BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1. Analisis Sistem Yang Sedang Berjalan

Analisis sistem merupakan proses memilah-milah suatu permasalahan menjadi elemen-elemen yang lebih kecil untuk dipelajari guna mempermudah permasalahan dari suatu sistem informasi. Hasil akhir dari analisis sistem merupakan cara pemecahan masalah yang terjadi dalam spesifikasi sistem baru.

Pada tahap analisis diperlukan suatu pendekatan analisis guna menghindari kesalahan-kesalahan yang mungkin muncul pada tahap berikutnya, yaitu perancangan sistem baru. Karena pada tahap ini merupakan tahapan yang sangat penting, pendekatan yang dilakukan adalah mendefinisikan masalah pada sistem yang sedang berjalan dan sekaligus melakukan evaluasi setiap cara kerja sistem yang sedang berjalan berdasarkan prosedur-prosedur yang ada. Sehingga akan diketahui permasalahannya serta kesulitan apa saja yang dihadapi oeh sistem yang sedang berjalan, apa saja pengaruhnya dan harus diperhatikan validasinya terhadap sasaran sistem yang dirancang sebelum dilakukan perbaikan.

Tahap analisis sistem merupakan tahap yang sangat kritis dan penting karena semua kesalahan pada tahap berikutnya bersumber pada kesalahan tahap analisis sistem. Oleh karena itu seorang analisis sistem harus cermat memperhitungkan segala kemungkinan yang akan terjadi sedini mungkin.

Langkah dasar tahap analisis sistem adalah :

1. Mengidentikasi permasalahan yang terjadi *(identify)*
2. Memahami permasalahan*(understand)*
3. Menganalisis sistem*(analyze)*

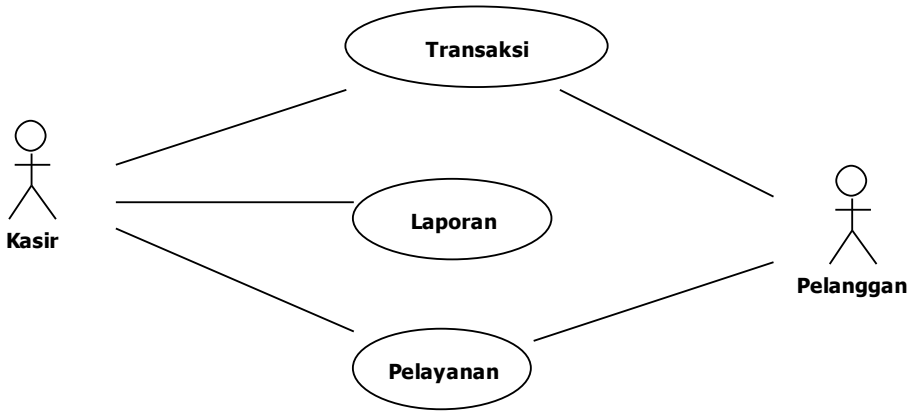
4. Membuat laporan hasil analis sistem *(report)*

4.1.1. Analisis Prosedur yang Sedang Berjalan

Pada analisis prosedur ini, harus diketahui prosedur yang sedang berjalan untuk perancangan sistem yang baru. Harus diketahui pula hal-hal yang menjadi tujuan pemakai sehingga masalah tersebut dapat didefinisikan secara jelas.

4.1.1.1. *UseCase* Diagram yang Sedang Berjalan

Berikut merupakan *UseCase* yang sedang sedang berjalan.



Gambar 4.1 *UseCase* Diagram yang berjalan

4.1.1.2. Definisi Aktor dan Deskripsi *UseCase* yang Sedang Berjalan

Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.

Tabel 4.1 Definisi Aktor dan Deskripsi yang Sedang Berjalan

No	Aktor	Deskripsi
1	Kasir	Divisi perusahaan yang mengelola transaksi penjualan pada perusahaan
2	Pelanggan	Pihak yang sudah mengenal dan pernah bertransaksi dengan Karnivor Restaurant dan bisa mendapatkan pelayanan.

4.1.1.3. Definisi *UseCase* dan Deskripsinya yang Sedang Berjalan

UseCase merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama *UseCase*. *UseCase* mendeskripsikan apa yang sistem (atau subsistem, kelas, atau antarmuka) kerjakan tetapi ia tidak menspesifikasikan (dan tidak memiliki kompetensi untuk menspesifikasi) bagaimana ia melakukannya.

Tabel 4.2 Definisi *UseCase* dan Deskripsi yang Sedang Berjalan

No	*UseCase*	Deskripsi
1	Transaksi	Proses yang terjadi antara kasir degan pelanggan, dimana yang sebelumnya pelanggan melakukan pemesanan menu makanan.
2	Laporan	Proses rekapitulasi transaksi yang telah dilakukan antara pelanggan dengan perusahaan.

3	Pelayanan	Strategi yang dilakukan perusahaan untuk mempertahankan pelanggan

4.1.1.4. Skenario *UseCase* yang Berjalan

Sekenario *UseCase* mendeskripsikan aktor-aktor yang melakukan prosedur dalam sistem, serta menjelaskan respon yang ditanggapi oleh sistem tersebut terhadap prosedur yang dilakukan oleh aktor. Berikut adalah sekenario *UseCase* yang berjalan saat ini, yaitu :

1. **Skenario *UseCase* Transaksi yang Berjalan**

   Interaksi antara aktor dengan *UseCase* Sistem yang dijelaskan dalam skenario *UseCase* sebagai berikut :

Tabel 4.3 Skenario *UseCase* Event yang Berjalan

Identifikasi	
Nomor	1
Nama	Transaksi
Tujuan	Untuk mengetahui pesanan pelanggan dan total harga yang harus dibayar
Aktor	Pelanggan
Worker	Perusahaan
Skenario Utama	

Aktor	Sistem
	1. Perusahaan memberikan sebuah tagihan pesanan
2. Pelanggan menerima tagihan pesanan.	
3. Membayar sejumlah uang atas pesanan yg dipesan.	
	4. Perusahaan memproses kuesioner yang telah diisi oleh pelanggan dan menyimpannya ke media penyimpanan.

