

BAB III METODE PENELITIAN

3.1 Jenis Penelitian

Penelitian ini menggunakan jenis penelitian kualitatif dimana data yang diperoleh berdasarkan observasi dan wawancara. Penelitian ini menggunakan jenis metode *waterfall* yang meliputi analisa kebutuhan, desain sistem, pengkodean, pengujian program, penerapan program dan perawatan program.

3.2 Waktu dan Tempat Penelitian

3.2.1. Waktu Penelitian

Penelitian ini dilakukan selama kurang lebih 4 bulan dengan rangkaian waktu seperti berikut:

Tabel 3. 1 Waktu Penelitian

Kegiatan	Bulan			
	Mei	Juni	Juli	Agustus
Pengumpulan Data				
Analisa Kebutuhan				
Perancangan Sistem				
Pembuatan Sistem				
Pengujian Sistem				
Maintenance				

3.2.2. Tempat Penelitian

Penelitian ini dilakukan di toko roti Gonzagy Cakes yang beralamat pada Jalan Tlogo Timur No. 4/8 Semarang.

3.3 Teknik Pengumpulan Data

3.3.1 Wawancara

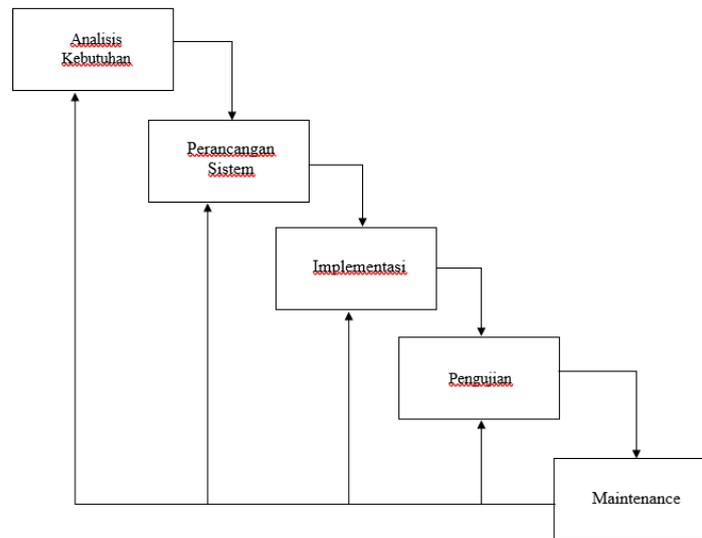
Pada wawancara ini bertujuan untuk memperoleh kendala apa saja yang dihadapi dan kebutuhan apa saja yang dibutuhkan toko roti Gonzagy Cakes

3.3.2 Observasi

Observasi dilakukan melalui pendekatan dan pengamatan langsung terhadap objek permasalahan yang terdapat di toko roti Gonzagy Cakes. Dalam upaya ini, fokus diberikan pada penelusuran mendalam untuk memahami secara lebih baik berbagai aspek yang berkaitan dengan situasi dan tantangan yang dihadapi oleh toko roti tersebut.

3.4 Metode Penelitian

Pada penelitian ini penulis menggunakan metode *waterfall* yang juga salah satu metode yang umum dan sering digunakan dalam pengembangan perangkat lunak. Berikut adalah proses pengembangan menggunakan metode waterfall



Gambar 3. 1 Metode *Waterfall*

3.4.1. Analisis Kebutuhan:

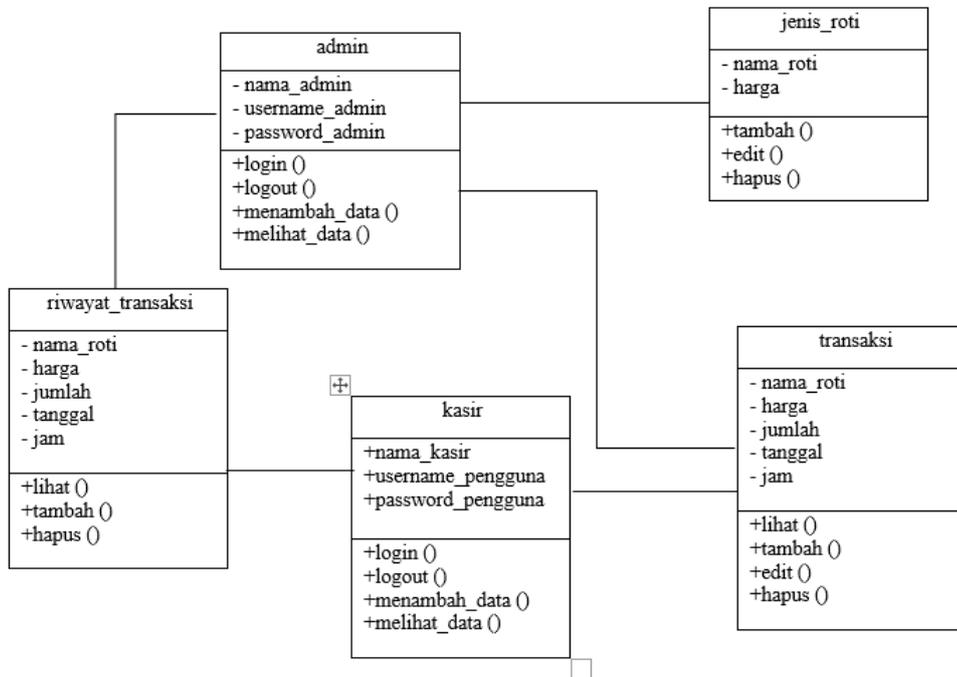
Pada tahap ini, dilakukan berbagai analisis mendalam terhadap kebutuhan dan harapan pemilik toko roti Gonzagy Cakes dalam pembuatan aplikasi penjualan berbasis web. Melalui wawancara dengan pemilik, akan diidentifikasi fitur-fitur yang diinginkan. Dalam konteks ini, penulis melakukan analisis terhadap segala kebutuhan terkait sistem penjualan, yang meliputi proses pengolahan data barang, data pengguna, dan data transaksi. Semua data ini direncanakan untuk disimpan dan dikelola melalui sebuah *database*, dengan tujuan utama untuk menyederhanakan proses penyimpanan data. Adapun hak akses pada sistem ini akan diatur dengan cermat, hanya pemilik yang akan diberikan akses penuh untuk melakukan tugas-tugas seperti penghitungan transaksi dan penambahan data barang. Di sisi lain, bagian kasir akan memiliki akses terbatas yang memungkinkan

