bensin	Keluar	2
4	7/10/2020	Pengeluaran B	200000	VIT	Vitamin	Keluar	2

id	Balance
1	Bank BCA
2	Kas Sekretariat

id	transaksi_id	Daging	Berat	NominalD
1	1	Flank	5340 gr	250000
2	1	Ribs	2421 gr	200000
3	2	Neck	3520 gr	250000
4	2	Ribs	2421 gr	200000
5	2	Leg	520 gr	50000

Gambar 4.14 Normalisasi Level 2nf

Dari tabel data tersebut masih dapat dilakukan normalisasi data menuju ke level 3nf. Bentuk 3nf pada data tersebut dapat dilihat pada gambar 4.15 berikut. Dimana dari gambar tersebut dapat dilihat bahwa sudah tidak adanya partial "transitive dependency" atau adanya suatu attribute yang tidak bergantung pada primary key tapi bergantung pada field lain maka attribute tersebut perlu dipisah ke tabel baru, sehingga memenuhi persyaratan level 3nf. Dari tabel dibawah field keterangan_hubungan tidak bergantung pada primary key dari tabel sebelumnya, maka setelah di normalisasi 3nf akan jadi seperti ini.

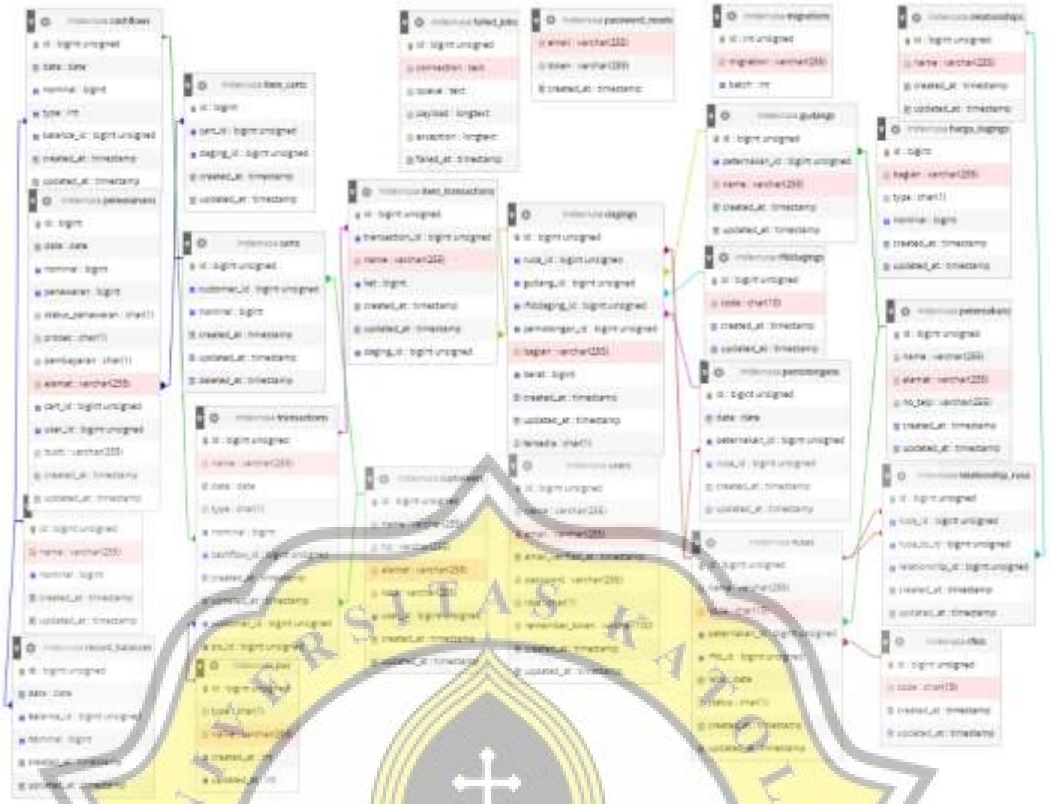
hubungan_rusa			
id	rusa_id	rusa_id_t	hubungan
1	1	2	Kawin
2	1	3	Anak
3	1	4	Anak
4	2	1	Kawin
5	2	3	Kawin
6	2	3	Anak
7	2	4	Anak

hubungan		
id	hubungan	keteranga
1	Kawin	Telah Kaw
2	Anak	Mempuny

Gambar 4.15 Normalisasi Level 3nf

4.1.4. Perancangan ERD (Entity Relationship Diagram)

Setelah Use Case Diagram dan flowchart dibuat dibutuhkan perancangan diagram relasi dengan tujuan untuk menjelaskan hubungan tiap entitas data yang terdapat didalam aplikasi. Hubungan tiap table data didalam database berupa *many-to-many (*menggunakan pivot table*), one-to-many*, dan *one-to-one*. Berikut diagram relasi pada gambar 4.2 dibawah ini.



Gambar 4. 2. ERD Diagram

Gambar 4.3. adalah detail kolom dari table users.

#	Name	Type
1	id	bigint
2	name	varchar(255)
3	email	varchar(255)
4	email_verified_at	timestamp
5	password	varchar(255)
6	role	char(1)
7	remember_token	varchar(100)
8	created_at	timestamp
9	updated_at	timestamp

Gambar 4. 3. Detail table users

Gambar 4.4. adalah detail kolom dari table customers. Dapat diperhatikan di table customer memiliki *relationship one-to-one* kepada table users, dimana jika users tersebut berupa customer dia akan memiliki relasi

tersebut untuk menyimpan data customer, jika users tersebut berupa admin/penjual maka dia tidak memiliki relasi tersebut.

#	Name	Type
1	id	bigint
2	name	varchar(255)
3	hp	varchar(255)
4	alamat	varchar(255)
5	kota	varchar(255)
6	user_id	bigint
7	created_at	timestamp
8	updated_at	timestamp

Gambar 4. 4. Detail table customers

Gambar 4.5. adalah detail kolom dari table peternakans.

#	Name	Type
1	id	bigint
2	name	varchar(255)
3	alamat	varchar(255)
4	no_telp	varchar(255)
5	created_at	timestamp
6	updated_at	timestamp

Gambar 4. 5. Detail table peternakans

Gambar 4.6. adalah detail kolom dari table rusa. Dapat dilihat bahwa table rusa dan peternakan memiliki relasi *one to many* dimana dalam peternakan memiliki banyak rusa didalamnya.

#	Name	Type
1	id	bigint
2	name	varchar(255)
3	kode	char(10)
4	peternakan_id	bigint
5	rfid_id	bigint
6	lahir	date
7	status	char(1)
8	created_at	timestamp
9	updated_at	timestamp

Gambar 4. 6. Detail table rusa

Gambar 4.7. adalah detail kolom dari table pemotongan. Dapat dilihat pada table pemotongan memiliki ketergantungan relasi *many to one* terhadap table peternakan dan *one to one* pada table rusa. Dimana data peternakan memiliki banyak data pemotongan dan setiap pemotongan memiliki data rusa.

#	Name	Type
1	id	bigint
2	date	date
3	peternakan_id	bigint
4	rusa_id	bigint
5	created_at	timestamp
6	updated_at	timestamp

Gambar 4. 7. Detail table pemotongan

Gambar 4.8. adalah detail dari table gudangs. Dapat dilihat pada table gudangs terdapat relasi *many to one* kepada table peternakan. Dimana data Gudang tergantung pada data peternakan yang ada.

#	Name	Type
1	id	bigint
2	peternakan_id	bigint
3	name	varchar(255)
4	created_at	timestamp
5	updated_at	timestamp

Gambar 4. 8. Detail table gudang

Gambar 4.9. adalah detail dari table rfiddaging.

#	Name	Type
1	id	bigint
2	code	char(10)
3	created_at	timestamp
4	updated_at	timestamp

Gambar 4. 9. Detail table rfiddaging

Gambar 4.10. adalah detail dari table daging. Dapat dilihat pada diagram ERD table daging memiliki relasi *many to one* terhadap data table rusa, *one to one* kepada table gudang, *one to one* kepada table rfiddaging, *many to one* terhadap table pemotongan.

#	Name	Type
1	id	bigint
2	rusa_id	bigint
3	gudang_id	bigint
4	rfiddaging_id	bigint
5	pemotongan_id	bigint
6	bagian	varchar(255)
7	berat	bigint
8	created_at	timestamp
9	updated_at	timestamp
10	tersedia	char(1)

Gambar 4. 10. Detail table daging

Gambar 4.11. adalah detail dari table cart.

#	Name	Type
1	id	bigint
2	customer_id	bigint
3	nominal	bigint
4	created_at	timestamp
5	updated_at	timestamp
6	deleted_at	timestamp

Gambar 4. 11. Detail table cart

Gambar 4.12. adalah detail dari table item_cart. Dapat dilihat pada diagram ERD bahwa table item_cart memiliki relasi *many to one* terhadap table cart.

#	Name	Type
1	id	bigint
2	cart_id	bigint
3	daging_id	bigint
4	created_at	timestamp
5	updated_at	timestamp

Gambar 4. 12. Detail table item_cart

Gambar 4.13. adalah detail dari table pemesanan. Dapat dilihat pada diagram ERD bahwa table pemesanan memiliki relasi *many to one* terhadap table user dan *one to one* pada table cart.

#	Name	Type
1	id	bigint
2	date	date
3	nominal	bigint
4	penawaran	bigint
5	status_penawaran	char(1)
6	proses	char(1)
7	pembayaran	char(1)
8	alamat	varchar(255)
9	cart_id	bigint
10	user_id	bigint
11	bukti	varchar(255)
12	created_at	timestamp
13	updated_at	timestamp

Gambar 4. 13. Detail table pemesanans

Gambar 4.14. adalah detail dari table transaction. Dapat dilihat pada diagram ERD bahwa table transaction memiliki relasi *many to one* terhadap table customer dan *one to many* pada table pos.

#	Name	Type
1	id	bigint
2	name	varchar(255)
3	date	date
4	type	char(1)
5	nominal	bigint
6	cashflow_id	bigint
7	created_at	timestamp
8	updated_at	timestamp
9	customer_id	bigint
10	po_id	bigint

Gambar 4. 14. Detail table transaction

Gambar 4.15. adalah detail dari table balance.

#	Name	Type
1	id	bigint
2	name	varchar(255)
3	nominal	bigint
4	created_at	timestamp
5	updated_at	timestamp

Gambar 4. 15. Detail table balance

Gambar 4.16. adalah detail dari table record_balance. Dapat dilihat pada table record_balance memiliki relasi *many to one* terhadap table balance. Dimana fungsi dari table record_balance ini untuk mengambil data balance pada bulan tertentu dalam laporan laba rugi dan arus kas.

#	Name	Type
1	id	bigint
2	date	date
3	balance_id	bigint
4	nominal	bigint
5	created_at	timestamp
6	updated_at	timestamp

Gambar 4. 16. Detail table record_balance

4.1.5. Perancangan Design Aplikasi

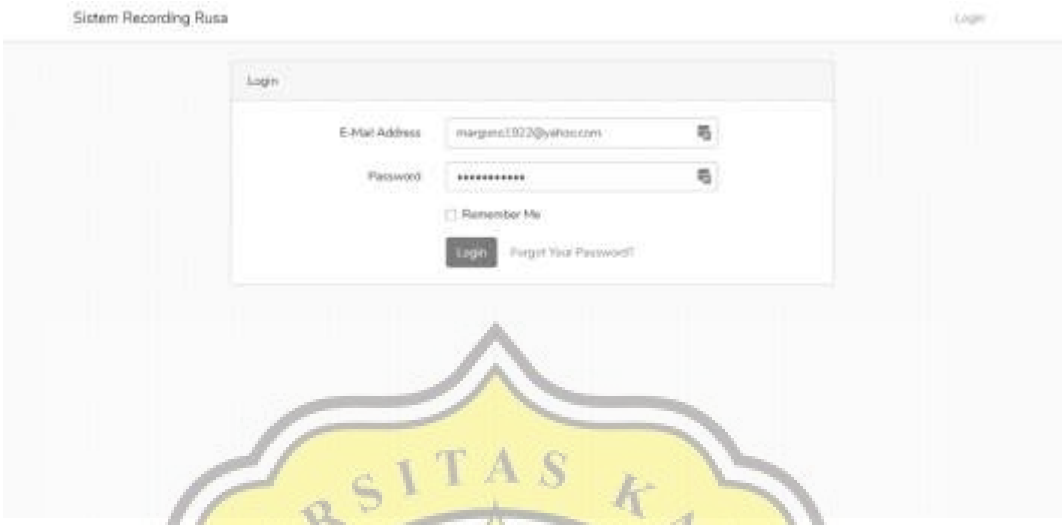
Gambar 4.18. dibawah berikut merupakan gambaran design aplikasi waktu awal diakses, pembeli dapat menclick gambar dan melakukan scan RFID.

Gambar 4. 17. Rancangan tampilan awal aplikasi

Gambar 4.19. dibawah berikut merupakan rancangan design aplikasi setelah pembeli melakukan scan RFID, data yang dikeluarkan berupa data rusa daging yang dipotong dan persebaran dagingnya.

Gambar 4. 18. Rancangan Tampilan User melakukan Scan Daging

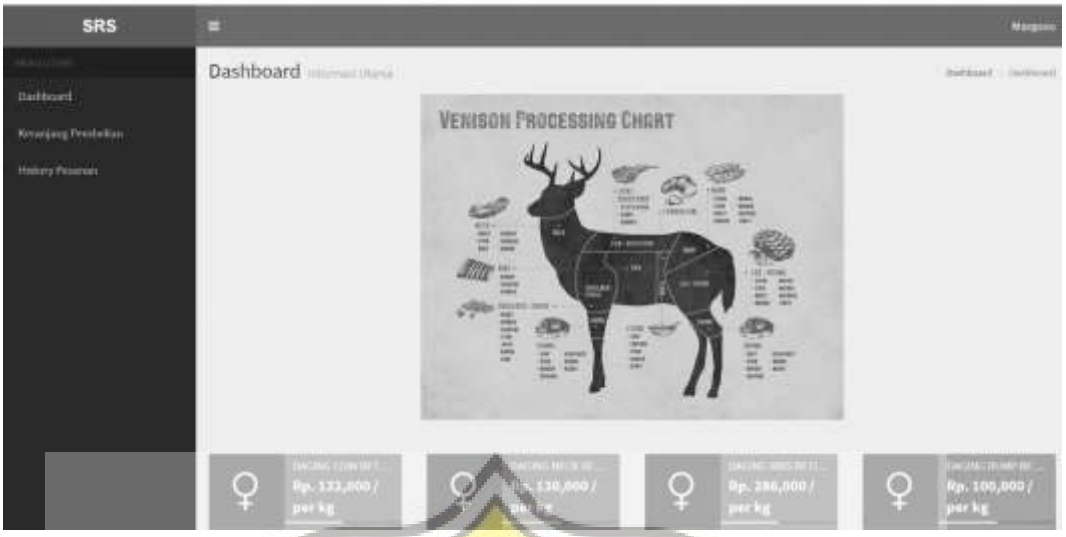
Gambar 4.20. dan 4.21. berikut merupakan rancangan design aplikasi Ketika pembeli melakukan registrasi dan login ke aplikasi.