2. Skenario *UseCase* Laporan yang Berjalan

Interaksi antara aktor dengan *UseCase* Laporan yang dijelaskan dalam skenario *UseCase* sebagai berikut :

Tabel 4.4 Skenario *UseCase* Event yang Berjalan

Identifikasi	
Nomor	2
Nama	Laporan
Tujuan	Untuk mengetahui banyaknya transaksi yang telah dilakukan
Aktor	Kasir

Worker	Perusahaan
Skenario Utama	
Aktor	Sistem
	1. Kasir mencetak laporan transaksi yang telah terjadi.
	2. Kasir memberikan laporan kepada atasan
3. Atasan mendapatkan laporan transaksi.	

3. Skenario *UseCase* Sistem Pelayanan yang Berjalan

Interaksi antara aktor dengan *UseCase* Sistem Pelayanan yang dijelaskan dalam skenario *UseCase* sebagai berikut :

Tabel 4.5 Skenario *UseCase* Event yang Berjalan

Identifikasi	
Nomor	3
Nama	Pelayanan
Tujuan	Untuk mempertahankan pelanggan yang telah melakukan transaksi
Aktor	Pelanggan
Worker	Kasir

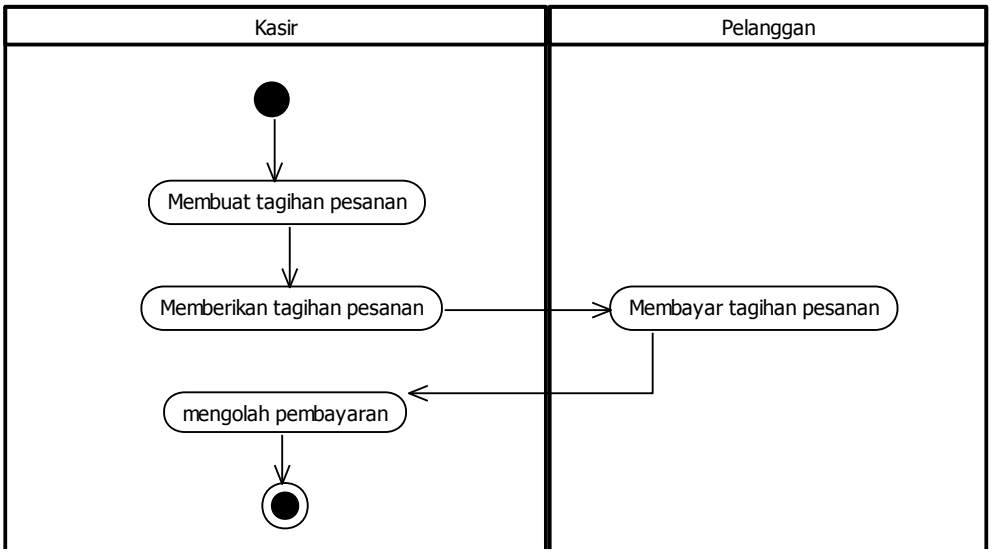
Skenario Utama	
Aktor	Sistem
1. Pelanggan melakukan transaksi	
	2. Perusahaan memberikan layanan berupa kupon undian.
3. Pelanggan menerima kupon undian	

4.1.1.5. Activity Diagram yang Berjalan

Activity diagram digunakan untuk menggambarkan kegiatan-kegiatan yang ada di dalam suatu sistem. Agar dapat lebih memahami tentang sistem yang akan dibuat, maka perlu dibuatkan *activity diagram* tentang sistem yang sedang berjalan. Adapun activity diagram yang sedang berjalan adalah sebagai berikut:

1. Activity Diagram Transaksi yang berjalan

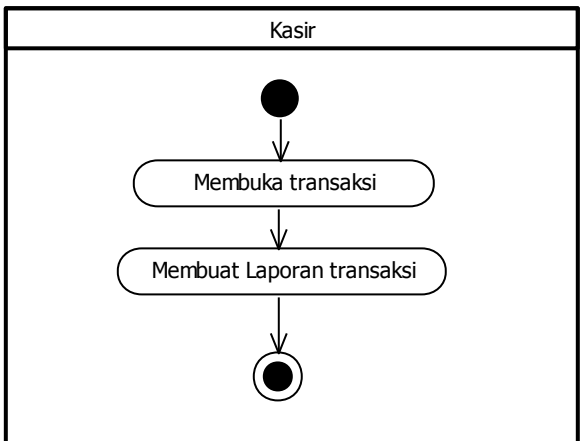
Diagram Aktivitas ini menunjukan aktivitas sistem transaksi yang berjalan, adalah sebagai berikut :