mereka untuk menginput transaksi dan melihat riwayat transaksi sebelumnya.

3.4.2. Perancangan Sistem:

Pada tahap perancangan sistem, dilakukan pengembangan desain dengan fokus pada perancangan database yang diperlukan untuk mendukung keseluruhan sistem yang direncanakan. Penulis merancang struktur dan skema database yang tepat guna dan sesuai dengan rencana proyek. Dengan merancang database dengan cermat, sistem akan dapat mengorganisir, menyimpan, dan mengelola data secara efisien dan akurat. Perancangan sistem dibuat dengan UML (*Unified Modelling Language*). Diagram-diagram UML yang digunakan oleh penulis adalah *class diagram*, *activity diagram*, *use case diagram* dan *sequence diagram*

3.4.2.1. Class Diagram



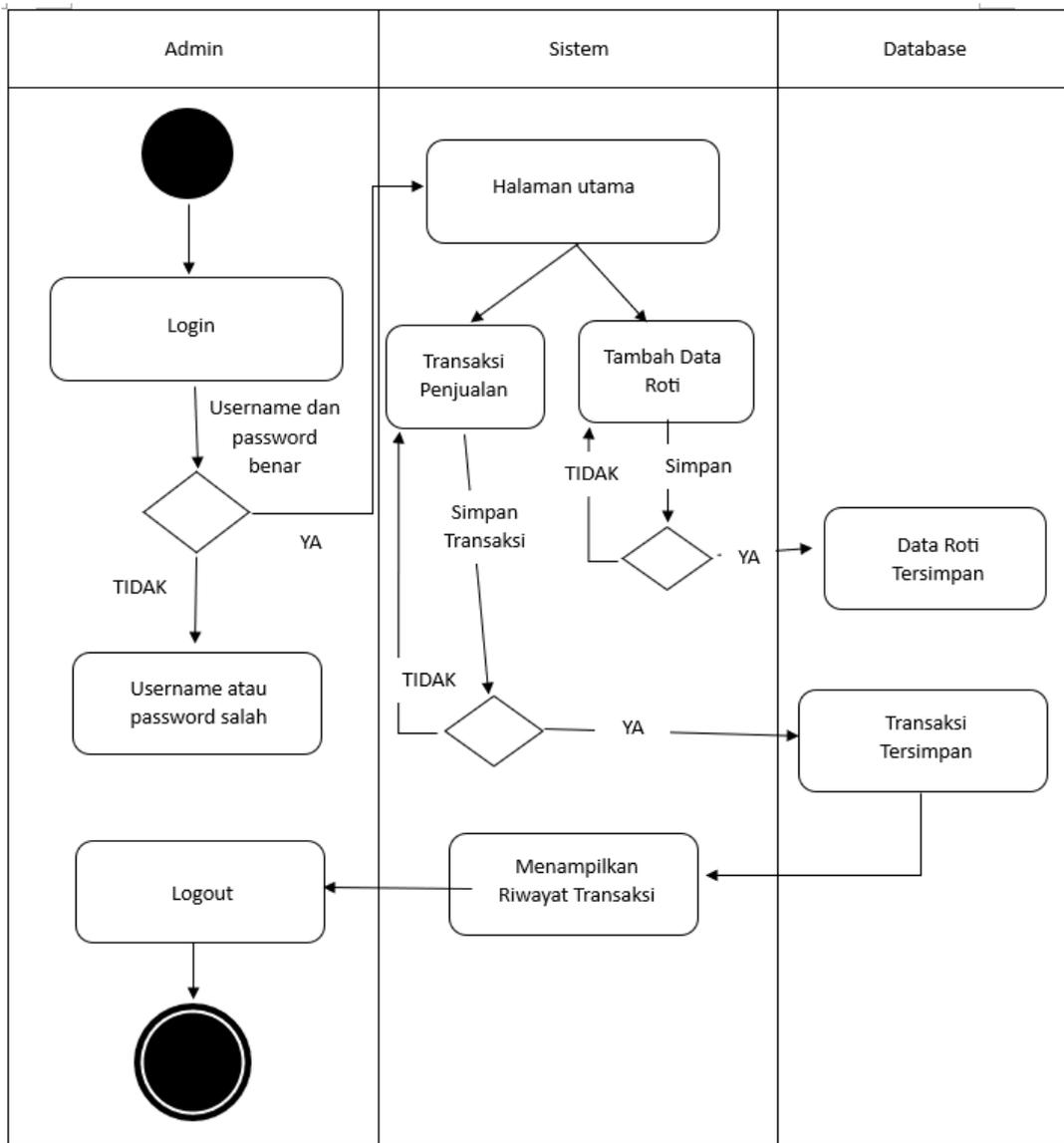
Gambar 3. 2 *Class Diagram* pada Admin pada toko roti Gonzagy Cakes