Gambar 4. 19. Rancangan tampilan Login

Gambar 4. 20. Rancangan tampilan Registrasi

Gambar 4.22. berikut merupakan rancangan design aplikasi setelah pembeli melakukan login atau registrasi kepada aplikasi. Data ditampilkan berupa daging yang tersedia untuk dijual.

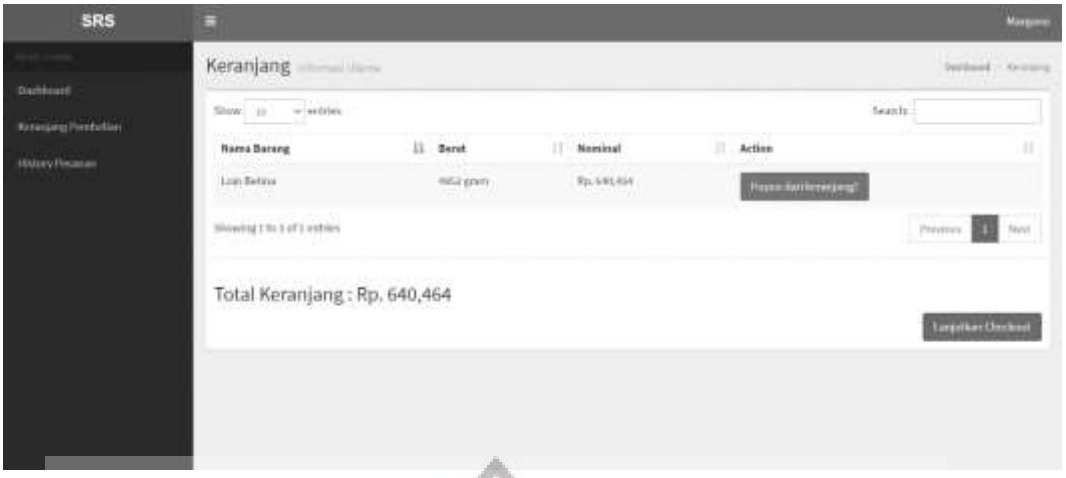


Gambar 4. 21. Rancangan tampilan awal setelah login

Gambar 4.23. berikut merupakan rancangan design ketika pembeli memilih salah satu jenis daging yang ingin dibeli, pembeli diarahkan dan diberikan info mengenai berat daging dan harga yang diberikan oleh penjual.

Gambar 4. 22. Rancangan tampilan ketika pembeli memilih daging

Gambar 4.24. berikut merupakan rancangan tampilan ketika pembeli memilih daging yang ada, serta memasukannya kedalam keranjang pembelian.

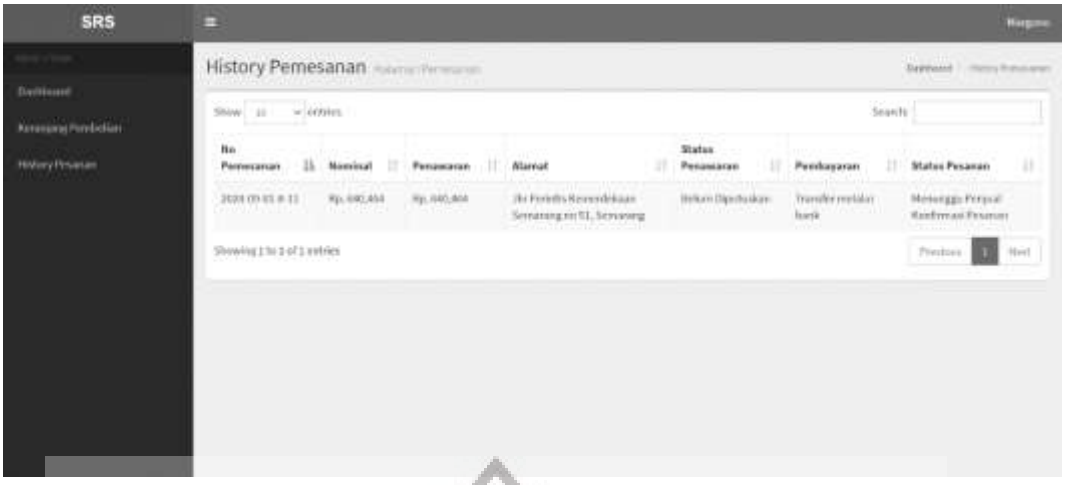


Gambar 4. 23. Rancangan tampilan keranjang pembelian

Gambar 4.25. berikut merupakan rancangan tampilan ketika pembeli melakukan checkout keranjang pembelian dan melakukan penawaran serta memilih cara pembayaran yang ada. Penawaran akan otomatis ditolak, jika penjual tidak merespon penawaran dari pembeli lebih dari 24 jam setelah penawaran dibuat.

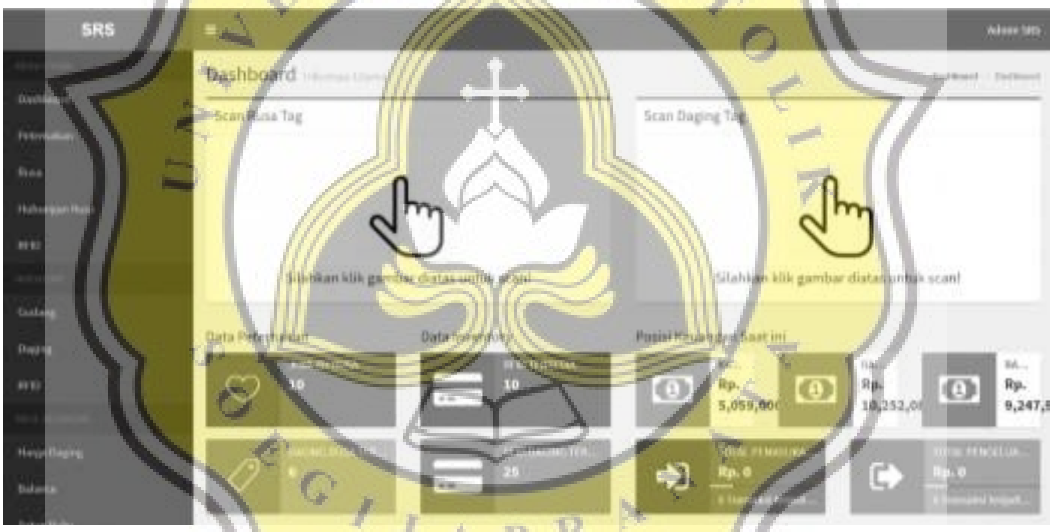
Gambar 4. 24. Rancangan tampilan checkout

Gambar 4.26. berikut merupakan rancangan tampilan history pemesanan pembeli dan pengecekan status atau proses pemesanan pada aplikasi.



Gambar 4. 25. Rancangan tampilan history pemesanan

Gambar 4.27. berikut merupakan rancangan tampilan pertama kali penjual login kedalam aplikasi.

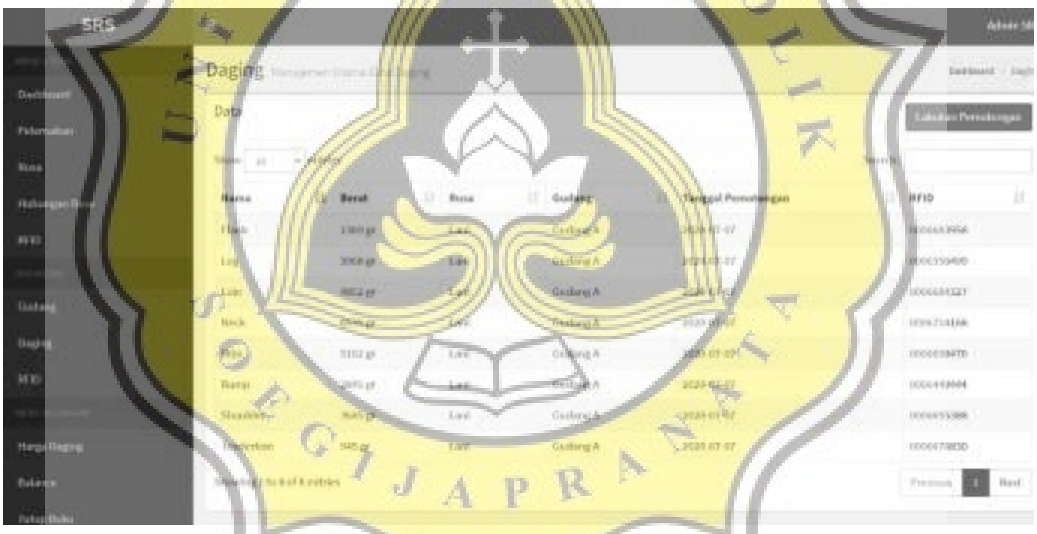


Gambar 4. 26. Rancangan tampilan dashboard penjual

Gambar 4.28. berikut merupakan rancangan tampilan pada penjual melakukan pemotongan daging, serta registrasi RFID pada daging.

Gambar 4. 27. Rancangan tampilan ketika pemotongan

Gambar 4.29. berikut merupakan rancangan tampilan pada inventory daging penjual.

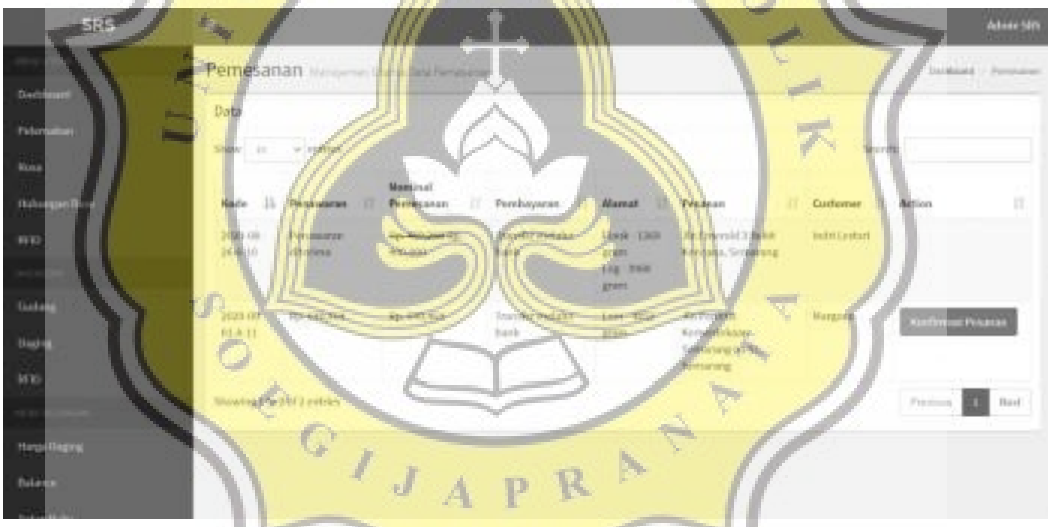


Gambar 4. 28. Rancangan tampilan inventory

Gambar 4.30. berikut merupakan rancangan tampilan recording atau pengaturan hubungan silsilah rusa pada penjual.

Gambar 4. 29. Rancangan tampilan recording rusa

Gambar 4.31. berikut merupakan rancangan tampilan history pemesanan dalam tampilan penjual.

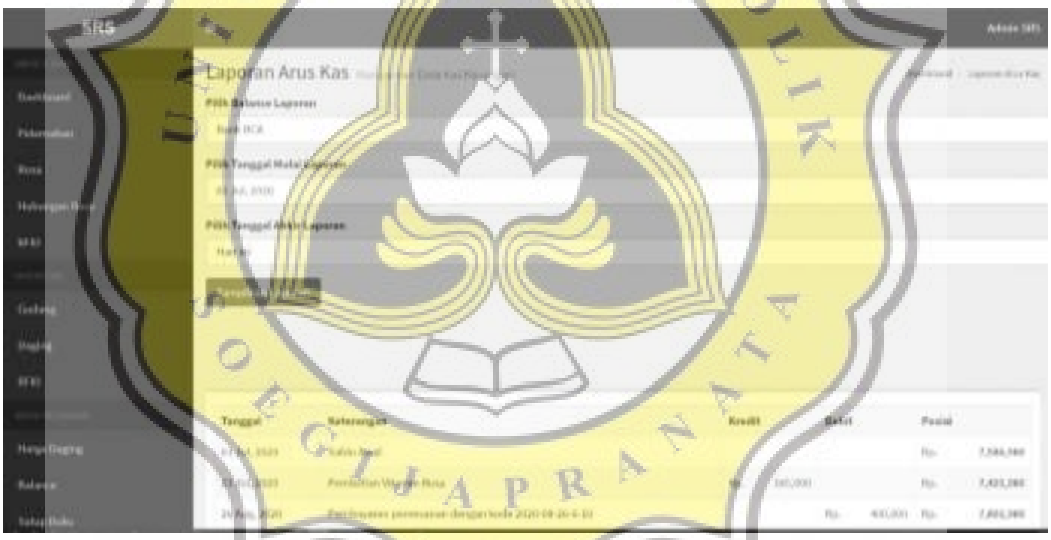


Gambar 4. 30. Rancangan tampilan pemesanan

Gambar 4.32. berikut merupakan rancangan tampilan saat input transaksi pengeluaran dalam tampilan penjual.

Gambar 4. 31. Rancangan tampilan transaksi

Gambar 4.33. berikut merupakan rancangan tampilan saat memilih laporan arus kas oleh penjual.



Gambar 4. 32. Rancangan tampilan laporan arus kas

Gambar 4.34. berikut merupakan rancangan tampilan saat memilih laporan laba rugi oleh penjual.

Gambar 4. 33. Rancangan tampilan laporan laba rugi

4.2. Pembuatan Aplikasi

Aplikasi dibuat menggunakan *framework* Laravel 7, dengan konsep MVC (model, view, controller) yaitu pembagian aplikasi dibagi menjadi 3 bagian model bagian yang berfungsi mengelola data pada aplikasi, view berfungsi mengelola tampilan data sehingga informasi mudah dipahami, serta controller sendiri berfungsi menjembatani antara model dan view.