Gambar 4.2 *Activity Diagram* Transaksi yang sedang berjalan

2. Activity Diagram Laporan yang berjalan

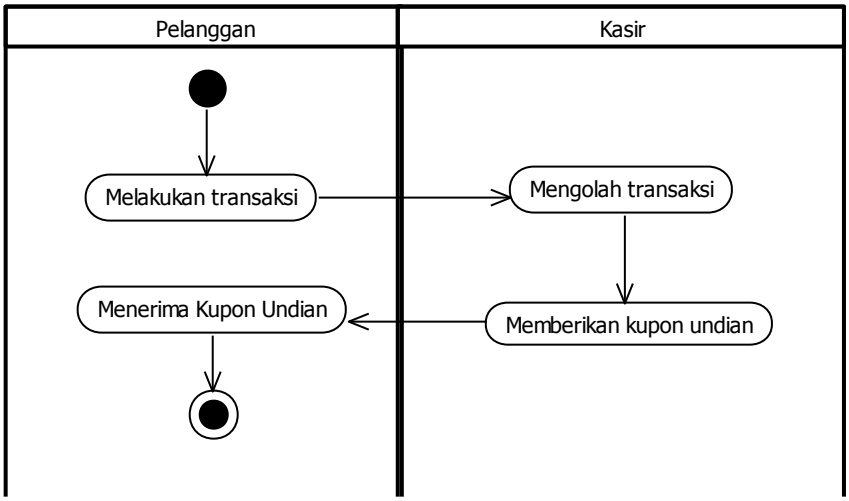
Diagram Aktivitas ini menunjukan aktivitas sistem laporan yang berjalan, adalah sebagai berikut :



Gambar 4.3 *Activity Diagram* Laporan yang sedang berjalan

3. Activity Diagram Pelayanan yang berjalan

Diagram Aktivitas ini menunjukan aktivitas Pelayanan yang berjalan, adalah sebagai berikut :



Gambar 4.4 *Activity Diagram* Pelayanan yang sedang berjalan

4.1.2. Evaluasi Sistem yang Sedang Berjalan

Melihat sistem pemasaran, event serta sistem pelayanan yang sedang berjalan di Karnivor Restaurant, yang mana masih menggunakan sistem manual, maka dapat disimpulkan beberapa kekurangan dalam prosedur yang sedang berjalan, diantaranya :

Tabel 4.6 Evaluasi Sistem yang Sedang Berjalan

No	Permasalahan	Rancangan Permasalahan
1	Proses Transaksi belum dimanfaatkan seutuhnya oleh	Membangun sebuah rancangan dimana pelanggan yang akan melakukan transaksi ditawarkan

	perusahaan untuk mendapatkan biodata pelanggan	menjadi member, sehingga bisa mendapatkan biodata pelanggan.
2	Proses Pelayanan masih menggunakan program yang kurang maksimal	Dibuatkan sebuah program pasca transaksi yang ter otomatisasi
3	Belum adanya sebuah program yang bisa menjaga pelanggan supaya tetap loyal dimana setelah terjadi transaksi antara perusahaan dengan pelanggan.	Dibangun sebuah program dimana bisa memelihara pelanggan pasca transaksi supaya tidak berpindah ke pesaing.

4.2. Perancangan Sistem

Perancangan sistem dilakukan apabila tahap dari analisis sistem telah dilakukan. Berdasarkan pada hasil analisis sistem yang sedang berjalan, maka diusulkan perancangan sistem baru, dimana kerja dari satu sistem yang baru diharapkan dapat mengatasi beberapa permasalahan yang timbul.

4.2.1. Tujuan Perancangan Sistem

Perancangan sistem bertujuan untuk memberikan gambaran yang jelas dan dan rancang bangun yang sesuai dengan kebutuhan *user* atau pemakai sistem itu sendiri. Adapun tujuan dari perancangan adalah :

1. Bisa mendapatkan biodata pelangan ketika melakukan transaksi.
2. Dapat mempermudah dalam pengelolaan data pelanggan yang telah didapatkan.
3. Dapat menghasilkan sebuah program yang akan dilakukan ketika pasca transaksi

4.2.2. Gambaran Umum Sistem yang Diusulkan

Aplikasi pelayanan pasca transaksi dengan pemanfaatan *Sms Gateway* ini dapat digambarkan sebagai sebuah fasilitas yang bisa memelihara pelanggan yang telah melakukan transaksi dengan karnivor restaurant.

Selain itu aplikasi yang akan dirancang ini mendukung untuk melakukan pengiriman tawaran menarik yang diberikan perusahaan kepada pelanggan yang telah melakukan transaksi berdasarkan nominal transaksi dan hari ulangtahun pelanggan.

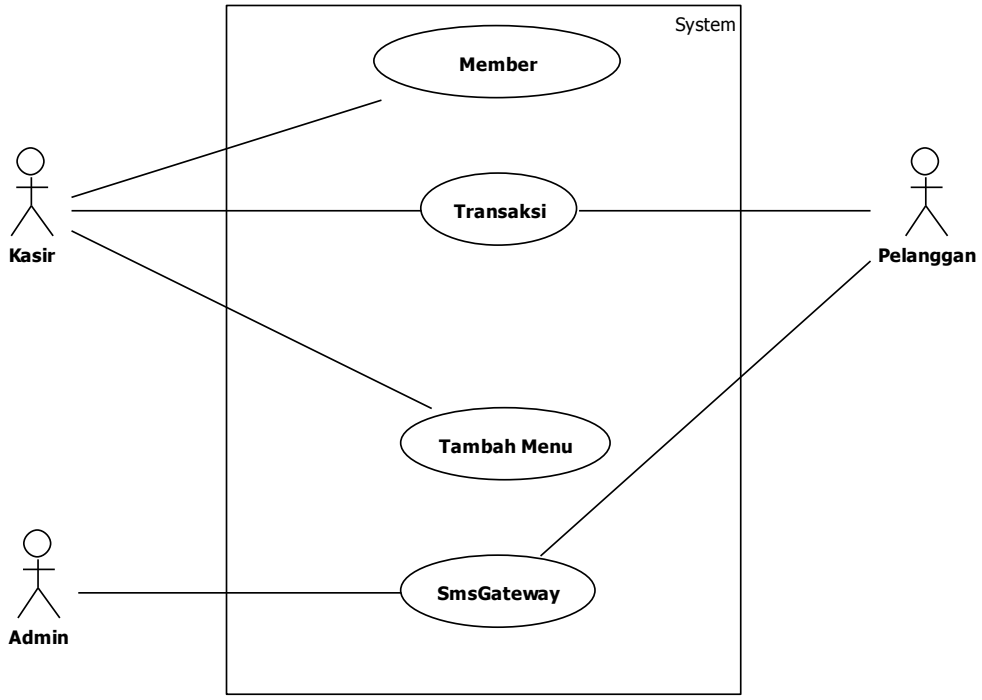
Dengan adanya aplikasi perancangan pelayanan pasca transaksi dengan pemanfaatan *Sms Gateway* diharapkan memudahkan perusahaan dalam menjaga pelanggan supaya tetap loyal terhadap perusahaan.