Dalam *Class diagram* diatas, terdapat empat kelas utama yang saling berinteraksi, yaitu kelas admin, kelas jenis_roti, kelas kasir, dan kelas transaksi. Kelas admin memiliki kewenangan untuk melakukan perubahan terhadap kelas jenis_roti, memungkinkan pengelolaan dan pengubahan jenis roti yang ada. Di sisi lain, kelas Kasir memiliki keterbatasan dalam hal ini, sehingga tidak memiliki izin untuk mengubah kelas jenis_roti. Namun, baik kelas admin maupun kasir memiliki kemampuan untuk melaksanakan aktivitas transaksi, termasuk melihat data riwayat transaksi. Hal ini berarti keduanya dapat melakukan proses penjualan melalui kelas Transaksi.

Sehingga, struktur dan interaksi antara keempat kelas ini menjadi dasar yang mendasari alur kerja sistem ini.

3.4.2.2. Activity Diagram

3.4.2.3 Activity Diagram Pada Admin



Gambar 3. 4 Activity Diagram pada Admin pada toko roti Gonzagy Cakes

Dalam Activity Diagram ini, proses dimulai saat admin melakukan tindakan login. Tahap ini melibatkan eksekusi verifikasi apakah username dan password yang dimasukkan benar atau tidak. Jika terjadi kesalahan, maka proses login harus diulang dengan memasukkan informasi yang tepat. Namun, jika informasi login benar, alur akan melanjutkan ke halaman utama. Dari sini, admin memiliki opsi untuk bergerak ke tahap transaksi penjualan.

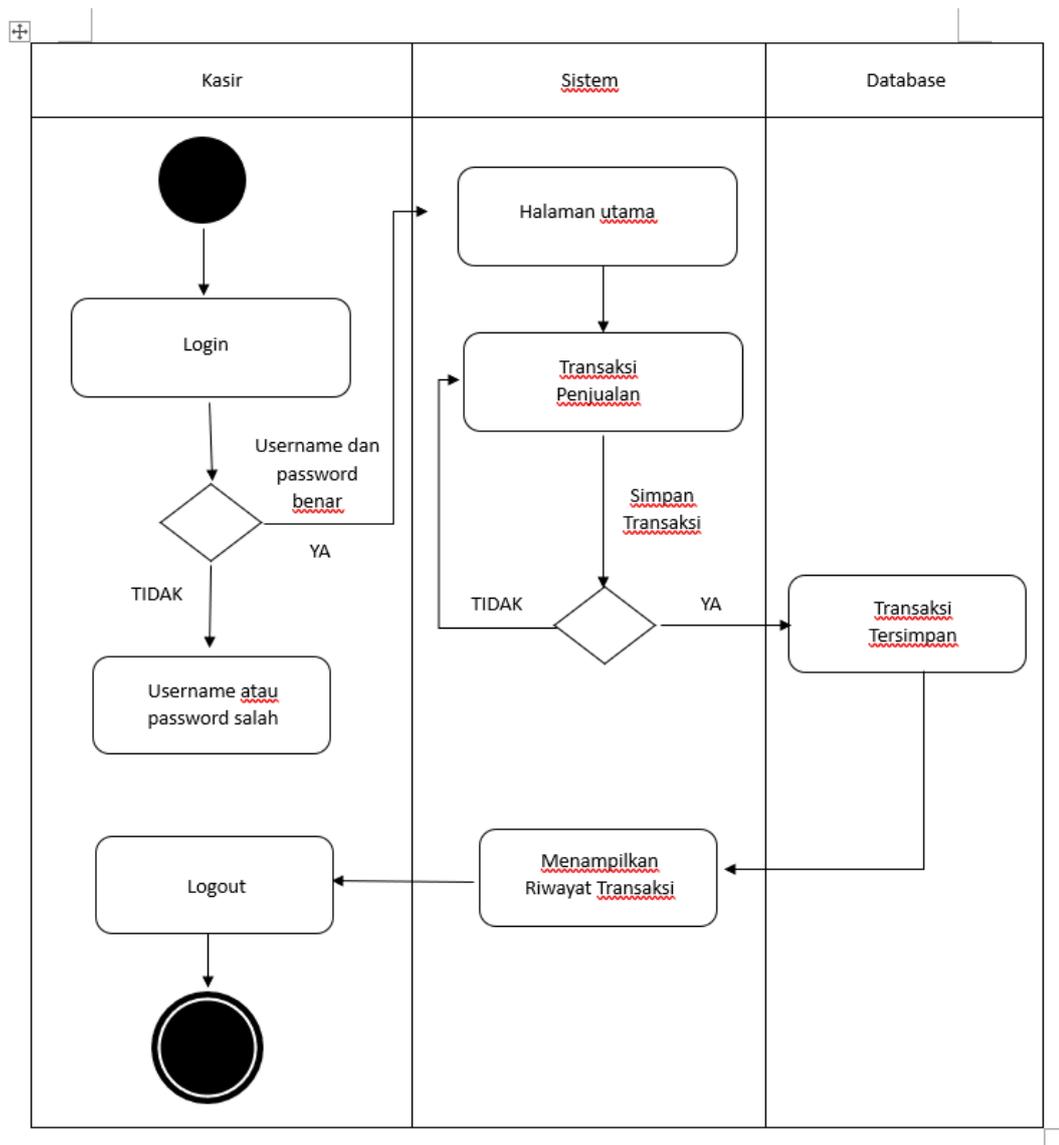
Pada tahap transaksi penjualan, terjadi percabangan. Admin dapat memutuskan untuk menyimpan transaksi atau tidak. Jika admin memilih untuk tidak menyimpan, maka proses akan kembali ke tahap transaksi penjualan untuk memungkinkan admin melakukan transaksi lainnya. Namun, jika admin memilih untuk menyimpan transaksi, data transaksi akan disimpan dan admin memiliki kemampuan untuk melihat riwayat transaksi yang telah terjadi.