4.2.1. Database

Menyiapkan database merupakan langkah pertama dalam pembuatan aplikasi. Database yang dipakai ialah MySQL dengan program pengelolanya phpMyAdmin. Berikut gambar 4.35. yang mejelaskan daftar table yang digunakan didalam aplikasi.

balances	InnoDB	utf8mb4_unicode_ci	16.0 KiB
carts	InnoDB	utf8mb4_general_ci	32.0 KiB
cashflows	InnoDB	utf8mb4_unicode_ci	32.0 KiB
customers	InnoDB	utf8mb4_unicode_ci	32.0 KiB
dagings	InnoDB	utf8mb4_unicode_ci	80.0 KiB
gudangs	InnoDB	utf8mb4_unicode_ci	32.0 KiB
harga_dagings	InnoDB	utf8mb4_general_ci	16.0 KiB
item_carts	InnoDB	utf8mb4_general_ci	48.0 KiB
item_transactions	InnoDB	utf8mb4_unicode_ci	48.0 KiB
pemesanans	InnoDB	utf8mb4_general_ci	48.0 KiB
pemotongans	InnoDB	utf8mb4_unicode_ci	32.0 KiB
peternakans	InnoDB	utf8mb4_unicode_ci	16.0 KiB
pos	InnoDB	utf8mb4_general_ci	16.0 KiB
record_balances	InnoDB	utf8mb4_unicode_ci	32.0 KiB
relationships	InnoDB	utf8mb4_unicode_ci	16.0 KiB
relationship_rusa	InnoDB	utf8mb4_unicode_ci	48.0 KiB
rfiddagings	InnoDB	utf8mb4_unicode_ci	16.0 KiB
rfids	InnoDB	utf8mb4_unicode_ci	16.0 KiB
rusas	InnoDB	utf8mb4_unicode_ci	48.0 KiB
transactions	InnoDB	utf8mb4_unicode_ci	64.0 KiB
users	InnoDB	utf8mb4_unicode_ci	32.0 KiB

Gambar 4. 34. Daftar table

4.2.2. Tampilan awal aplikasi

Didalam tampilan awal aplikasi terdapat feature berupa pengecekan daging menggunakan rfid secara langsung oleh pembeli tanpa harus melakukan login ataupun registrasi. Gambar 4.36. berikut menjelaskan script pada controller serta gambar 4.37. menjelaskan script pada tampilan.

```
public function scan(Request $request){
        $rfid = $request->scan;

        $rfid = Rfiddaging::where('code',$rfid)->first();
        $daging = $rfid->daging->first();
        $daginglainnya = Daging::where('pemotongan_id',$daging->pemotongan_id)->get();
        $rusa = Rusa::find($daging->rusa_id);
        return view('scan')->with(compact('rusa','daging','daginglainnya'));
    }
```

Gambar 4.36. Script dalam controller

```
<!DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">

        <title>SRS SYSTEM</title>

        <!-- Fonts -->
        <link href="https://fonts.googleapis.com/css?family=Nunito:200,600" rel="styleshee
t">

        <!-- Styles -->
        <style>
            html, body {
                background-color: #fff;
                color: #636b6f;
                font-family: 'Nunito', sans-serif;
                font-weight: 200;
                height: 100vh;
                margin: 0;
            }

            .full-height {
                height: 100vh;
            }

            .flex-center {
                align-items: center;
                display: flex;
                justify-content: center;
            }

            .position-ref {
                position: relative;
            }

            .top-right {
                position: absolute;
                right: 10px;
                top: 18px;
            }

            .content {
                text-align: center;
            }

            .title {
                font-size: 84px;
            }

            .links > a {
                color: #636b6f;
                padding: 0 25px;
                font-size: 13px;
                font-weight: 600;
                letter-spacing: .1rem;
                text-decoration: none;
                text-transform: uppercase;
            }

            .m-b-md {
                margin-bottom: 30px;
            }
        </style>
    </head>
    <body>
        <div class="flex-center position-ref full-height">
```

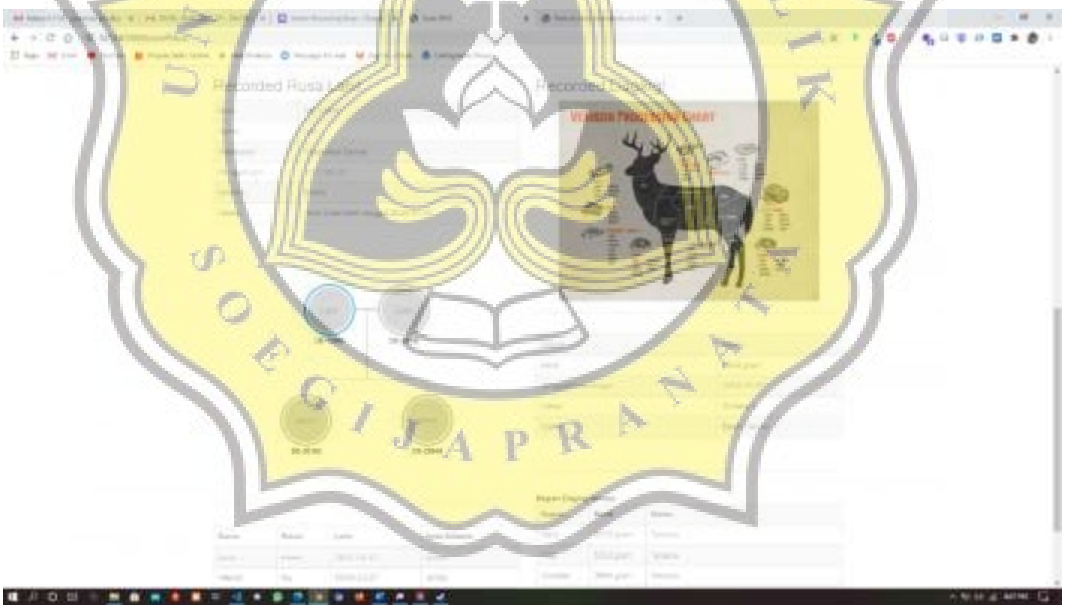
```
            @if (Route::has('login'))
                <div class="top-right links">
                    @auth
                        <a href="{{ url('/home') }}">Home</a>
                    @else
                        <a href="{{ route('login') }}">Login</a>
                        <a href="{{ route('register') }}">Register</a>
                    @endauth
                </div>
            @endif

            <div class="content">
                <h2>Sistem Pengecekan Daging RFID</h2>
                    <form action="{{route('scan')}}" method="post" style="opacity:0;">
                        {!! csrf_field() !!}
                        <input type="text" name="scan" id="rusa" onfocus="rusafocus()" onb
lur="rusablur()">
                        <button type="submit"></button>
                    </form>
                    <a href="#" id="rusarfid"><img id="gambarrusa" src="{{asset('tap.gif')
}}" style=" width:100%;"alt=""></a>
                    <h3><center id="rusa-
help">Silahkan klik gambar diatas untuk scan!</center></h3>
                    @if(@$rusa)
                    <div class="box box-success">
                        <div class="box-header with-border">
                            <h2 class="box-title">Recorded {{@$rusa->name}}!</h2>
                        </div>
                        <div class="box-body">
                            <table class="table table-striped table-bordered">
                                <tr>
                                    <td>Kode</td>
                                    <td>{{@$rusa->kode}}</td>
                                </tr>
                                <tr>
                                    <td>Name</td>
                                    <td>{{@$rusa->name}}</td>
                                </tr>
                                <tr>
                                    <td>Peternakan</td>
                                    <td>{{@$rusa->peternakan->name}}</td>
                                </tr>
                                <tr>
                                    <td>Tanggal Lahir</td>
                                    <td>{{@$rusa->lahir}}</td>
                                </tr>
                                <tr>
                                    <td>Kelamin</td>
                                    <td>@if(@$rusa-
>status) Jantan @else Betina @endif</td>
                                </tr>
                                <tr>
                                    <td>Status</td>
                                    <td>@if($potong = DB::table('pemotongans')-
>where('rusa_id',$rusa->id)->first()) Telah Disembelih tanggal {{@$potong-
>date}} @else Hidup @endif</td>
                                </tr>

                            </table>
                            <br/>
                            <hr/>
                                <iframe src="{{route('rusa.bagan', ['id' => @$rusa-
>id])}}" width="100%" height="400" frameborder="0" zoom="0.5"></iframe>
                            <hr/>
                            <br/>
                            <table class="table table-bordered table-striped">
                                <thead>
                                <tr>
                                <th>Nama</th>
                                <th>Relasi</th>
                                <th>Lahir</th>
```

```
                                <th>Jenis Kelamin</th>
                                </tr>
                                </thead>
                                <tbody>
                                    @foreach($rusa->relationship as $row)
                                        @php
                                            $rusa = DB::table('rusas')->where('id',$row-
>pivot->rusa_to_id)->first();
                                        @endphp
                                        <tr>
                                        <td>{{$rusa->name}}</td>
                                        <td>{{$row->name}}</td>
                                        <td>{{$rusa->lahir}}</td>
                                        <td>@if($rusa-
>status) Jantan @else Betina @endif</td>
                                        </tr>
                                    @endforeach
                                </tbody>
                            </table>
                        </div>
                    </div>
                    @endif
                </div>
            </div>
            <script src="{{asset('bower_components/jquery/dist/jquery.min.js')}}"></script>
            <script>
                $('#rusarfid').on('click',function(e){
                    $('#rusa').focus();
                });

                function rusafocus(){
                    $('#gambarrusa').attr('src','{{asset("sound.gif")}}');
                    $('#rusa-help').html('Silahkan tempelkan tag rfid!');
                }

                function rusablur(){
                    $('#gambarrusa').attr('src','{{asset("tap.gif")}}');
                    $('#rusa-help').html('Silahkan klik gambar diatas untuk scan!');
                }
            </script>
        </body>
</html>
```

Gambar 4.37. Script pada tampilan

Dengan melakukan scan rfid, maka pembeli dapat melacak asal usul daging, peredaran potongan daging lainnya, serta silsilah bibit daging pada penangkaran. Gambar 4.38 berikut merupakan bentuk jadi sebelum melakukan scan, sementara gambar 4.39 merupakan bentuk jadi setelah user melakukan scan pengecekan rfid daging pada sistem tampilan awal.

Gambar 4.38 Tampilan sebelum scan



Gambar 4.39. Tampilan setelah user melakukan scan

4.2.3. Registrasi dan Login

Pada login dan registrasi terdapat sejumlah fitur yang digunakan, ketika user melakukan registrasi maka password akan terenkripsi menggunakan *hash* feature pada Laravel, sehingga pencocokan password hanya dapat

dilakukan satu arah atau dapat dikatakan password tidak dapat di *decrypt* sehingga menambahkan fitur keamanan. Gambar 4.40. menunjukan proses registrasi dalam controller sementara gambar 4.41. menunjukan script tampilan registrasi. Gambar 4.42. merupakan tampilan registrasi yang dapat dilihat dari browser client.

```
public function __construct()
    {
        $this->middleware('guest');
    }

    protected function validator(array $data)
    {
        return Validator::make($data, [
            'name' => ['required', 'string', 'max:255'],
            'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
            'password' => ['required', 'string', 'min:8', 'confirmed'],
        ]);
    }

    protected function create(array $data)
    {
        $user = User::create([
            'name' => $data['name'],
            'email' => $data['email'],
            'role' => $data['role'],
            'password' => Hash::make($data['password']),
        ]);

        Customer::create([
            'name'=>$data['name'],
            'alamat' => $data['alamat'],
            'hp' => $data['nohp'],
            'kota' => $data['kota'],
            'user_id'=>$user->id,
        ]);

        return $user;
    }
```

Gambar 4.40. Script Registrasi

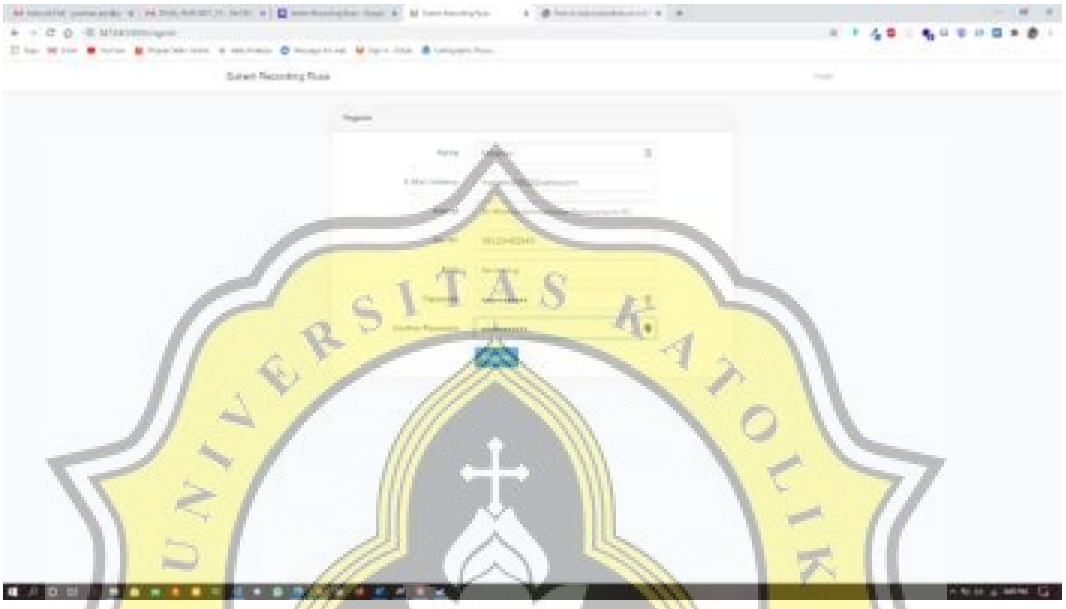
```
<div class="card">
    <div class="card-header">{{ __('Register') }}</div>

    <div class="card-body">
        <form method="POST" action="{{ route('register') }}">
            @csrf

            <div class="form-group row">
                <label for="name" class="col-md-4 col-form-label text-md-
right">{{ __('Name') }}</label>

                <div class="col-md-6">
```

```
                    <input id="name" type="text" class="form-control @error('name') is-
invalid @enderror" name="name" value="{{ old('name') }}" required autocomplete="name" auto
focus>

                    @error('name')
                        <span class="invalid-feedback" role="alert">
                            <strong>{{ $message }}</strong>
                        </span>
                    @enderror
                </div>
            </div>

            <div class="form-group row">
                <label for="email" class="col-md-4 col-form-label text-md-right">{{ __('E-
Mail Address') }}</label>

                <div class="col-md-6">
                    <input id="email" type="email" class="form-control @error('email') is-
invalid @enderror" name="email" value="{{ old('email') }}" required autocomplete="email">

                    @error('email')
                        <span class="invalid-feedback" role="alert">
                            <strong>{{ $message }}</strong>
                        </span>
                    @enderror
                </div>
            </div>
            <div class="form-group row">
                <label for="email" class="col-md-4 col-form-label text-md-
right">Alamat</label>

                <div class="col-md-6">
                    <input type="text" class="form-
control " name="alamat" value="" required autocomplete="email">
                </div>
            </div>
            <div class="form-group row">
                <label for="email" class="col-md-4 col-form-label text-md-
right">No HP</label>

                <div class="col-md-6">
                    <input type="text" class="form-
control " name="nohp" value="" required autocomplete="email">
                </div>
```

```
            </div>
            <div class="form-group row">
                <label for="email" class="col-md-4 col-form-label text-md-
right">Kota</label>

                <div class="col-md-6">
                    <input type="text" class="form-
control " name="kota" value="" required autocomplete="email">
                </div>
            </div>
            <input type="hidden" name="role" value="B">

            <div class="form-group row">
                <label for="password" class="col-md-4 col-form-label text-md-
right">{{ __('Password') }}</label>

                <div class="col-md-6">
                    <input id="password" type="password" class="form-
control @error('password') is-
invalid @enderror" name="password" required autocomplete="new-password">

                    @error('password')
                        <span class="invalid-feedback" role="alert">
                            <strong>{{ $message }}</strong>
                        </span>
                    @enderror
                </div>
            </div>

            <div class="form-group row">
                <label for="password-confirm" class="col-md-4 col-form-label text-md-
right">{{ __('Confirm Password') }}</label>

                <div class="col-md-6">
                    <input id="password-confirm" type="password" class="form-
control" name="password_confirmation" required autocomplete="new-password">
                </div>
            </div>

            <div class="form-group row mb-0">
                <div class="col-md-6 offset-md-4">
                    <button type="submit" class="btn btn-primary">
                        {{ __('Register') }}
                    </button>
```

```
                </div>
            </div>
        </form>
    </div>
</div>
```