4.2.3. Perancangan Prosedur yang Diusulkan

Perancangan ini mencakup *Use Case Diagram, Activity Diagram, Collaboration Diagram, Class Diagram, Sequence Diagram, Component Diagram dan Deployment Diagram* yang akan menghasilkan sistem yang lebih baik. Proses yang dirancang diuraikan menjadi beberapa bagian yang dapat membentuk sistem tersebut menjadi satu kesatuan komponen.

4.2.3.1. *UseCase* Diagram yang Diusulkan

Diagram *UseCase* memperlihatkan hubungan-hubungan yang terjadi antara aktor-aktor dengan *UseCase* dalam sistem, dan berikut *UseCase* yang diusulkan:



Gambar 4.5 *UseCase* Diagram yang Diusulkan

4.2.3.2. Definisi Aktor yang Diusulkan

Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.

Tabel 4.7 Definisi Aktor dan deskripsinya yang Diusulkan

4.2.3.3. Definisi *UseCase* yang Diusulkan

UseCase merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata

No	Aktor	Deskripsi
1	Admin	Pihak yang mengelola program pasca transaksi yang telah terjadi antara pelanggan dengan konsumen.
2	Pelanggan	Pihak yang melakukan tansaksi dengan restaurant.
3	Kasir	Pihak yang melakukan transaksi dengan pelanggan, yang membuat laporan transaksi restaurant.

kerja diawal frase nama *UseCase*. *UseCase* mendeskripsikan apa yang sistem (atau subsistem, kelas, atau antarmuka) kerjakan tetapi ia tidak menspesifikasikan (dan tidak memiliki kompetensi untuk menspesifikasi) bagaimana ia melakukannya.

Tabel 4.8 Definisi *UseCase* dan deskripsinya yang Diusulkan

No	*UseCase*	Deskripsi
1	Member	Proses pendaftaran pelanggan yang dilakukan oleh kasir sebagai tindakan untuk mendapatkan biodata pelanggan
2	Transaksi	Untuk mengelola transaksi pemesanan yang telah dilakukan pelanggan.

3	Penjualan	Proses jual beli yang dilakukan pelanggan dengan restaurant
4	Tambah menu	Penambahan data menu makanan yang dilakukan oleh kasir untuk diolah dalam transaksi
5	*Sms Gateway*	Program yang diadakan perusahaan untuk menjaga pelanggan supaya tetap loyal dan program ini dilakukan setelah pasca transaksi.

4.2.3.4. Skenario *UseCase* yang Diusulkan

Sekenario *UseCase* mendeskripsikan aktor-aktor yang melakukan prosedur dalam sistem, serta menjelaskan respon yang ditanggapi oleh sistem tersebut terhadap prosedur yang dilakukan oleh aktor. Berikut adalah sekenario *UseCase* yang diusulkan, yaitu :

1. Skenario *UseCase* Pemasaran yang Diusulkan

Interaksi antara aktor dengan *UseCase* pendaftaran yang dijelaskan dalam skenario *UseCase* sebagai berikut:

Tabel 4.9 Skenario *UseCase* Pemasaran yang Diusulkan

Identifikasi	
Nomor	1
Nama	Member
Tujuan	Untuk mencatat data pelanggan

Aktor	Kasir,Pelanggan
Worker	Sistem
Skenario Utama	
Aktor	Sistem
1. Kasir Memberikan form pendaftaran member	
2. Pelanggan mengisi form pendaftaran member	
3. Pelanggan memberikan form pendaftaran member kepada kasir	
4. Kasir menginputkan data pelanggan	
	5. Cek kelengkapan data
	6. Simpan ke database
Skenario Alternatif (data produk yang akan dipasarkan tidak lengkap)	
	1. Mengecek kelengkapan data
	2. Menampilkan alert “data belum lengkap”
3. Mengisi kelengkapan data	

	4. Cek kelengkapan data
	5. Simpan data kedalam database

2. Skenario *UseCase* Transaksi yang Diusulkan

Interaksi antara aktor dengan *UseCase* transaksi yang dijelaskan dalam skenario *UseCase* sebagai berikut:

Tabel 4.10 Skenario *UseCase* Pemasaran yang Diusulkan

Identifikasi	
Nomor	2
Nama	Transaksi
Tujuan	Untuk mengelola pemesanan produk.
Aktor	Kasir
Worker	Sistem
Skenario Utama	
Aktor	Sistem
1. Kasir menginputkan pesanan pelanggan	
	2. Mengecek kelengkapan data
	3. Menyimpan data ke database.