Pada tahap tambah data roti, terjadi percabangan. Admin dapat memutuskan untuk menambahkan data atau tidak. Jika admin memilih untuk tidak menambahkan data, maka proses akan kembali ke tahap tambah data roti. Namun, jika admin memilih untuk menyimpan data, maka data akan tersimpan.

Dengan demikian, Activity Diagram ini menggambarkan rangkaian tindakan yang dilakukan oleh admin mulai dari login, navigasi ke halaman utama, melakukan transaksi penjualan, hingga pilihan untuk menyimpan atau tidak menyimpan transaksi dan akses ke riwayat transaksi. Diagram ini

menggambarkan alur kerja sistem secara visual, membantu pemahaman tentang interaksi dan keputusan yang terjadi dalam proses tersebut.

3.4.2.4 Activity Diagram Pada Kasir



Gambar 3. 5 Activity Diagram Pada Kasir pada toko roti Gonzagy Cakes

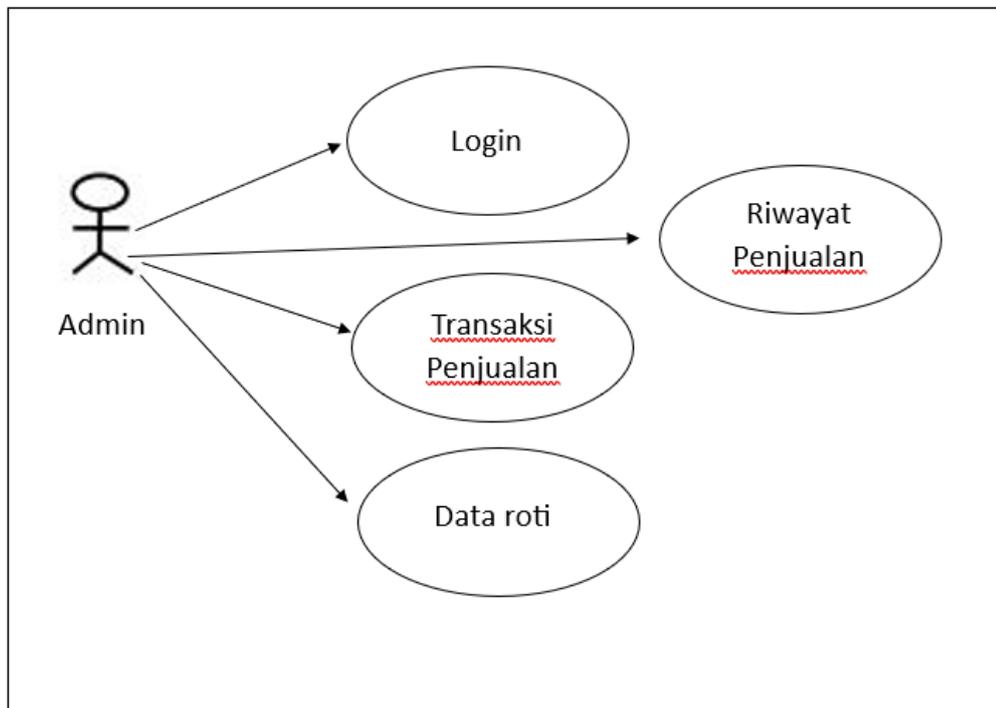
Dalam Activity Diagram ini, proses dimulai saat kasir melakukan tindakan login. Tahap ini melibatkan eksekusi verifikasi apakah username dan password yang dimasukkan benar atau tidak. Jika terjadi kesalahan, maka proses login harus diulang dengan memasukkan informasi yang tepat. Namun, jika informasi login benar, alur akan melanjutkan ke halaman utama. Dari sini, kasir memiliki opsi untuk bergerak ke tahap transaksi penjualan.