Gambar 4.41. Script Tampilan Registrasi



Gambar 4.42. Hasil Tampilan Registrasi

Ketika user telah memiliki akun maka user akan dapat langsung melakukan login dengan *credential* akun yang ada. Gambar 4.43 berikut merupakan script saat login, dimana sistem akan mengecheck password dan email yang ada

4.2.4. Dashboard Pembeli

Pada dashboard pembeli, disediakan informasi berupa daging yang tersedia dikelompokan berdasaarkan bagian dan jenis kelamin asal daging. Gambar 4.43. dibawah merupakan script controller pada dashboard pembeli sementara gambar 4.44. merupakan script view pada aplikasi. Gambar 4.45. menjelaskan hasil dari script yang ada.

```
public function showbarang(Request $request)
    {
        $title = "Dashboard";
        $desc = "Informasi Utama";

        $cart = Cart::where('customer_id',Auth::user()->id)->count();
        if($cart < 1){
            $cart = new Cart;
            $cart->customer_id = Auth::user()->id;
            $cart->nominal = 0;
            $cart->save();
            $cart = Cart::find($cart->id);
        }else{
            $cart = Cart::where('customer_id',Auth::user()->id)->first();
        }

        $bagian = $request->bagian;
        $type = $request->type;

        $daging = Daging::where('dagings.bagian',$bagian)-
>join('rusas',function($join) use($type){
            $join->on('rusas.id','=','dagings.rusa_id')->where('rusas.status',$type);
        })->get('dagings.*');

        return view('auth.customer.detail')-
>with(compact('daging','bagian','type','title','desc'));
    }
```

Gambar 4.43. Script Controller Dashboard Pembeli

```
<div class="row">
    @if(@$dagingcust)
    @foreach($dagingcust as $key => $data)
    @php $harga = DB::table('harga_dagings')->where('type',$data->rusa->status)-
>where('bagian',$data->bagian)->first(); @endphp
    <div class="col-lg-3">
        @if($data->rusa->status)
        <a href="#" onclick="event.preventDefault(); document.getElementById('frm').submit
();">
            <form action="{{route('detailbarang')}}" id="frm" method="post">
                {{ csrf_field() }}
                <input type="hidden" name="bagian" value="{{$data->bagian}}">
                <input type="hidden" name="type" value="{{$data->rusa->status}}">
            </form>
        <div class="info-box bg-green">
            <span class="info-box-icon"><i class="fa fa-mars"></i></span>

            <div class="info-box-content">
              <span class="info-box-text">Daging {{$data->bagian}} Jantan</span>
              <span class="info-box-number">Rp. {{number_format($harga-
>nominal)}} / per kg</span>

              <div class="progress">
                <div class="progress-bar" style="width: 20%"></div>
              </div>
            </div>
            <!-- /.info-box-content -->
          </div>
          </a>
          @else
        <a href="#" onclick="event.preventDefault(); document.getElementById('frm{{$key}}'
).submit();">
            <form action="{{route('detailbarang')}}" id="frm{{$key}}" method="post">
                {{ csrf_field() }}
                <input type="hidden" name="bagian" value="{{$data->bagian}}">
                <input type="hidden" name="type" value="{{$data->rusa->status}}">
            </form>

          <div class="info-box bg-yellow">
            <span class="info-box-icon"><i class="fa fa-venus"></i></span>
```

```
            <div class="info-box-content">
              <span class="info-box-text">Daging {{$data->bagian}} Betina</span>
              <span class="info-box-number">Rp. {{number_format($harga-
>nominal)}} / per kg</span>

              <div class="progress">
                <div class="progress-bar" style="width: 50%"></div>
              </div>
            </div>
            <!-- /.info-box-content -->
          </div>
        </a>
          @endif
    </div>
    @endforeach
    @endif
</div>
```

Gambar 4.44. Script Tampilan Dashboard Pembeli

Gambar 4.45. Hasil Tampilan

4.2.5. Keranjang belanja pembeli

Ketika pembeli telah memilih daging yang ada di halaman utama, daging yang terpilih akan masuk kedalam keranjang belanja pembeli. Dimana tiap pembeli memiliki keranjang belanja sendiri. Pembeli dapat melakukan checkout dan memilih metode pembayaran setelah memasukan daging kedalam keranjang belanja. Gambar 4.46. dibawah menunjukan script pada controller dan gambar 4.47. menunjukan script pada tampilan, sementara penjelasan diatas dapat dilihat hasilnya pada gambar 4.48.

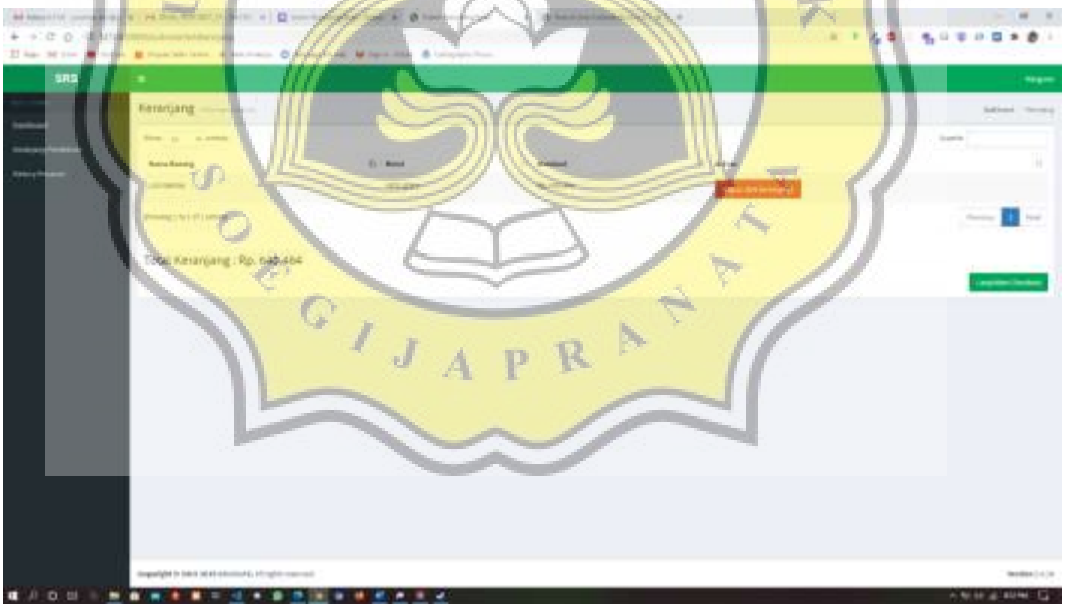
```
public function masukankeranjang(Request $request)
    {
        $daging = $request->daging;
        $nominal = $request->nominal;

        $get = Daging::find($daging);
        $cart = Cart::where('customer_id',Auth::user()->id)->first();


        $item = new ItemCart;
        $item->cart_id = $cart->id;
        $item->daging_id = $daging;
        $item->save();

        $carts = Cart::find($cart->id);
        $carts->nominal = $carts->nominal+$nominal;
        $carts->save();

        // $get->tersedia = 0;
        // $get->save();

        return redirect('/customer/beli/keranjang')-
>with(['messages'=>'Sukses menambahkan ke keranjang']);
    }

    public function hapuskeranjang(Request $request)
    {
        $item = $request->item;
        $nominal = $request->nominal;

        $cart = Cart::where('customer_id',Auth::user()->id)->first();


        $item = ItemCart::destroy($item);

        $carts = Cart::find($cart->id);
        $carts->nominal = $carts->nominal-$nominal;
        $carts->save();
        // $get->tersedia = 0;
        // $get->save();
        return redirect('/customer/beli/keranjang')-
>with(['messages'=>'Sukses menambahkan ke keranjang']);
    }
```

Gambar 4.46.Script Controller Keranjang Belanja

```
<div class="row">
    @if($cart->item->count() > 0)
    <div class="col-lg-12">
    <div class="box">
    <div class="box-body">
        <table id="example1" class="table table-bordered table-striped">
            <thead>
            <tr>
                <th>Nama Barang</th>
                <th>Berat</th>
                <th>Nominal</th>
                <th>Action</th>
            </tr>
            </thead>
            <tbody>
                @foreach($cart->item as $key => $row)
                @php $daging = DB::table('dagings')->where('id',$row->daging_id)->first(); @endphp
                @php $rusa = DB::table('rusas')->where('id',$daging->rusa_id)->first(); @endphp
                @php $harga = DB::table('harga_dagings')->where('type',$rusa->status)->where('bagian',$daging->bagian)->first();  @endphp
                    <tr>
                    <td>{{$daging->bagian}} @if($rusa->status) Jantan @else Betina @endif</td>
                    <td>{{$daging->berat}} gram</td>
                    <td>Rp. {{number_format(($harga->nominal*$daging->berat)/1000)}}</td>
                    <td>
                        <form action="{{route('hapuskeranjang')}}" id="frm{{$key}}" method="post">
                            {{ csrf_field() }}
                            <input type="hidden" name="nominal" value="{{($harga->nominal*$daging->berat)/1000}}">
                            <input type="hidden" name="item" value="{{$row->id}}">
                        </form>
                        <a href="#" onclick="event.preventDefault(); document.getElementById('frm{{$key}}').submit();" class="btn btn-danger">
                        Hapus dari keranjang?
                        </a>
                    </td>
                    </tr>
                @endforeach
            </tbody>
```

```
        </table>
    </div>
    <div class="box-footer">
        <h3>Total Keranjang : Rp. {{number_format($cart->nominal)}}</h3>
        <a href="{{route('checkout')}}" class="btn btn-success pull-
right">Lanjutkan Checkout</a>
    </div>
    </div>
    </div>
    @else
    <br>
    <br>
    <br>
        <center>
            <h4>Keranjang Anda kosong! Silahkan Belanja dulu</h4>
            <a href="{{route('home')}}" class="btn btn-primary">Belanja</a>
        </center>
    @endif
</div>
```

Gambar 4.47. Script Tampilan Keranjang Belanja



Gambar 4.48. Tampilan Keranjang Belanja

4.2.6. Checkout dan penawaran harga oleh pembeli

Pembeli mendapatkan fitur penawaran harga saat membeli daging didalam aplikasi, serta untuk membuat pembeli tidak menunggu terlalu lama,

sistem akan otomatis menolak penawaran yang tidak direspon oleh penjual selama lebih dari 1 hari setelah penawaran dibuat. Gambar 4.49. berikut merupakan script daripada controller, gambar 4.50. merupakan script tampilan halaman penawaran, sementara gambar 4.51. menjelaskan bentuk halaman penawaran tersebut.

```
public function keranjang()
    {
        $title = "Keranjang";
        $desc = "Informasi Utama";
        $cart = Cart::where('customer_id',Auth::user()->id)->count();
        if($cart < 1){
            $cart = new Cart;
            $cart->customer_id = Auth::user()->id;
            $cart->nominal = 0;
            $cart->save();
            $cart = Cart::find($cart->id);
        }else{
            $cart = Cart::where('customer_id',Auth::user()->id)->first();
        }
        return view('auth.customer.cart')->with(compact('cart','title','desc'));
    }
    public function checkout()
    {
        $title = "Checkout Belanja";
        $desc = "Halaman Checkout";
        $cart = Cart::where('customer_id',Auth::user()->id)->first();
        $cart = Cart::find($cart->id);
        return view('auth.customer.checkout')->with(compact('cart','title','desc'));
    }
    public function konfirmasipesanan(Request $request)
    {
        $title = "Checkout Belanja";
        $desc = "Halaman Checkout";
        Cart::destroy($request->cart_id);
        $pesan = new Pemesanan;
        $pesan->date = \Carbon\Carbon::now()->format('Y-m-d');
        $pesan->nominal = $request->nominal;
        $pesan->penawaran = $request->penawaran;
        $pesan->status_penawaran = 'B';
        $pesan->proses = 'W';
        $pesan->cart_id = $request->cart_id;
        $pesan->user_id = Auth::user()->id;
```

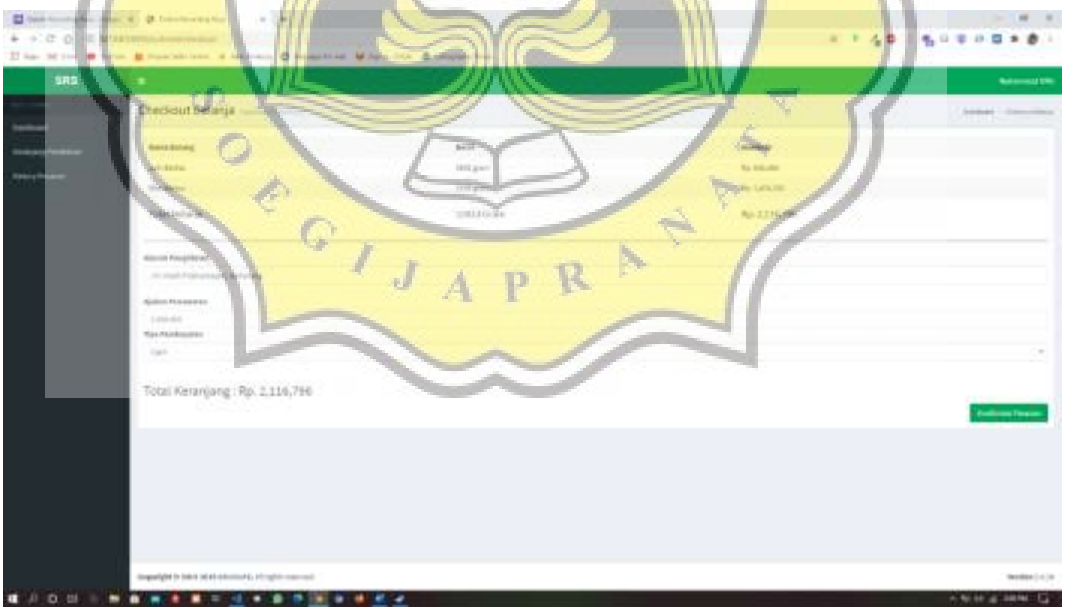
```
        $pesan->pembayaran = $request->pembayaran;
        $pesan->alamat = $request->alamat;
        $pesan->save();
        return view('auth.customer.konfirmasi')->with(compact('title','desc'));
    }
```