	4. Menampilkan data transaksi yang telah di inputkan
Skenario Alternatif (data produk yang akan dipasarkan tidak lengkap)	
	5. Mengecek kelengkapan data
	6. Menampilkan alert “data belum lengkap”
7. Mengisi kelengkapan data	
	8. Cek kelengkapan data
	9. Simpan data kedalam database

3. Skenario *UseCase* Pemesanan yang Diusulkan

Interaksi antara aktor dengan *UseCase* event yang dijelaskan dalam skenario *UseCase* berikut :

Tabel 4.11 Skenario *UseCase* Pemesanan yang Diusulkan

Identifikasi	
Nomor	3
Nama	Menambahkan Menu
Tujuan	Mengupdate data menu makanan yang ada
Aktor	Kasir
Worker	Sistem

Skenario Utama	
Aktor	Sistem
1. Masuk ke halaman Menu	
	2. Menampilkan halaman menu
3. Menginputkan data menu makanan	
	4. Cek kelengkapan data
	5. Simpan data ke database.
Skenario Alternatif (data produk yang akan dipasarkan tidak lengkap)	
	1. Menampilkan alert “data tidak lengkap.
2. Melengkapi data yang tidak lengkap	
	3. Cek kelengkapan data
	4. Simpan data kedalam database

4. Skenario *UseCase* Laporan yang Diusulkan

Interaksi antara aktor dengan *UseCase* event yang dijelaskan dalam skenario *UseCase* berikut :

Tabel 4.12 Skenario *UseCase Event* yang Diusulkan

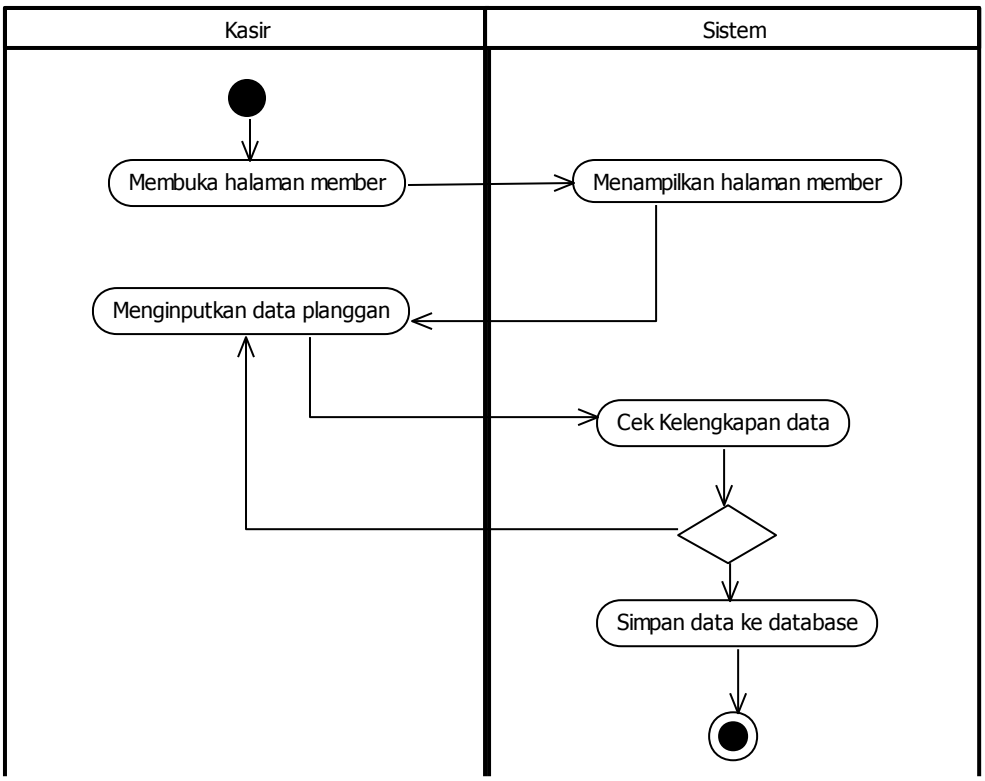
Identifikasi	
Nomor	4
Nama	Sms Gate Way
Tujuan	Menjaga pelanggan supaya tidak berpindah kepesaing
Aktor	Admin
Worker	Sistem
Skenario Utama	
Aktor	Sistem
1. Membuka halaman *Sms Gateway*	
	2. Menampilkan halaman *Sms Gateway*
3. Mengoperasikan *Sms Gateway*	

4.2.3.5. Activity Diagram yang Diusulkan

Activity diagram digunakan untuk menggambarkan kegiatan-kegiatan yang ada di dalam suatu sistem. Agar dapat lebih memahami tentang sistem yang dibuat, maka perlu dibuatkan *activity diagram* tentang sistem yang akan diusulkan. Adapun activity diagram yang sedang diusulkan adalah sebagai berikut:

1. Activity Diagram Tambah member yang Diusulkan

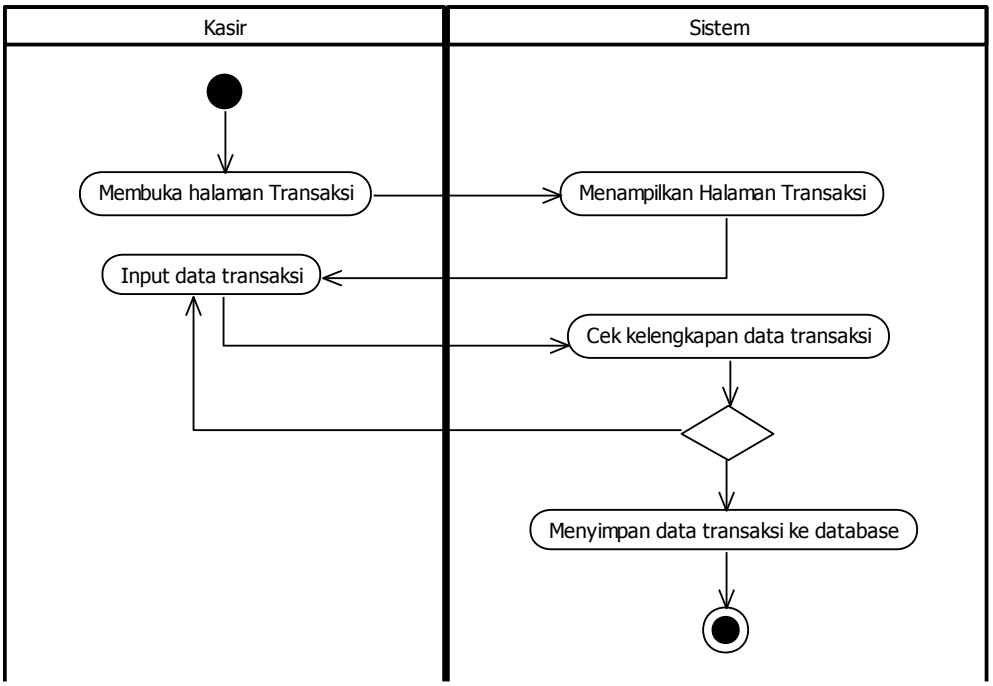
Activity diagram ini menunjukan aktivitas tambah member yang diusulkan, adalah sebagai berikut :