Pada tahap transaksi penjualan, terjadi percabangan. Kasir dapat memutuskan untuk menyimpan transaksi atau tidak. Jika kasir memilih untuk tidak menyimpan, maka proses akan kembali ke tahap transaksi penjualan untuk memungkinkan kasir melakukan transaksi lainnya. Namun, jika kasir memilih untuk menyimpan transaksi, data transaksi akan disimpan dan kasir memiliki kemampuan untuk melihat riwayat transaksi yang telah terjadi.

Dengan demikian, Activity Diagram ini menggambarkan rangkaian tindakan yang dilakukan oleh kasir mulai dari login, navigasi ke halaman utama, melakukan transaksi penjualan, hingga pilihan untuk menyimpan atau tidak menyimpan transaksi dan akses ke riwayat transaksi. Diagram ini menggambarkan alur kerja sistem secara visual, membantu pemahaman tentang interaksi dan keputusan yang terjadi dalam proses tersebut.

3.4.2.5 Use Case Diagram

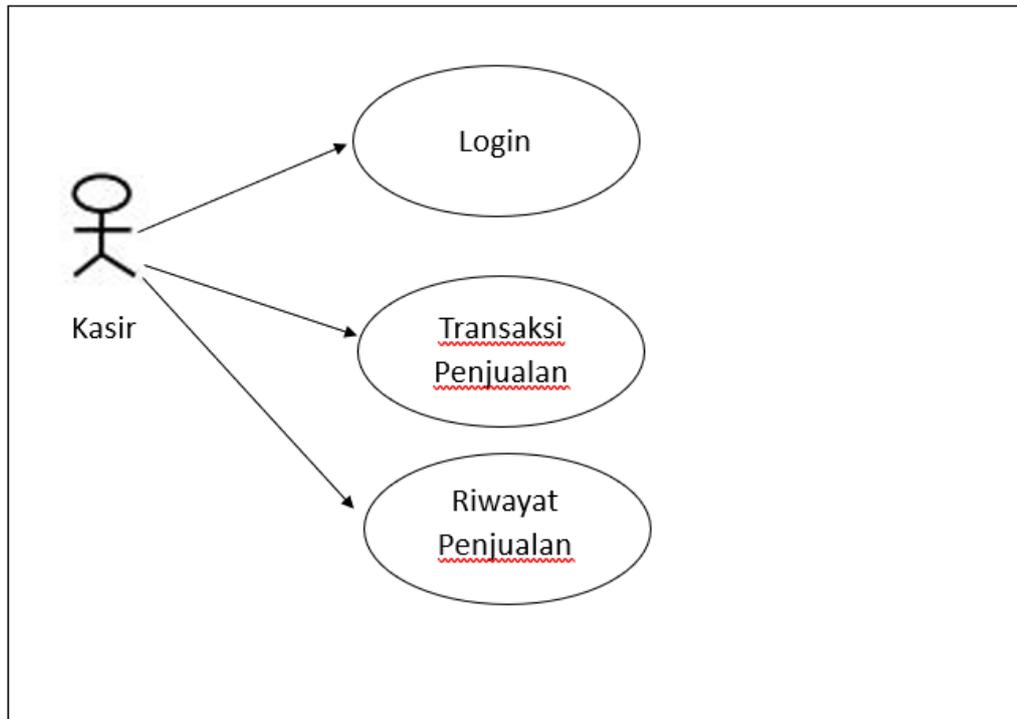
3.4.2.6 Use Case Diagram pada Admin



Gambar 3. 6 Use Case Diagram pada Admin pada toko roti Gonzagy Cakes

Dalam Use Case Diagram ini, Admin dapat melakukan proses login untuk mengakses sistem, serta memiliki kemampuan untuk melihat riwayat penjualan yang telah terjadi sebelumnya. Selain itu, admin juga dapat melakukan transaksi penjualan. Kemampuan lain yang dimiliki admin adalah kemampuan untuk menambahkan data roti.