Gambar 4.49. Script Controller Halaman Penawaran

```
    <div class="col-lg-12">
        <form action="{{route('konfirmasipesanan')}}" method="post">
        <div class="box">
        <div class="box-body">
            <table class="table table-bordered table-striped">
                <tr>
                    <th>Nama Barang</th>
                    <th>Berat</th>
                    <th>Nominal</th>
                </tr>
                @php $tberat= 0;$tharga=0; @endphp
                    @foreach($cart->item as $key => $row)
                    @php $daging = DB::table('dagings')->where('id',$row->daging_id)-
>first(); @endphp
                    @php $rusa = DB::table('rusas')->where('id',$daging->rusa_id)-
>first(); @endphp
                    @php $harga = DB::table('harga_dagings')->where('type',$rusa->status)-
>where('bagian',$daging->bagian)->first(); @endphp
                    @php $tberat= $tberat+$daging->berat;$tharga=$tharga+(($harga-
>nominal*$daging->berat)/1000); @endphp
                        <tr>
                        <td>{{$daging->bagian}} @if($rusa-
>status) Jantan @else Betina @endif</td>
                        <td>{{$daging->berat}} gram</td>
                        <td>Rp. {{number_format(($harga->nominal*$daging-
>berat)/1000)}}</td>
                        </tr>
                    @endforeach
                <tr>
                    <td><h4>Total Belanja</h4></td>
                    <td><h4>{{$tberat}} Gram</h4></td>
                    <td><h4>Rp. {{number_format($tharga)}}</h4></td>
                </tr>
            </table>
            <hr/>
            {{ csrf_field() }}
            <input type="hidden" name="cart_id" value="{{$cart->id}}">
```

```
            <input type="hidden" name="nominal" value="{{$cart->nominal}}">
            <label> Alamat Pengiriman</label>
            <input type="text" name="alamat" class="form-control" value="{{Auth::user()-
>customer->alamat}}, {{Auth::user()->customer->kota}}"><br/>
            <label> Ajukan Penawaran</label>
            <input type="text" name="penawaran" class="form-control money" value="{{$cart-
>nominal}}">
            <label >Tipe Pembayaran</label>
            <select name="pembayaran" id="pembayaran" class="select2 form-control">
                <option value="c">Cash</option>
                <option value="b">Transfer Bank</option>
            </select>
        </div>
        <div class="box-footer">
            <h3>Total Keranjang : Rp. {{number_format($cart->nominal)}}</h3>
            <button type="submit" class="btn btn-success pull-
right">Konfirmasi Pesanan</button>
        </div>
        </div>
    </form>
</div>
```

Gambar 4.50. Script Tampilan Halaman Penawaran



Gambar 4.51. Tampilan Halaman Penawaran

4.2.7. Pemotongan Rusa

Penjual dapat memasukan data pemotongan rusa, tanggal dan tempat pemotongan, serta berat daging yang telah ditimbang pada saat pemotongan. Berikut gambar 4.52. merupakan script yang terdapat pada controller, sementara gambar 4.53. merupakan script untuk tampilan dan gambar 4.54. merupakan tampilan dari sistem pemotongan.

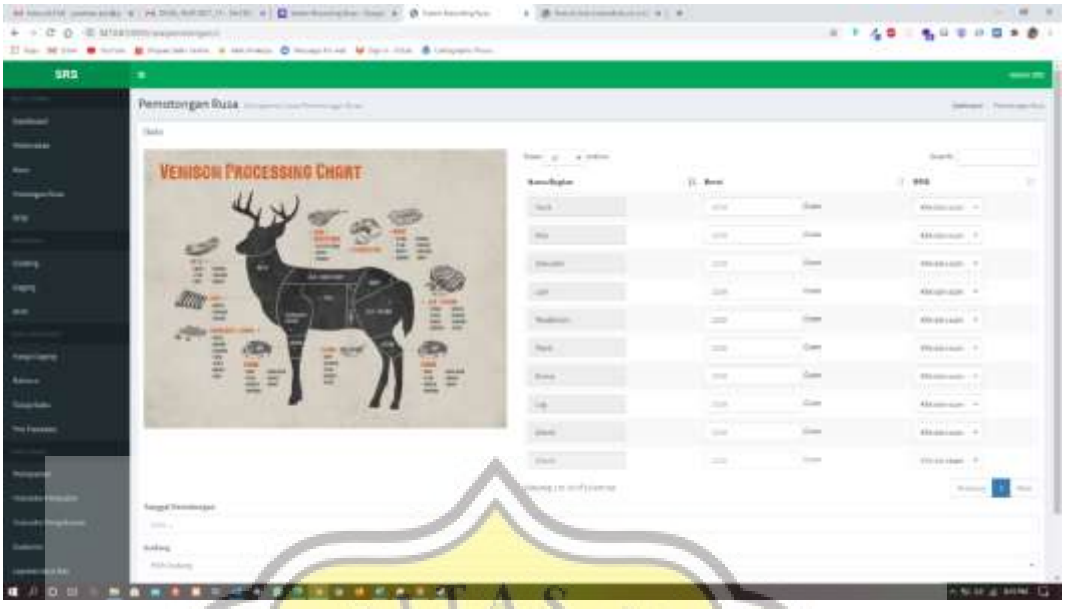
```
public function pemotongan($rusa_id){
    $title = "Pemotongan Rusa";
    $desc = "Manajemen Data Pemotongan Rusa";
    $rusa = Rusa::find($rusa_id);
    $rfid = Rfiddaging::doesntHave('daging')->get();
    $gudang = Gudang::where('peternakan_id',$rusa->peternakan_id)->get();
    $hargadaging = HargaDaging::where('type',$rusa->status)->get();

    $pemotongan = Pemotongan::where('rusa_id',$rusa_id)->first();
    return view('pemotongan.index')-
>with(compact('title','desc','rusa','gudang','rfid','hargadaging','pemotongan'));
}
public function potong(Request $request){
    $row = new Pemotongan;
    $row->date = $request->tanggal;
    $row->peternakan_id = $request->peternakan_id;
    $row->rusa_id = $request->rusa_id;
    $row->save();
    $i = 0;
    foreach($request->name as $name){
        $daging = new Daging;
        $daging->rusa_id = $request->rusa_id;
        $daging->gudang_id = $request->gudang_id;
        $daging->pemotongan_id = $row->id;
        $daging->bagian = $name;
        $daging->berat = $request->berat[$i];
        $daging->rfiddaging_id = $request->rfid_id[$i];
        $daging->save();
        $i++;
    }
    return redirect('rusa')->with(['messages'=>'Rusa telah dipotong']);
}
```

Gambar 4.52. Script pada Controller Pemotongan

```
<div class="row">
    <div class="col-lg-5">
        <img src="{{asset('cut.jpg')}}" style="width:100%;">
    </div>
    <div class="col-lg-7">
        <table class="table table-striped">
            <thead>
                <tr>
                    <th>Nama Bagian</th>
                    <th>Berat</th>
                    <th>RFID</th>
                </tr>
            </thead>
            <tbody>
            @if(@$pemotongan->date)
                @php  $hargadaging = DB::table('dagings')-
>where('pemotongan_id',@$pemotongan->id)->get(); @endphp
                @foreach($hargadaging as $data)
                <tr>
                    <td><input type="text" name="name[]" class="form-
control" value="{{$data->bagian}}" readonly></td>
                    <td><input type="number" name="berat[]" class="form-
control" value="{{$data->berat}}" placeholder="2250" readonly> Gram</td>
                    <td>
                        <div class="form-group">

                            <select name="rfid_id[]"  class="select2 form-
control" disabled>
                                <option value="#">Recorded!</option>
                            </select>
                        </div>
                    </td>
                </tr>
                @endforeach
            @else
                @foreach($hargadaging as $data)
                <tr>
                    <td><input type="text" name="name[]" class="form-
control" value="{{$data->bagian}}" readonly></td>
                    <td><input type="number" name="berat[]" class="form-
control" placeholder="2250"> Gram</td>
                    <td>
                        <div class="form-group">
```

```
                        <select name="rfid_id[]"  class="select2 form-control">
                            <option value="#">Klik dan scan</option>
                            @foreach($rfid as $data)
                                <option value="{{$data->id}}">{{$data->code}}</option>
                            @endforeach
                        </select>
                    </div>
                </td>
            </tr>
            @endforeach
        @endif
        </tbody>
    </table>

</div>
</div>
<div class="form-group">
    <input type="hidden" name="rusa_id" value="{{$rusa->id}}">
    <input type="hidden" name="peternakan_id" value="{{$rusa->peternakan_id}}">
    <label>Tanggal Pemotongan</label>
    <input type="text" name="tanggal" class="form-
control datepickers" placeholder="Date ..." required @if(@$pemotongan-
>date) value="{{$pemotongan->date}}" readonly @endif>
</div>
<div class="form-group">
    <label>Gudang</label>
    <select name="gudang_id" id="gudang_id" class="select2 form-control" @if(@$pemotongan-
>date) disabled @endif>
        <option value="#">Pilih Gudang</option>
        @foreach($gudang as $data)
            <option value="{{$data->id}}" @if(@$pemotongan->gudang_id == $data-
>id) selected @endif>{{$data->name}}</option>
        @endforeach
    </select>
</div>
@if(@$pemotongan->date)
<a href="{{route('rusa.index')}}" class="pull-right btn btn-info">Kembali</a>
@else
<button type="submit" class="pull-right btn btn-success">Simpan Data Pemotongan</button>
@endif
</form>
</div>
```

Gambar 4.53. Script pada Tampilan Pemotongan



Gambar 4.54. Tampilan pada Halaman Pemotongan

4.2.8. Inventory Daging

Pada inventory daging terdapat data dari daging yang telah dipotong pada fitur pemotongan. Disini penjual dapat melihat daging apa saja yang tersedia dan tersimpan pada gudang tertentu serta nomor RFID pada tiap potongan daging. Berikut gambar 4.55. merupakan script pada controller halaman inventory daging dan gambar 4.56. merupakan script pada tampilan halaman inventory daging, sementara gambar 4.57. merupakan penjelasan dan tampilan dari halaman inventory daging.

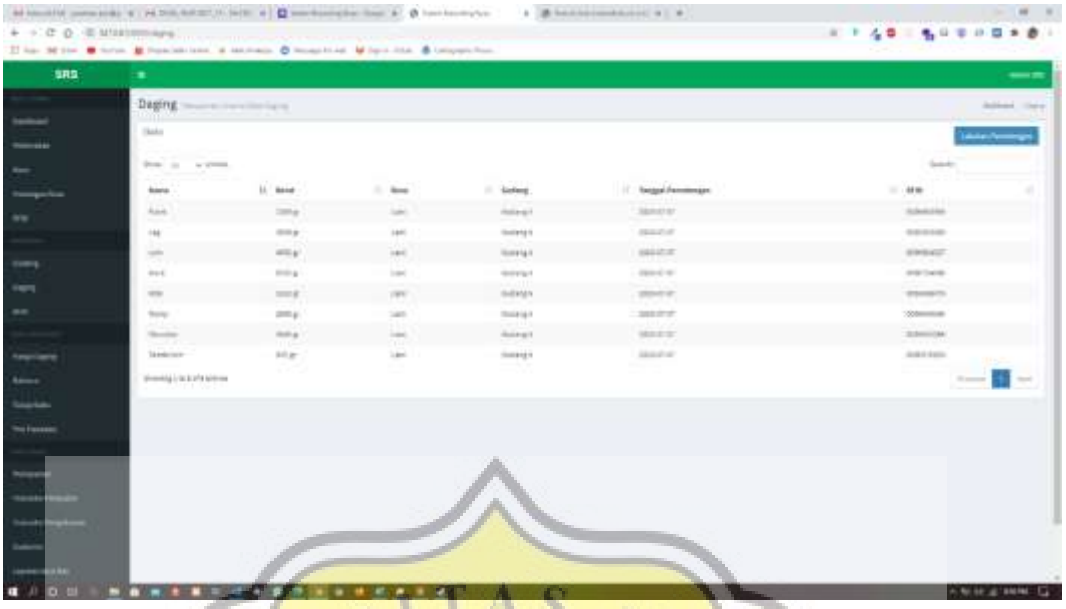
```
public function index(){
    $title = "Daging";
    $desc = "Manajemen Utama Data Daging";
    $data = Daging::orderBy('created_at','desc')->get();
    return view('daging.index')->with(compact('title','desc','data'));
}

public function create(){
    $title = "Tambah Data Gudang";
    $peternakan = Peternakan::orderBy('name')->get();
    $desc = "Manajemen Utama Data Gudang";
    $rfid = Rfiddaging::doesntHave('rusa')->get();
    return view('gudang.form')->with(compact('title','desc','peternakan'));
}
```

```php
public function store(Request $request)
{
    Gudang::insert($request->except(['_token']));
    return redirect('gudang')->with(['messages'=>'Data telah sukses di input']);
}
```

Gambar 4.55. Script pada Tampilan Inventory

```blade
<div class="box-body">
    <table id="example1" class="table table-bordered table-striped">
        <thead>
        <tr>
            <th>Nama</th>
            <th>Berat</th>
            <th>Rusa</th>
            <th>Gudang</th>
            <th>Tanggal Pemotongan</th>
            <th>RFID</th>
        </tr>
        </thead>
        <tbody>
            @foreach($data as $row)
                <tr>
                <td>{{$row->bagian}}</td>
                <td>{{$row->berat}} gr</td>
                <td>{{$row->rusa->name}}</td>
                <td>{{$row->gudang->name}}</td>
                <td>{{$row->pemotongan->date}}</td>
                <td>
                    {{$row->rfid->code}}
                </td>
            </tr>
            @endforeach
        </tbody>
    </table>
</div>
```

Gambar 4.56. Script tampilan pada Inventory



Gambar 4.57. Tampilan pada Inventory

4.2.9. Recording Silsilah Rusa

Untuk dapat memberikan data silsilah bibit kepada pembeli, penjual harus menyiapkan data silsilah pada rusa sendiri, berupa status hubungan dan perkawinan antar rusa. Berikut gambar 4.58. merupakan script controller pada silsilah rusa dan gambar 4.59. merupakan script tampilan pada silsilah rusa, sementara gambar 4.60. merupakan penjelasan dan tampilan pada manajemen data silsilah rusa.

```
public function recording($id){
        $this->middleware('auth');
        $rusa = Rusa::find($id);
        $title = $rusa->name." Recording";
        $desc = "Recording Relationship Rusa";
        return view('rusa.recording')->with(compact('rusa','title','desc'));
    }
public function tambahRelasi($id){
        $self = Rusa::find($id);
        $relasi = Relationship::get();
        $rusa = Rusa::get();
        $title = $self->name." Recording Relasi";
        $desc = "Recording Relationship Rusa";
```

```
        return view('rusa.form-recording')-
>with(compact('self','rusa','relasi','title','desc'));
    }
public function storeRelasi(Request $request){
        $rusa = Rusa::find($request->rusa_id);
        if($request->relationship_id == '2' || $request->relationship_id == '3'){
            $anak = Rusa::find($request->rusa_to_id);
            $anak->relationship()->attach(4,['rusa_to_id'=>$request->rusa_id]);
        }
        if($request->relationship_id == '1'){
            $anak = Rusa::find($request->rusa_to_id);
            $anak->relationship()->attach(1,['rusa_to_id'=>$request->rusa_id]);
        }
        $rusa->relationship()->attach($request->relationship_id,['rusa_to_id'=>$request-
>rusa_to_id]);
        return redirect('rusa/recording/'.$request->rusa_id)-
>with(['messages'=>'Berhasil menambahkan relasi']);
    }
```