Gambar 4.6 *Activiy Diagram* Pendaftaran member yang Diusulkan

2. Activity Diagram Transaksi yang Diusulkan

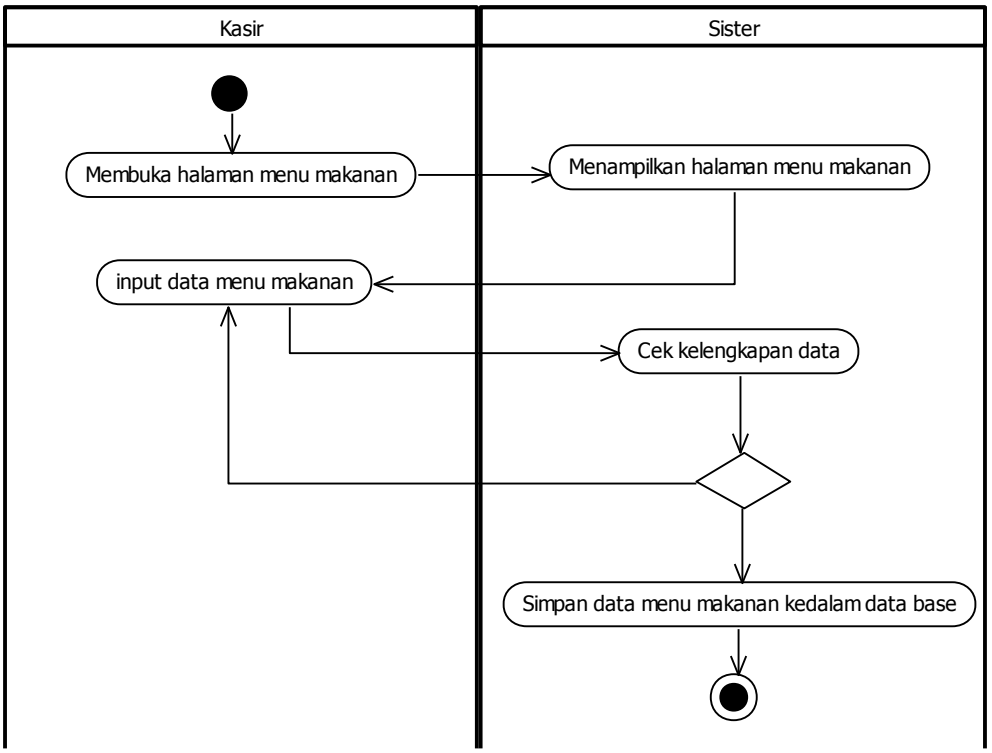
Activity diagram ini menunjukan aktivitas Event yang diusulkan, adalah sebagai berikut :



Gambar 4.7 *Activiy Diagram* Transaksi yang Diusulkan

3. Activity Diagram Tambah menu makanan yang Diusulkan

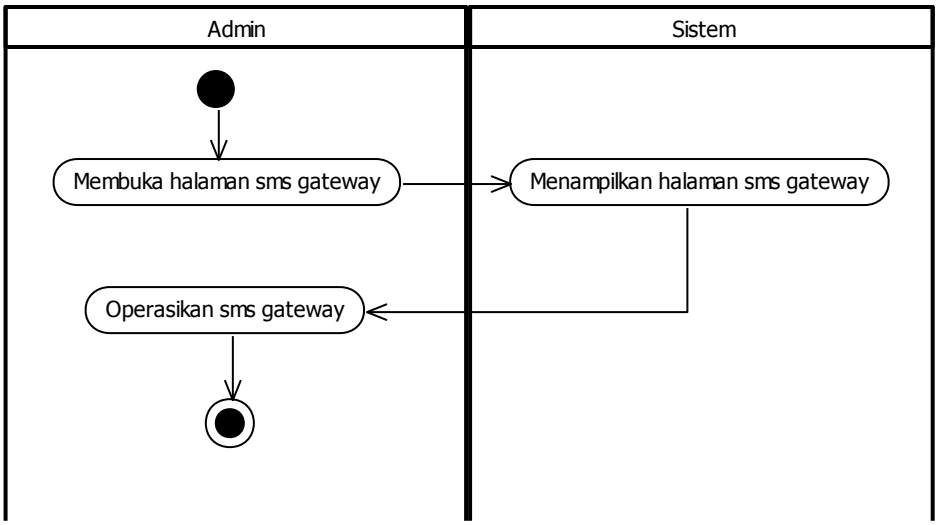
Activity diagram ini menunjukan aktivitas tambah menu makanan yang diusulkan, adalah sebagai berikut :