3.4.2.7 Use Case Diagram pada Kasir



Gambar 3. 7 Gambar *Use Case Diagram* pada kasir di toko roti Gonzagy Cakes

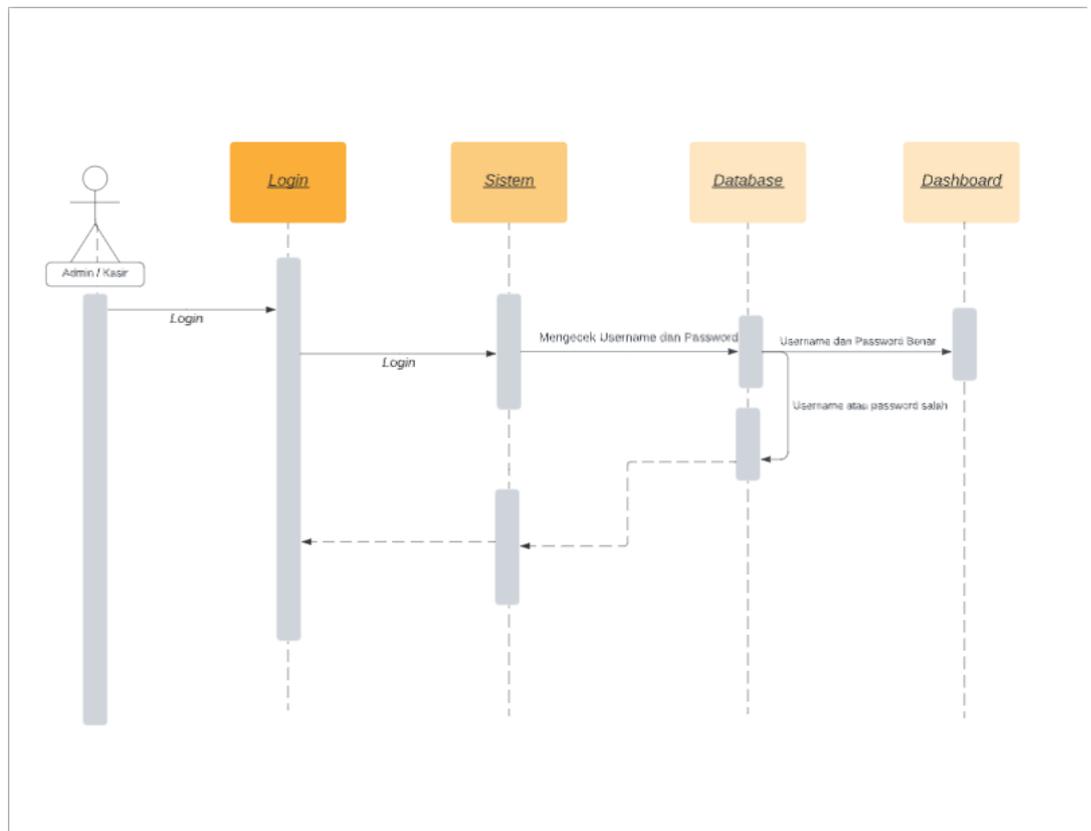
Dalam Use Case Diagram ini, kasir dapat melakukan proses login untuk mengakses sistem, serta memiliki kemampuan untuk melihat riwayat penjualan yang telah terjadi sebelumnya. Selain itu, kasir juga dapat melakukan transaksi penjualan.

3.4.2.8 Sequence Diagram

3.4.2.9 Sequence Diagram Login

Baik admin maupun kasir dapat melakukan proses login. Jika username dan password yang dimasukkan sesuai, sistem akan mengarahkan pengguna ke dashboard. Namun, jika terdapat kesalahan entah pada username atau password, sistem akan mengarahkan pengguna kembali ke

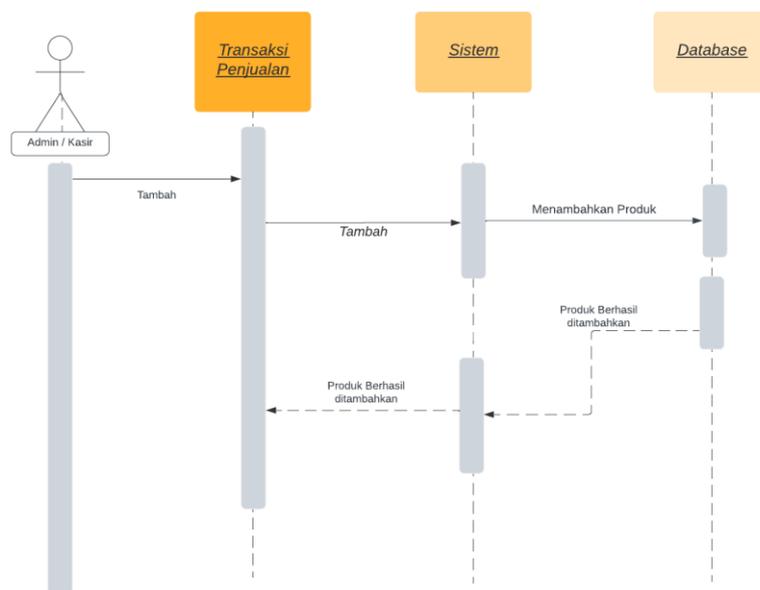
menu login untuk melakukan proses login ulang menggunakan informasi yang benar.



Gambar 3. 8 *Sequence Diagram Login*

3.4.2.10 Sequence Diagram Transaksi Penjualan

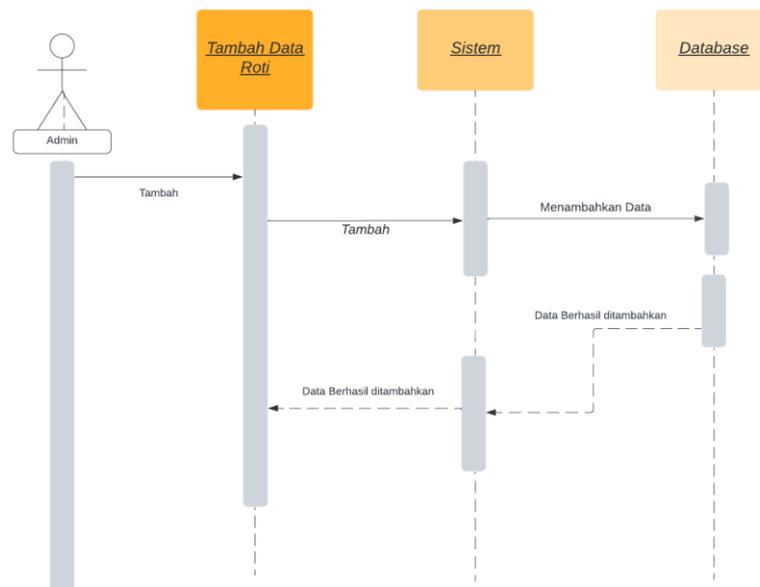
Admin atau kasir memiliki kemampuan untuk menambahkan transaksi penjualan. Setelah langkah penambahan produk dilakukan, rincian produk akan disimpan secara otomatis ke dalam database sistem.