Gambar 4.58. Script Controller pada Halaman silsilah rusa

```
<div class="box-body">
            <table id="example1" class="table table-bordered table-striped">
                <thead>
                <tr>
                  <th>Nama</th>
                  <th>Relasi</th>
                  <th>Lahir</th>
                  <th>Jenis Kelamin</th>
                  <th>Action</th>
                </tr>
                </thead>
                <tbody>
                    @php $id = $rusa->id; @endphp
                    @foreach($rusa->relationship as $row)
                        @php
                            $rusa = DB::table('rusas')->where('id',$row->pivot-
>rusa_to_id)->first();
                        @endphp
                        <tr>
                        <td>{{$rusa->name}}</td>
                        <td>{{$row->name}}</td>
                        <td>{{$rusa->lahir}}</td>
                        <td>@if($rusa->status) Jantan @else Betina @endif</td>
```

```
                        <td><a href="{{route('rusa.recording.destroy.relasi',$row->pivot->id)}}" class="btn btn-danger">Hapus Relasi</a></td>
                        </tr>
                    @endforeach
                </tbody>
            </table>
        </div>
        <iframe src="{{route('rusa.bagan', ['id' => $id])}}" width="100%" height="400" frameborder="0" zoom="0.5"></iframe>
    </div>
```

Gambar 4.59. Script Tampilan pada Halaman silsilah rusa

Gambar 4.60. Tampilan Halaman Silsilah Rusa

4.2.10. Sistem Pemesanan pada Penjual

Penjual dapat memproses pemesanan, menerima dan menolak penawaran serta pembayaran dalam halaman sistem ini. Data transaksi yang terdapat pada sistem ini akan otomatis masuk kedalam laporan keuangan arus kas dan laba rugi. Berikut gambar 4.61. merupakan script controller pada pemesanan dan gambar 4.62. merupakan script tampilan pada pemesanan, sementara gambar 4.63. merupakan penjelasan dan tampilan pada manajemen data pemesanan.

```php
public function confirmpemesanan($id)
{
    $title = "Pemesanan";
    $desc = "Manajemen Utama Data Pemesanan";

    $data = Pemesanan::find($id);
    $balance = Balance::get();

    return view('pemesanan.confirm')->with(compact('title','desc','data','balance'));
}

public function confirmedpemesanan(Request $request)
{
    $pemesanan = $request->pemesanan_id;
    $data = Pemesanan::find($pemesanan);
    if($request->status_penawaran){
        $nominal = $data->penawaran;
    }else{
        $nominal = $data->nominal;
    }
    if($data->pembayaran === "b"){
        $data->proses = "M";
    }else{
        $data->proses = "P";

        $cash = new Cashflow;
        $cash->date = \Carbon\Carbon::now()->format('Y-m-d');
        $cash->nominal = $nominal;
        $cash->balance_id = $request->balance_id;
        $cash->type = 1;
        $cash->save();
        $customer = DB::table('customers')->where('user_id',$data->user_id)->first();
        $penjualan = new Transaction;
        $penjualan->name = "Pembayaran pemesanan dengan kode ".$data->date."-".$data->user_id."-".$data->cart_id;
        $penjualan->date = \Carbon\Carbon::now()->format('Y-m-d');
        $penjualan->type = 1;
        $penjualan->nominal = $nominal;
        $penjualan->cashflow_id = $cash->id;
        $penjualan->customer_id = $customer->id;
        $penjualan->po_id = 2;
        $penjualan->save();
        foreach(ItemCart::where('cart_id',$data->cart_id)->get() as $daging){
            $dagingnya = Daging::find($daging->daging_id);
```

```
            $dagingnya->tersedia = 0;
            $dagingnya->save();
            $item = new ItemTransaction;
            $item->transaction_id = $penjualan->id;
            $item->daging_id = $dagingnya->id;
            $item->name = $dagingnya->bagian;
            $item->ket = $dagingnya->berat;
            $item->save();
        }
    }
    $data->status_penawaran = $request->status_penawaran;
    $data->save();
    return redirect('pemesanan-admin')->with(['messages'=>'Pesanan terkonfirmasi!']);
}
```

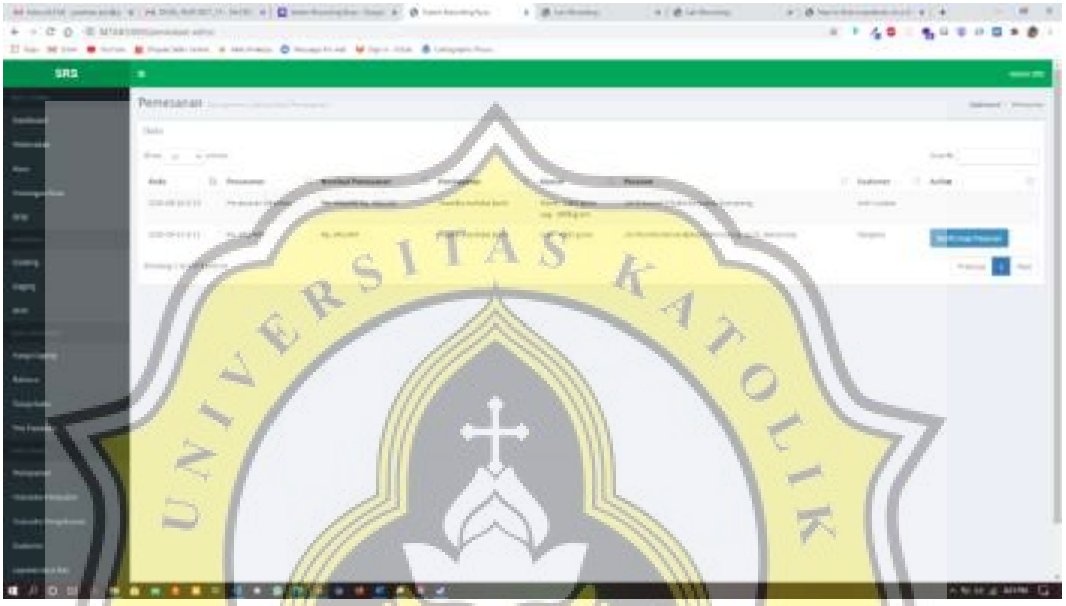
Gambar 4.61. Script Controller pada Pemesanan

```
<div class="box-body">
    <table id="example1" class="table table-bordered table-striped">
        <thead>
        <tr>
            <th>Kode</th>
            <th>Penawaran</th>
            <th>Nominal Pemesanan</th>
            <th>Pembayaran</th>
            <th>Alamat</th>
            <th>Pesanan</th>
            <th>Customer</th>
            <th>Action</th>
        </tr>
        </thead>
        <tbody>
            @foreach($data as $row)
                <tr>
                <td>{{$row->date}}-{{$row->user_id}}-{{$row->cart_id}}</td>
                <td>
                    @if($row->status_penawaran === "Y")
                        Penawaran diterima
                    @elseif($row->status_penawaran === "T")
                        Penawaran ditolak
                    @else
                    Rp. {{number_format($row->penawaran)}}
                    @endif
                </td>
                <td>
```

```
                    @if($row->status_penawaran === "Y")
                    <strike>Rp. {{number_format($row->nominal)}}</strike>
                    Rp. {{number_format($row->penawaran)}}
                    @else
                    Rp. {{number_format($row->penawaran)}}
                    @endif
                </td>
                <td>
                    @if($row->pembayaran === "b")
                        Transfer melalui bank
                    @else
                        Cash
                    @endif
                </td>
                <td>
                    @php $item = DB::table('item_carts')->where('cart_id',$row->cart_id)->get(); @endphp
                    @foreach($item as $data)
                        @php $daging = DB::table('dagings')->where('id',$data->daging_id)->first(); @endphp
                        {{$daging->bagian}} -
                        {{$daging->berat}} gram<br/>
                    @endforeach
                </td>
                <td>{{$row->alamat}}</td>
                <td>{{$row->user->name}}</td>
                <td>
                    @if($row->proses === 'W')
                    <a href="{{route('confirm.pemesanan',$row->id)}}" class="btn btn-primary">Konfirmasi Pesanan</a>
                    @elseif($row->proses === 'M')
                    <a href="{{route('confirm.pemesanan',$row->id)}}" class="btn bg-gray">Menunggu Konfirmasi Pembayaran Customer</a>
                    @elseif($row->proses === 'C')
                    <a href="{{route('confirm.payment',$row->id)}}" class="btn btn-warning">Check Pembayaran Customer</a>
                    @elseif($row->proses === 'P')
                    <a href="{{route('confirm.packing',$row->id)}}" class="btn btn-success">Siap Kirim?</a>
                    @elseif($row->proses === 'K')
                    Menunggu Konfirmasi Customer Menerima Barang
                    @elseif($row->proses === 'K')
                    <div class="btn btn-success">Barang telah diterima</div>
                    @endif
```

```
                </td>
                </tr>
            @endforeach
        </tbody>
    </table>
</div>
```

Gambar 4.62. Script Tampilan pada Halaman Pemesanan



Gambar 4.63 Tampilan Halaman Pemesanan

4.2.11. Sistem Pembuatan Laporan Arus Kas dan Laba Rugi

Terdapat fitur pembuatan Laporan arus kas secara sistematis dari sistem. Dengan mengolah data inputan pada pemasukan dan pengeluaran transaksi dapat memberikan output berupa Laporan arus kas dan laba rugi pada organisasi. Berikut gambar 4.63. merupakan script controller pada laporan dan gambar 4.64. merupakan script tampilan pada laporan arus kas serta gambar 4.65 merupakan script tampilan pada laporan laba rugi, sementara gambar 4.66. dan gambar 4.67 merupakan penjelasan dan tampilan pada laporan arus kas dan laba rugi.

```
public function aruskas()
{
    $title = "Laporan Arus Kas";
```

```
    $desc = "Manajemen Data Kas Keuangan";

    $balance = Balance::orderBy('name','desc')->get();
    $record = RecordBalance::groupBy('date')->get();

    return view('laporan.aruskas')->with(compact('title','desc','balance','record'));
}

public function aruskasfilter(Request $request)
{
    $title = "Laporan Arus Kas";
    $desc = "Manajemen Data Kas Keuangan";
    $balance = Balance::orderBy('name','desc')->get();
    $balance_id = $request->balance_id;
    $record = RecordBalance::groupBy('date')->get();

    $start = $request->start;
    $end = $request->end;
    if($request->end === 'now'){
        $end = \Carbon\Carbon::now()->format('Y-m-d');
    }
    $startbalance = RecordBalance::where('date',$start)->where('balance_id',$balance_id)-
>first();
    $endbalance = RecordBalance::where('date',$end)->where('balance_id',$balance_id)-
>first();
    if(empty($endbalance)){
        $endbalance = Balance::find($balance_id);
    }


    $trans = Cashflow::with('trans')->where('balance_id',$request->balance_id)-
>whereBetween('date',[$start,$end])->get();

    return view('laporan.aruskas')-
>with(compact('title','desc','balance','record','trans','balance_id','start','end','startb
alance','endbalance'));
}

public function labarugi()
{
    $title = "Laporan Laba Rugi";
    $desc = "Manajemen Data Laba Rugi";

    $balance = Balance::orderBy('name','desc')->get();
    $record = RecordBalance::groupBy('date')->get();
```

```
    return view('laporan.labarugi')->with(compact('title','desc','balance','record'));
}

public function labarugifilter(Request $request)
{
    $title = "Laporan Laba Rugi";
    $desc = "Manajemen Data Laba Rugi";
    $balance = Balance::orderBy('name','desc')->get();
    $balance_id = $request->balance_id;
    $record = RecordBalance::groupBy('date')->get();

    $start = $request->start;
    $end = $request->end;
    if($request->end === 'now'){
        $end = \Carbon\Carbon::now()->format('Y-m-d');
    }

    $startbalance = RecordBalance::where('date',$start)->where('balance_id',$balance_id)->first();
    $endbalance = RecordBalance::where('date',$end)->where('balance_id',$balance_id)->first();

    $totalpenjualan = Transaction::where('po_id',2)->whereBetween('date',[$start,$end])->sum('nominal');
    $hpp = Transaction::where('po_id',6)->whereBetween('date',[$start,$end])->sum('nominal');

    $pos = Po::where('type',0)->whereNotIn('id',['1','6'])->get();

    if(empty($endbalance)){
        $endbalance = Balance::find($balance_id);
    }

    $trans = Cashflow::with('trans')->where('balance_id',$request->balance_id)->whereBetween('date',[$start,$end])->get();

    return view('laporan.labarugi')->with(compact('title','desc','balance','record','trans','balance_id','start','end','startbalance','endbalance','totalpenjualan','hpp','pos'));
}
```

Gambar 4.63. Script Controller untuk Laporan Arus Kas dan Laba Rugi

```
<div class="box">
    <div class="box-body">
            <table class="table table-striped table-bordered">
                <tr>
                    <th>Tanggal</th>
                    <th>Keterangan</th>
                    <th>Kredit</th>
                    <th>Debit</th>
                    <th>Posisi</th>
                </tr>
                <tr>
                    <td>{{\Carbon\Carbon::createFromFormat('Y-m-d',$startbalance->date)-
>format('d M, Y')}}</td>
                    <td>Saldo Awal {{$startbalance->name}}</td>
                    <td></td>
                    <td></td>
                    <td>Rp. <span class="pull-right">{{number_format($startbalance-
>nominal)}}</span></td>
                </tr>
                @php $posisi = $startbalance->nominal @endphp
                @if(!empty($trans))
                @foreach($trans as $data)
                    <tr>
                        <td>{{\Carbon\Carbon::createFromFormat('Y-m-d',$data->trans-
>date)->format('d M, Y')}}</td>
                        <td>{{$data->trans->name}}</td>
                        @if($data->trans->type)
                            <td></td>
                            <td>Rp. <span class="pull-right">{{number_format($data->trans-
>nominal)}}</span></td>
                            @php $posisi = $posisi + $data->trans->nominal @endphp
                        @else
                            <td>Rp. <span class="pull-right">{{number_format($data->trans-
>nominal)}}</span></td>
                            <td></td>
                            @php $posisi = $posisi - $data->trans->nominal @endphp
                        @endif
                        <td>Rp. <span class="pull-
right">{{number_format($posisi)}}</span></td>
                    </tr>
                @endforeach
                @endif
                <tr>
                    @if($endbalance->date)
```