Gambar 4.8 *Activiy Diagram* Transaksi yang Diusulkan

4. Activity Diagram *Sms Gateway* yang diusulkan

Diagram Aktivitas ini menunjukan aktivitas *Sms Gateway* yang diusulkan adalah sebagai berikut ini :



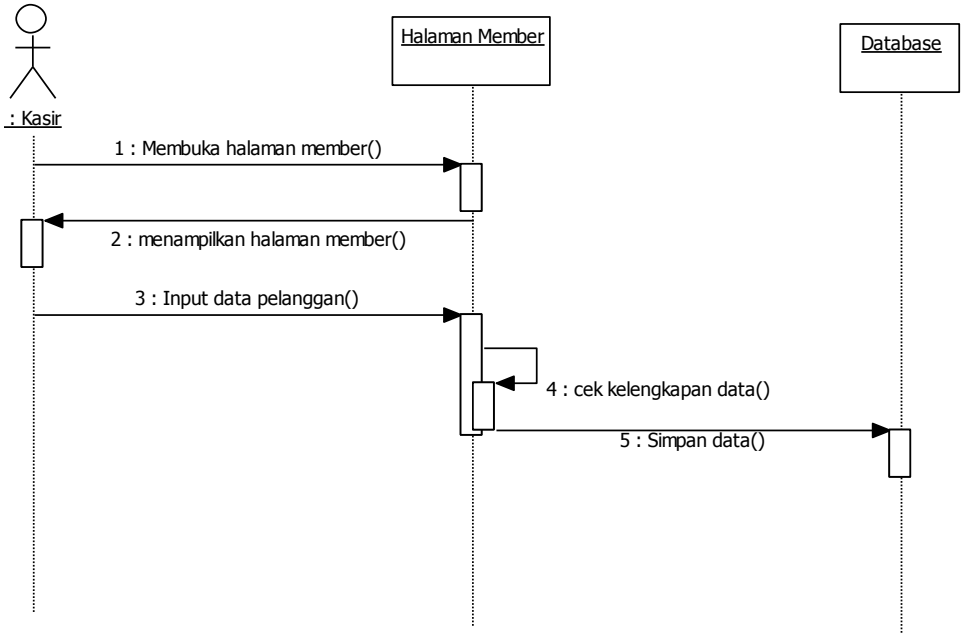
Gambar 4.9 *Activiy Diagram Sms Gateway* yang Diusulkan

4.2.3.6. Sequence Diagram yang Diusulkan

Sequence Diagram digunakan untuk menggambarkan interaksi antar objek dalam waktu yang berurutan. Tetapi pada dasarnya *Sequence Diagram* selain digunakan dalam lapisan abstraksi model objek, kegunaannya untuk menunjukan rangkaian pesan yang dikirim antar *object* juga interaksi antara *object*. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. Komponen utama *Sequence diagram* terdiri atas objek yang dituliskan dengan kotak segiempat bernama pesan diwakili oleh garis dengan tanda panah dan waktu yang ditunjukan dengan proses vertikal. Berikut adalah *sequence diagram* yang diusulkan, yaitu:

1. Sequence Diagram Tambah member yang Diusulkan

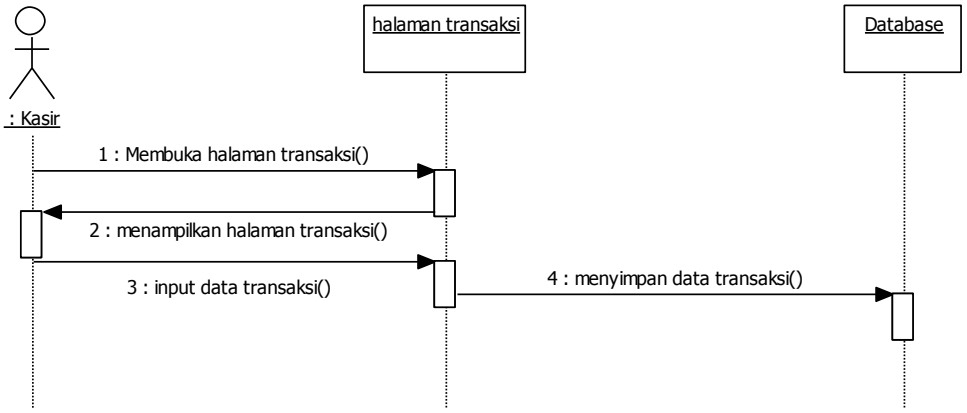
Berikut merupakan *sequence diagram* untuk mengelola member. Adapun *sequence diagramnya* sebagai berikut :