Gambar 3. 9 Sequence Diagram Transaksi Penjualan

3.4.2.11 Sequence Diagram Tambah Data Roti

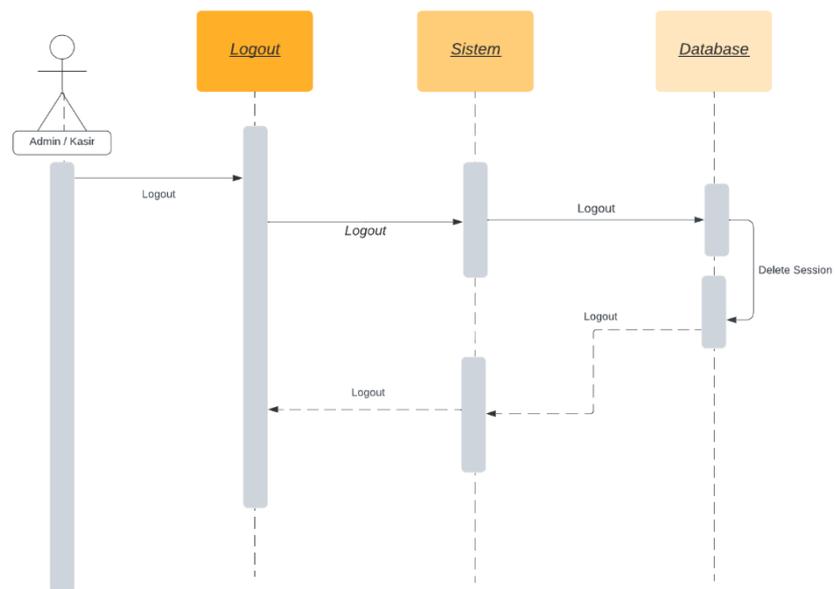
Pada sequence diagram ini hanya admin yang dapat menambahkan data roti. Setelah langkah penambahan produk dilakukan, rincian produk akan disimpan secara otomatis ke dalam database sistem.



Gambar 3. 10 Sequence Diagram Tambah Data Roti

3.4.2.12 Sequence Diagram Logout

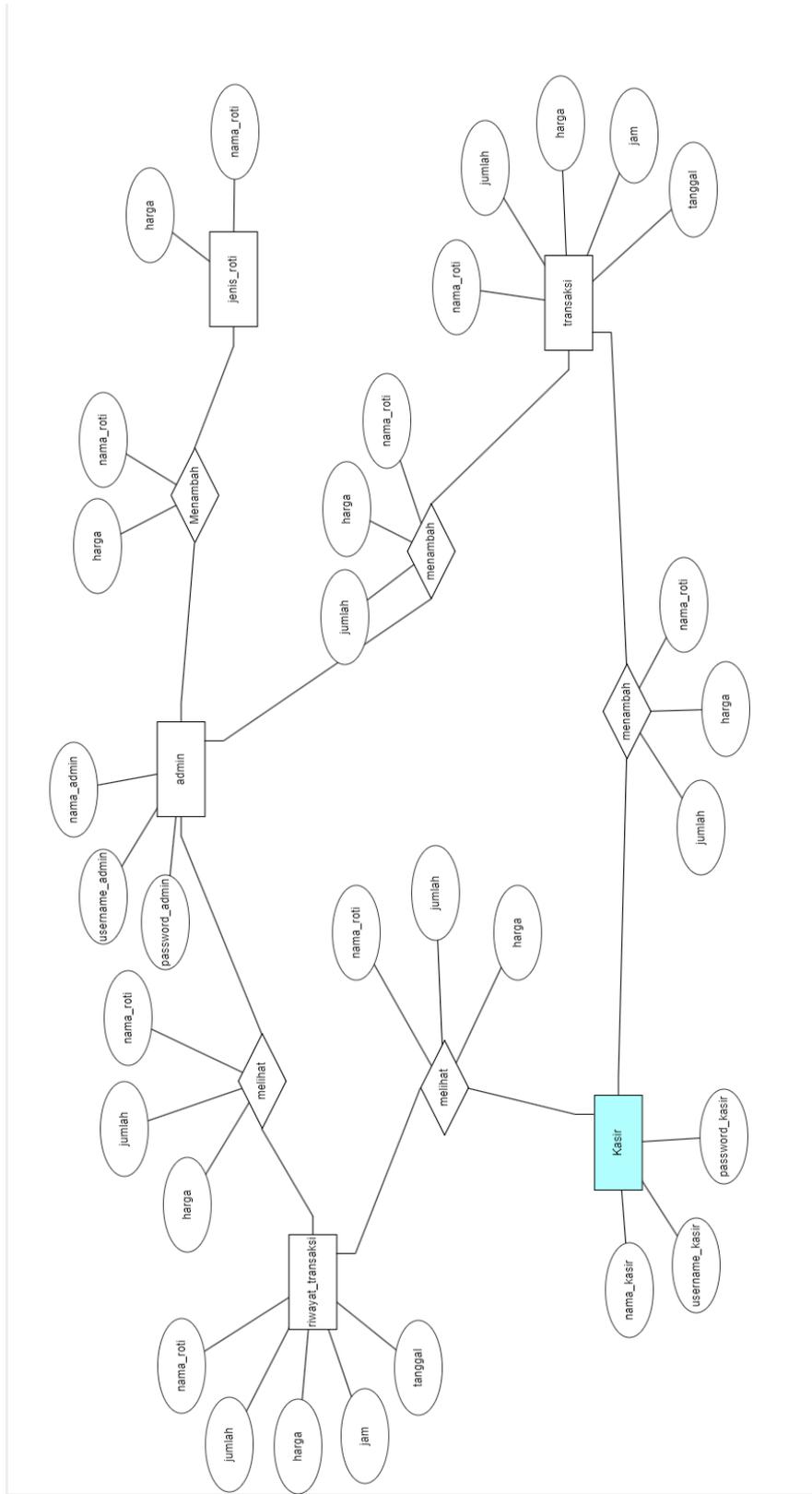
Pada sequence diagram ini, kedua peran, baik admin maupun kasir, memiliki kemampuan untuk melakukan proses logout.