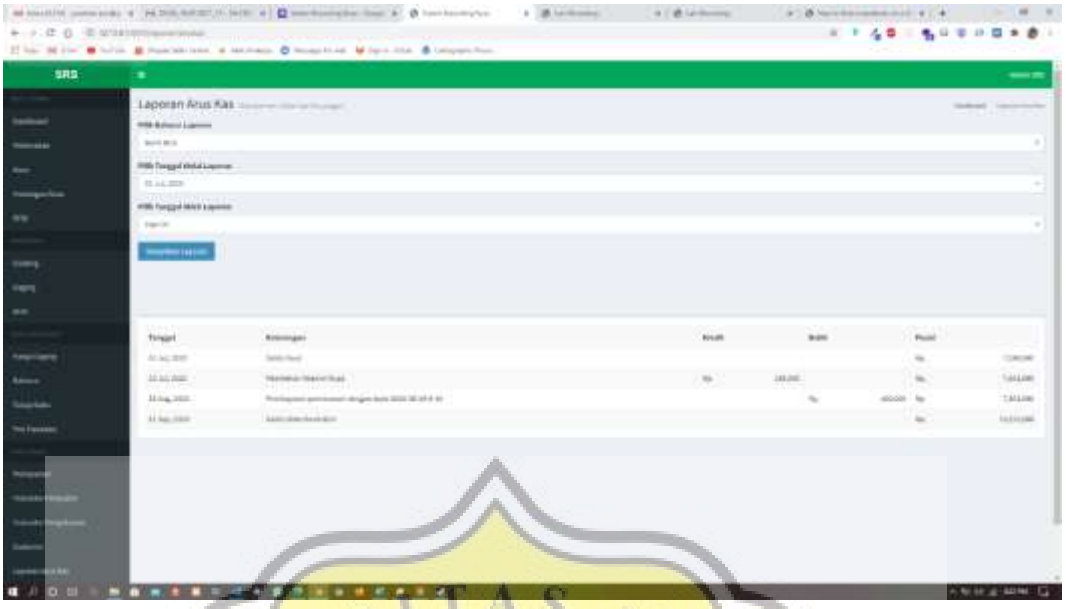
```
                    <td>{{\Carbon\Carbon::createFromFormat('Y-m-d',$endbalance->date)-
>format('d M, Y')}}</td>
                    @else
                    <td>{{\Carbon\Carbon::now()->format('d M, Y')}}</td>
                    @endif
                    <td>Saldo Ahkir {{$endbalance->name}}</td>
                    <td></td>
                    <td></td>
                    <td>Rp. <span class="pull-right">{{number_format($endbalance-
>nominal)}}</span></td>
                </tr>
            </table>
    </div>
</div>
```

Gambar 4.64. Script Tampilan Laporan Arus Kas

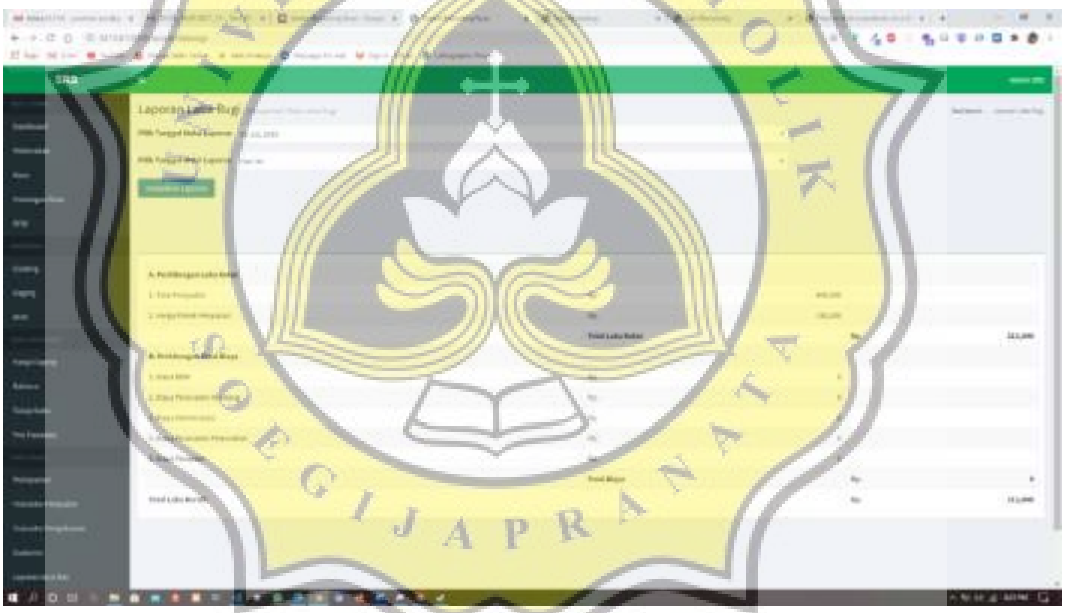
```
<div class="box">
    <div class="box-body">
        <table class="table table-striped table-bordered">
            <tr>
                <th>A. Perhitungan Laba Kotor</th>
                <th></th>
                <th></th>
            </tr>
            {{-- Total Penjualan --}}
            <tr>
                <td>1. Total Penjualan</td>
                <td>Rp. <span class="pull-
right">{{number_format($totalpenjualan)}}</span></td>
                <td></td>
            </tr>
            {{-- HPP --}}
            <tr>
                <td>2. Harga Pokok Penjualan</td>
                <td>Rp. <span class="pull-right">{{number_format($hpp)}}</span></td>
                <td></td>
            </tr>
            {{-- Laba Kotor --}}
            <tr>
                @php $labakotor = $totalpenjualan - $hpp; @endphp
                <td></td>
                <td><b> Total Laba Kotor</b></td>
                <td><b>Rp. <span class="pull-
right">{{number_format($labakotor)}}</span></b></td>
```

```
                </tr>
                {{-- Total Biaya --}}
                <tr>
                    <th>B. Perhitungan Total Biaya</th>
                    <th></th>
                    <th></th>
                </tr>
                {{-- Looping Pos Biaya --}}
                @php $totalbiaya = 0; $number = 1; @endphp
                @foreach($pos as $data)
                    @php  $nominal = DB::table('transactions')->where('po_id',$data->id)-
>whereBetween('date',[$start,$end])->sum('nominal'); @endphp
                    <tr>
                        <td>{{$number}}. {{$data->name}}</td>
                        <td>Rp. <span class="pull-
right">{{number_format($nominal)}}</span></td>
                        <td></td>
                    </tr>
                    @php $totalbiaya = $totalbiaya + $nominal; $number++; @endphp
                @endforeach
                {{-- Total Biaya Kotor --}}
                <tr>
                    <td></td>
                    <td><b> Total Biaya</b></td>
                    <td><b>Rp. <span class="pull-
right">{{number_format($totalbiaya)}}</span></b></td>
                </tr>
                {{-- Laba Bersih --}}
                <tr>
                    @php $labakotor = $labakotor - $totalbiaya; @endphp
                    <td colspan="2"><b> Total Laba Bersih</b></td>
                    <td><b>Rp. <span class="pull-
right">{{number_format($labakotor)}}</span></b></td>
                </tr>

            </table>
    </div>
</div>
```

Gambar 4.65. Script Tampilan Laporan Laba Rugi



Gambar 4.66. Tampilan Laporan Arus Kas



Gambar 4.67. Tampilan Laporan Laba Rugi

4.3. Pengujian Aplikasi

Setelah dilakukan testing terhadap aplikasi dan melakukan pengumpulan data dengan cara membagikan kuisioner terhadap 84 responden dengan usia diantara 25 – 40 tahun diperoleh sebagai berikut.

4.3.1. Profil Responden

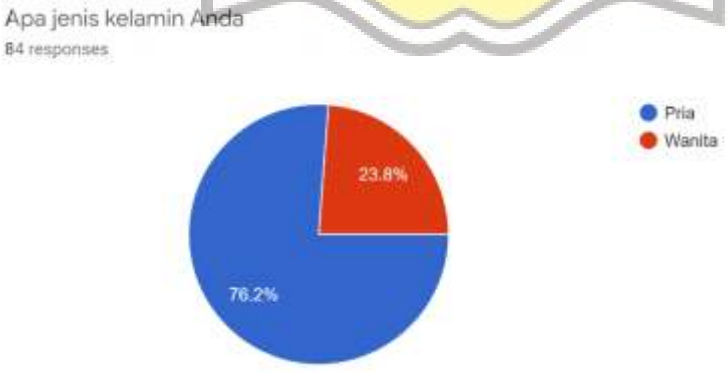
1. Usia

   Responden terbanyak adalah yang berusia 30 – 35 tahun dengan prosentase 65,5%, disusul dengan responden berusia 25 – 30 tahun dengan jumlah prosentase 21,4% dan terahkir 36 tahun keatas dengan jumlah prosentase 13,1%, sesuai dengan gambar 4.68. yang menunjukan diagram usia responden.

Gambar 4.68.Diagram Usia Responden

2. Jenis Kelamin

   Responden terbanyak merupakan pria dengan prosentase 76,2% dan wanita dengan prosentase 23,8% sesuai dengan gambar 4.69. yang menunjukan diagram jenis kelamin responden.



Gambar 4.69. Diagram Jenis Kelamin

4.3.2. Pembahasan Data Kuisioner terhadap Aplikasi

Data kuisioner yang didapatkan dari 84 responden digunakan untuk menilai aplikasi SRS sebagai berikut.

Gambar 4.70. menunjukan 25% sangat setuju, 57,1% setuju, 14,3% netral dan 3,6% tidak setuju bahwa aplikasi SRS mudah untuk dioperasikan.

Gambar 4.70. Diagram Aplikasi mudah dioperasikan

Gambar 4.71. menunjukan 25% sangat setuju, 63,1% setuju, 8,3% netral dan 3,6% tidak setuju bahwa aplikasi SRS memerlukan keahlian khusus dalam memakainya.

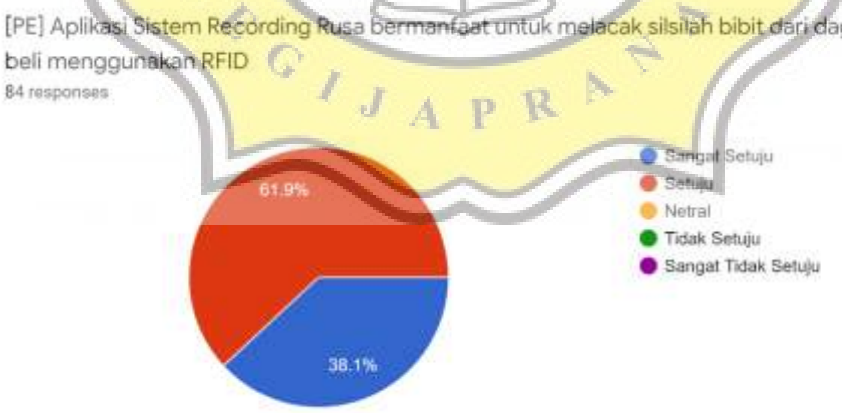
Gambar 4.71. Diagram Aplikasi tidak memerlukan keahlian khusus

Gambar 4.72. menunjukan 26,2% sangat setuju, 58,3% setuju, 10,7% netral dan 4,8% tidak setuju bahwa mereka mudah dalam memahami penggunaan aplikasi.

Gambar 4.72. Diagram Aplikasi mudah dipahami

Dari 3 pertanyaan diatas rata – rata mayoritas responden setuju untuk bahwa aplikasi mudah digunakan (variable EE).

Gambar 4.73. menunjukan 38,1% sangat setuju dan 61,9% setuju bahwa aplikasi bermanfaat untuk melacak silsilah bibit dari daging yang dibeli menggunakan RFID.

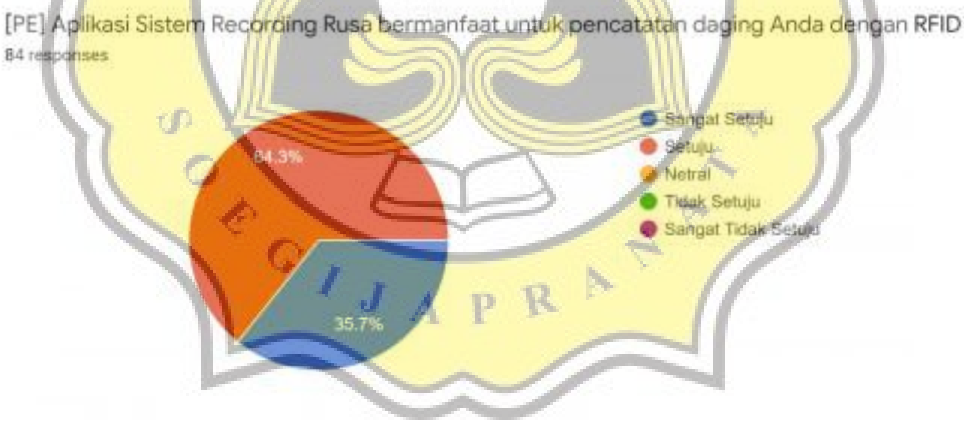


Gambar 4.73. Diagram Aplikasi bermanfaat melacak silsilah bibit daging menggunakan RFID

Gambar 4.74. menunjukan 27,4% sangat setuju, 67,9% setuju dan 4,8% netral bahwa aplikasi bermanfaat untuk pembelian dan penawaran daging.

Gambar 4.74. Diagram Aplikasi bermanfaat untuk pembelian dan penawaran

Gambar 4.75. menunjukan 35,7% sangat setuju dan 64,3% setuju bahwa aplikasi bermanfaat untuk pencatatan daging dengan RFID.



Gambar 4.75. Diagram Aplikasi bermanfaat pencatatan daging dengan RFID

Dari 3 pertanyaan diatas rata – rata mayoritas responden percaya bahwa aplikasi memberikan keuntungan bagi diri mereka (variable PE).

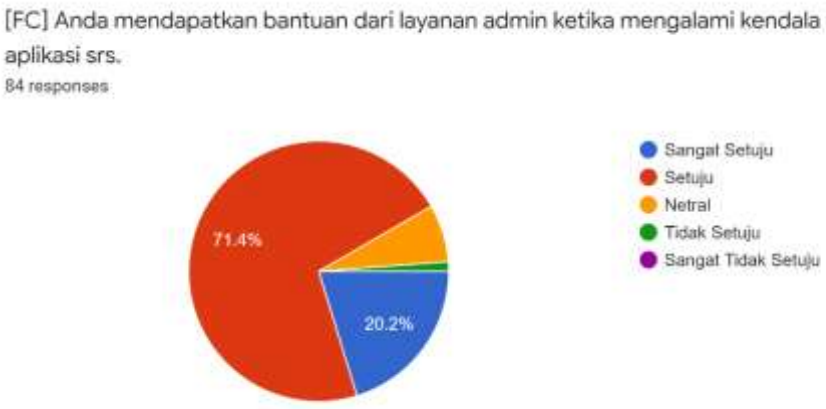
Gambar 4.76. menunjukan 34,5% sangat setuju dan 65,5% setuju bahwa memiliki sumber daya untuk menggunakan aplikasi berupa web browser dan internet.

Gambar 4.76. Diagram Responden memiliki sumber daya untuk menggunakan aplikasi

Gambar 4.77. menunjukan 32,1% sangat setuju, 48,8% setuju, 17,9% netral dan 1,2% tidak setuju bahwa responden memiliki pengetahuan yang dibutuhkan untuk menggunakan aplikasi.

Gambar 4.77. Diagram Responden memiliki pengetahuan yang dibutuhkan untuk menggunakan aplikasi

Gambar 4.78. menunjukan 20,2% sangat setuju, 71,4% setuju, 7,1% netral dan 1,2% tidak setuju bahwa responden mendapatkan bantuan layanan admin ketika mengalami kendala dalam menggunakan aplikasi.



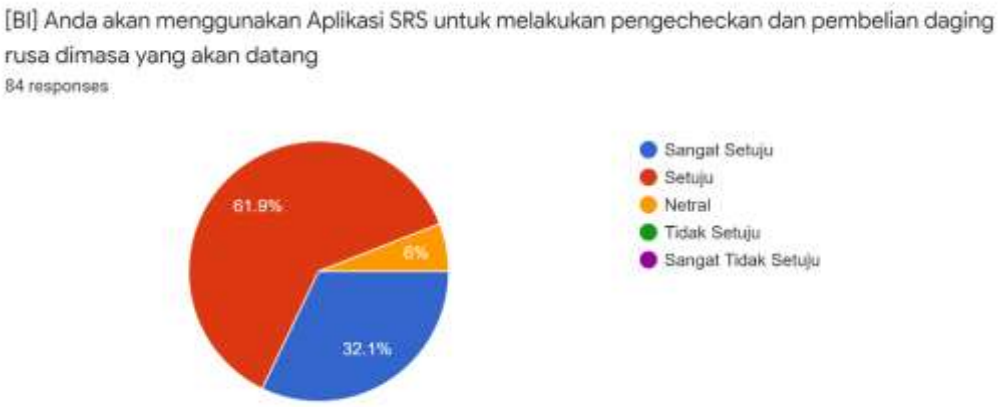
Gambar 4.78. Diagram Responden mendapatkan bantuan layanan admin

Dari 3 pertanyaan diatas rata – rata mayoritas responden setuju bahwa mereka mampu menggunakan aplikasi dan memiliki sumber daya pendukung (internet dan browser) (variable FC).

Gambar 4.79. menunjukan 15,5% sangat setuju, 75% setuju, 9,5% netral bahwa responden akan terus menggunakan aplikasi.

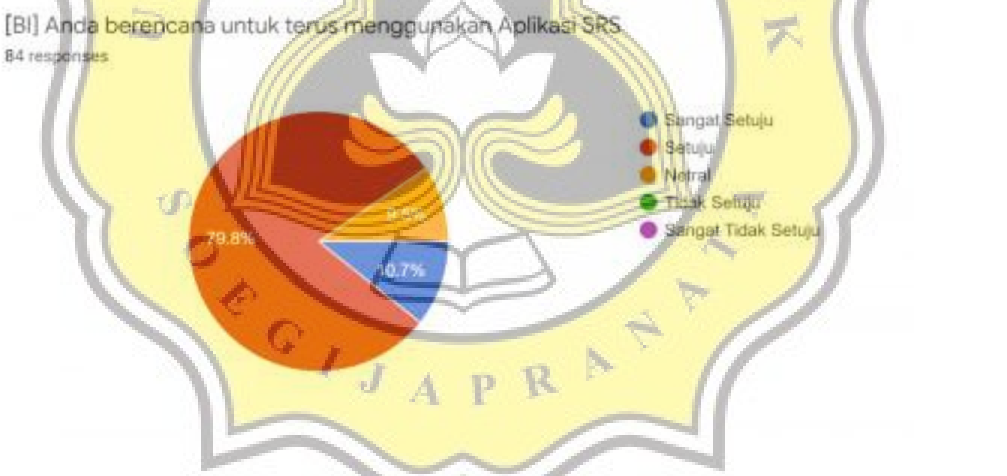
Gambar 4.79 Diagram responden berniat untuk terus menggunakan aplikasi

Gambar 4.80. menunjukan 32,1% sangat setuju, 61,9% setuju, 6% netral bahwa responden akan menggunakan aplikasi untuk melakukan pengecheckan dan pembelian daging dimasa mendatang.



Gambar 4.80. Diagram responden akan menggunakan aplikasi untuk melakukan pengechekan dan pembelian daging

Gambar 4.81. menunjukan 10,7% sangat setuju, 79,8% setuju, 9,5% netral bahwa responden berencana untuk terus menggunakan aplikasi.

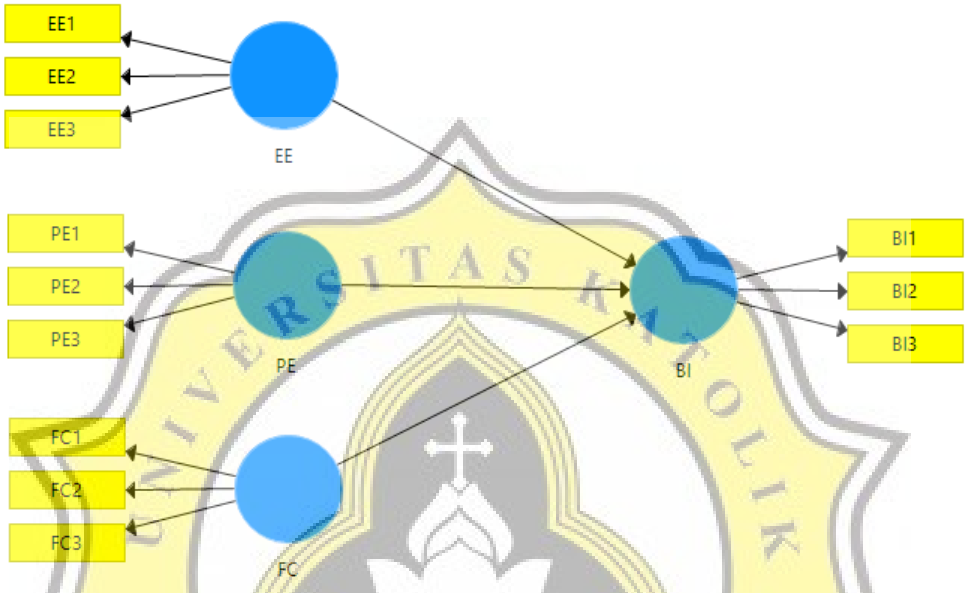


Gambar 4.81. Diagram responden berencana untuk terus menggunakan aplikasi

Dari 3 pertanyaan diatas rata – rata menunjukan mayoritas responden bersedia untuk menggunakan aplikasi secara berkala dalam jangka kedepan (variable BI).

4.3.3. Model Variable Penelitian

Gambar 4.82. berikut merupakan gambar model variable penelitian dimana variable independent merupakan EE, PE, dan FC serta variable dependent merupakan BI.



Gambar 4.82. Model Variable Peneltiain

4.3.4. Uji Validitas

Data tabel 4.1 menunjukan bukti validitas sebuah kuisioner yang dapat dilihat dari variable yang mengelompok (konvergen) pada Rotated Component Matrix serta memiliki nilai variable diatas 0.5. Dapat dilihat pada variable EE yang mengelompok pada nilai 0.574 – 0.823, variable PE yang mengelompok pada 0.814 – 0.935, variable FC yang mengelompok pada nilai 0.563 – 0.848 serta variable BI yang mengelompok pada nilai 0.503 – 0.906.

Tabel 4.1 Tabel hasil uji validitas

Rotated Component Matrix[a]

	Component		
	1	2	3
EE1	**.765**	.210	.360
EE2	**.574**	-.041	.577
EE3	**.823**	.175	.300
PE1	.040	**.922**	.036
PE2	.129	**.814**	-.015
PE3	.060	**.935**	.092
FC1	.028	-.054	**.848**
FC2	.341	.250	**.705**
FC3	.302	.001	**.563**
BI1	**.906**	.057	.213
BI2	**.503**	.305	.451
BI3	**.900**	.047	.192

a. Rotation converged in 5 iterations.

4.3.5. Uji Realibilitas

Data tabel 4.2 menunjukan tentang nilai dari sebuah hasil uji realibilitas.

Tabel 4.2 Tabel rentang uji realibilitas

Cronbach's alpha	*Internal consistency*
α ≥ 0.9	*Excellent*
0.9 > α ≥ 0.8	*Good*
0.8 > α ≥ 0.7	*Acceptable*
0.7 > α ≥ 0.6	*Questionable*
0.6 > α ≥ 0.5	*Poor*
0.5 > α	*Unacceptable*

.Sesuai dengan table 4.3 dibawah yang menunjukan variable EE, PE, BI dapat dipertanggung jawabkan, sementara variable FC menunjukan Questionable dimana hasil dari variable ini masih dipertanyakan readibilitasnya.

Tabel 4.3 Tabel hasil uji realibilitas

VARIABLE	CRONBACH'S ALPHA	INTERNAL CONSISTENCY
EE	0.869	Good
PE	0.877	Good
FC	0.629	Questionable
BI	0.855	Good

4.3.6. Uji Korelasi

Tabel 4.4 mengenai uji korelasi menunjukan korelasi variable EE dan FC menunjukan korelasi yang kuat terhadap variable BI sementara untuk variable PE memiliki korelasi yang lemah terhadap variable BI.

Tabel 4.4. Hasil Uji Korelasi

Correlations

		SEE	SPE	SFC	SBI
SEE	Pearson Correlation	1	.226*	.665**	.766**
	Sig. (2-tailed)		.039	.000	.000
	N	84	84	84	84
SPE	Pearson Correlation	.226*	1	.176	.242*
	Sig. (2-tailed)	.039		.109	.027
	N	84	84	84	84
SFC	Pearson Correlation	.665**	.176	1	.574**
	Sig. (2-tailed)	.000	.109		.000
	N	84	84	84	84
SBI	Pearson Correlation	.766**	.242*	.574**	1
	Sig. (2-tailed)	.000	.027	.000	
	N	84	84	84	84

*. Correlation is significant at the 0.05 level (2-tailed).

**. Correlation is significant at the 0.01 level (2-tailed).

4.3.7. Wawancara Penjual

Tabel 4.5 dibawah merupakan hasil wawancara yang didapatkan penulis kepada salah satu peternak selaku pengelola/admin aplikasi.

Tabel 4.5 Hasil Wawancara kepada Admin

Pertanyaan	Jawaban	Keterangan
Bagaimana penggunaan aplikasi oleh pengelola selama ini?	Cukup lancar, sudah tidak ada bug seperti pada awal2 pembuatan mas, untuk laporan bulan ini sudah mulai digunakan	Aplikasi telah digunakan dengan lancar oleh pengelola dan fungsi aplikasi sudah digunakan
Apakah aplikasi ini sudah sesuai dengan fungsi yang diharapkan?	Sudah, yang terpenting kami dapat memberikan rekap silsilah bibit dari daging rusa dan pengecheckan daging dengan cara scan	Aplikasi berfungsi dengan baik dan memenuhi kebutuhan pihak pengelola
Bagaimana dengan kelebihan aplikasi?	Scan dagingnya sudah sangat mendukung waktu pemotongan mas serta laporan keuangannya dapat dengan mudah didownload	Kelebihan Aplikasi : - Scan RFID - Laporan Keuangan
Bagaimana dengan kekurangan aplikasi?	Bisa ditambahkan mas supaya peternak lain juga dapat mengakses laporan keuangan dan pemesanan yang ada, sehingga bisa saling mengkoreksi bersama	Kekurangan Aplikasi : - User Peternak memerlukan akses untuk melihat laporan keuangan dan pemesanan

Tabel 4.6 dibawah merupakan hasil wawancara yang didapatkan penulis kepada salah satu peternak selaku penjual didalam aplikasi.

Tabel 4.6 Hasil Wawancara kepada Admin

Pertanyaan	Jawaban	Keterangan
Bagaimana penggunaan aplikasi oleh pengelola selama ini?	Sudah baik	Sudah baik
Apakah aplikasi ini sudah sesuai dengan fungsi yang diharapkan?	Sudah sesuai, dapat memberikan tampilan keturunan bibit rusa	Sudah berjalan dengan baik
Bagaimana dengan kelebihan aplikasi?	Bisa scan itu mas waktu pemotongan, jadi tidak perlu susah2 catat lagi	Kelebihan Aplikasi : - Scan RFID
Bagaimana dengan kekurangan aplikasi?	Saya tidak bisa lihat laporan keuangan mas, harus minta ke bagian keuangan dulu baru dicetak	Kekurangan Aplikasi : - User Peternak memerlukan akses untuk melihat laporan keuangan

Tabel 4.7 dibawah merupakan hasil wawancara yang didapatkan penulis kepada salah satu peternak selaku penjual didalam aplikasi.

Tabel 4.7 Hasil Wawancara kepada Peternak

Pertanyaan	Jawaban	Keterangan

Bagaimana penggunaan aplikasi oleh pengelola selama ini?	Lancar mas, berjalan baik	Sudah baik
Apakah aplikasi ini sudah sesuai dengan fungsi yang diharapkan?	Sesuai	Sudah berjalan dengan baik
Bagaimana dengan kelebihan aplikasi?	Kelebihannya bisa scan itu kan mas waktu pemotongan	Kelebihan Aplikasi : - Scan RFID
Bagaimana dengan kekurangan aplikasi?	Kekurangannya kalau mau lihat laporan langsung dari aplikasi tidak bisa ya mas, harus minta dulu ke keuangan	Kekurangan Aplikasi : - User Peternak memerlukan akses untuk melihat laporan keuangan

Tabel 4.8 dibawah merupakan hasil wawancara yang didapatkan penulis kepada administrasi keuangan selaku penjual didalam aplikasi.

Tabel 4.8 Hasil Wawancara kepada Admin

Pertanyaan	Jawaban	Keterangan
Bagaimana penggunaan aplikasi oleh pengelola selama ini?	Baik ya sudah dapat diakses dan digunakan dengan lancar	Sudah baik dan berjalan dengan lancar

Apakah aplikasi ini sudah sesuai dengan fungsi yang diharapkan?	Menurut saya sudah sesuai ya	Sudah sesuai
Bagaimana dengan kelebihan aplikasi?	Untuk kelebihannya pembuatan laporan tinggal download saja ya jadi tidak repot tulis di buku tangan	Kelebihan Aplikasi : - Laporan dapat langsung di download
Bagaimana dengan kekurangan aplikasi?	Untuk laporannya bisa dibikin untuk diakses semua penangkar ya mas, supaya semuanya dapat lihat	Kekurangan Aplikasi : - User Peternak memerlukan akses untuk melihat laporan keuangan