Gambar 3. 11 Sequence Diagram Logout

3.4.2.13 ERD

Dalam ERD ini, terdapat fitur-fitur yang dapat diakses oleh admin dan kasir. Admin memiliki kemampuan untuk menambahkan data jenis_roti ke dalam sistem, serta mampu menambahkan transaksi dan melihat riwayat transaksi yang telah terjadi. Sementara itu, bagi kasir, fitur yang tersedia adalah kemampuan untuk menambahkan transaksi dan melihat riwayat transaksi yang juga mencakup data-data penjualan sebelumnya.



Gambar 3. 12 ERD

Pada gambar 3.12, admin memiliki kemampuan untuk menambahkan jenis-jenis roti dan transaksi, serta memiliki hak akses untuk melihat riwayat transaksi. Di sisi lain, peran kasir memiliki kewenangan untuk melakukan penambahan transaksi, dan juga diberi izin untuk mengakses riwayat transaksi yang ada.

3.4.3. Implementasi

Aplikasi sistem penjualan ini dibuat dengan menggunakan Bahasa pemrograman PHP dan framework CodeIgniter. Bahasa pemrograman PHP digunakan sebagai bahasa utama dalam pengembangan website ini, sementara framework CodeIgniter digunakan sebagai kerangka kerja untuk memudahkan proses pengembangan dan mempercepat waktu pengembangan. Dengan menggunakan framework CodeIgniter, pengembangan website dapat dilakukan dengan lebih efisien dan efektif, karena telah menyediakan banyak fitur yang siap digunakan, seperti manajemen database, pembuatan form, dan lain sebagainya.

3.4.4. Pengujian

Pada tahap ini dilakukan pengujian untuk memastikan aplikasi berjalan secara normal dan tidak ada kesalahan atau kerusakan. Pengujian *black box* testing merupakan salah satu dari pengujian unit, pengujian *black box testing* pada aplikasi penjualan difokuskan pada proses penginputan dan keluaran dari program penjualan. Pengujian testing ini digunakan untuk mengetahui keterbatasan dari program. Diharapkan dengan adanya pengujian ini aplikasi dapat running sesuai yang diinginkan.

Tabel 3. 2 contoh pengujian blackbox

Aktivitas Pengujian	Realisasi yang diharapkan	Hasil Pengujian	Kesimpulan
Masuk halaman awal	Terhubung dengan API Google Maps	Muncul Peta Kota Kediri.	[x] Diterima [] Ditolak
Klik Dropdown Filter criteria	Muncul kriteria sesuai dengan label	Dropdown dapat berfungsi sesuai yang diharapkan.	[x] Diterima [] Ditolak
Klik tombol “Cari Lokasi”	Muncul peta Lokasi sesuai kriteria yang dimasukkan	Tombol “Cari Lokasi” berfungsi sesuai yang diharapkan.	[x] Diterima [] Ditolak
Klik tombol “Cari Lokasi”	Muncul peta lokasi dengan warna yang berbeda.	Ada beberapa peta lokasi yang memiliki warna yang sama	[] Diterima [x] Ditolak
Klik pada peta lokasi	Muncul informasi detail lokasi pada setiap peta	Popup informasi berfungsi sesuai yang diharapkan	[x] Diterima [] Ditolak
Akses Melalui perangkat <i>smart phone</i> layar ≥ 5 inch	Tampilan sesuai dengan ukuran layar smartphone	Belum mampu menampilkan tampilan sesuai dengan layar smartphone	[] Diterima [x] Ditolak

3.4.5. Maintenance

Tahap terakhir dalam metode waterfall adalah maintenance atau pemeliharaan. Pada tahap ini, penulis bertanggung jawab melakukan pemeliharaan dan mengatasi masalah yang muncul saat sistem diluncurkan. Dengan cara pengecekan berkala pada sistem.