Gambar 4.10 *Sequence Diagram* Penambahan member yang Diusulkan

2. Sequence Diagram Transaksi yang Diusulkan

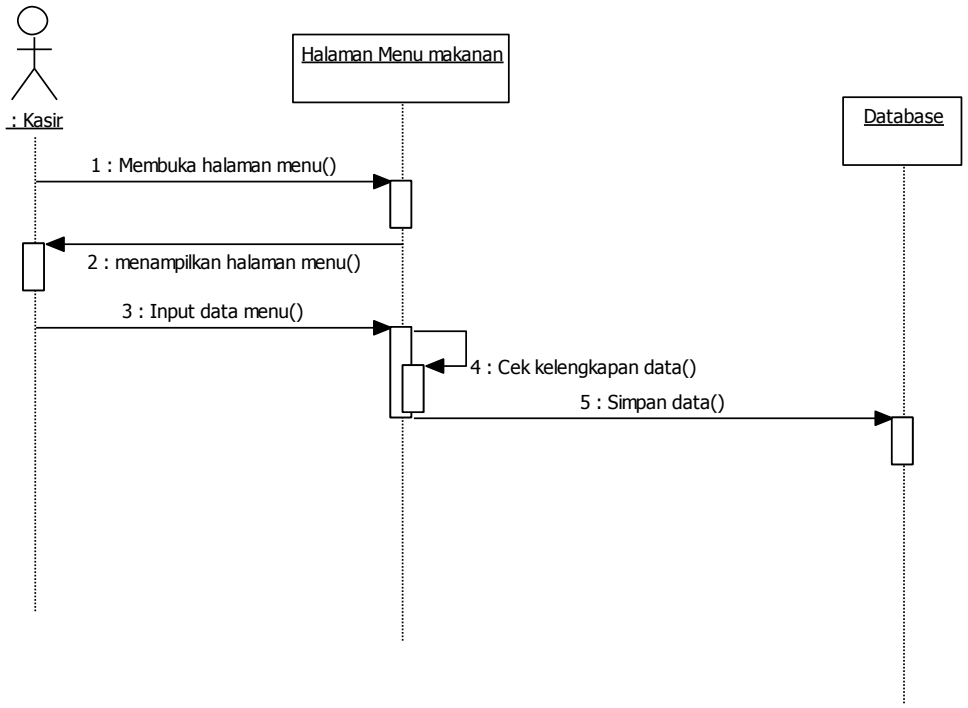
Berikut merupakan *sequence diagram* untuk mengelola Transaksi. Adapun *sequence diagramnya* sebagai berikut :



Gambar 4.11 *Sequence Diagram* Transaksi yang Diusulkan

3. Sequence Diagram Tambah menu

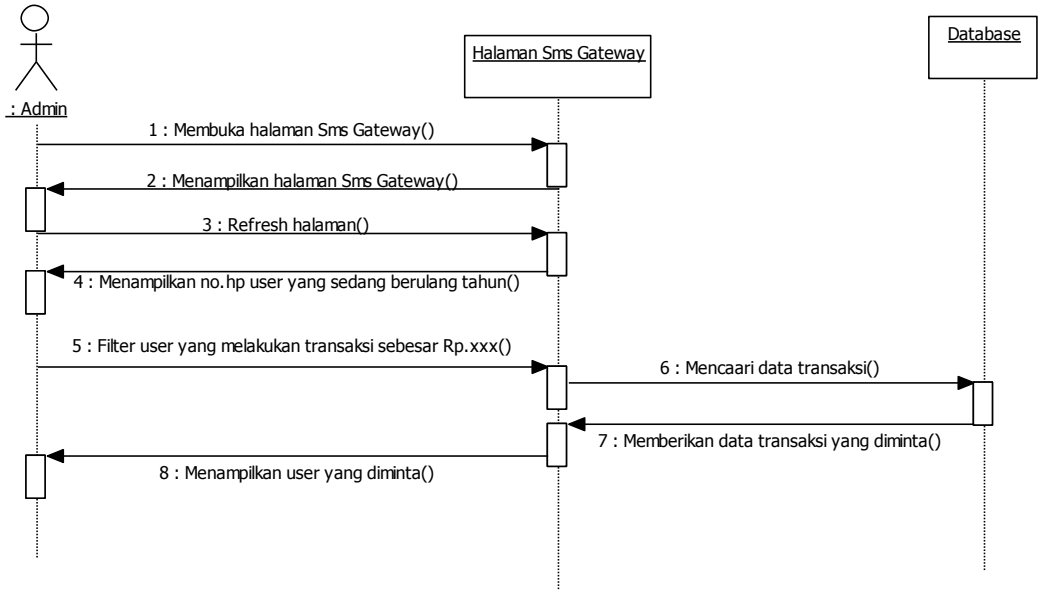
Berikut merupakan *sequence diagram* untuk mengelola Tamba Menu. Adapun *sequence diagramnya* sebagai berikut :



Gambar 4.12 Sequence Diagram *Sms Gateway*

4. Sequence Diagram *Sms Gateway*

Berikut merupakan *sequence diagram* untuk mengelola *Sms Gateway*. Adapun *sequence diagramnya* sebagai berikut :



Gambar 4.13 Sequence Diagram *Sms Gateway*

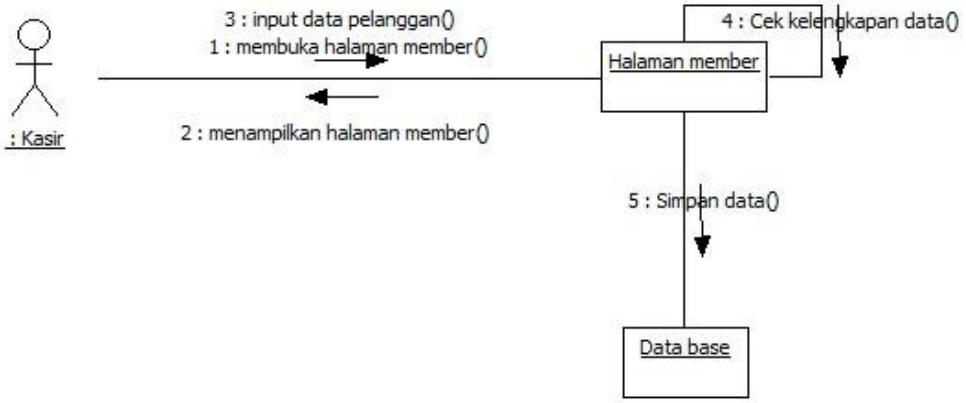
4.2.3.7. Collaboration Diagram

Seperti Sequence Diagram, Collaboration diagram juga digunakan untuk memperhatikan aliran-aliran pada *UseCase.* Sementara sequence diagram berurutan menurut waktu, collaboration diagram berfokus pada relasi-relasi yang terjadi antara objek yang satu dengan objek-objek yang lainya.

Adapun Collaboration diagram yang diusulkan adalah sebagai berikut :

1. **Collaboration diagram Tambah member**

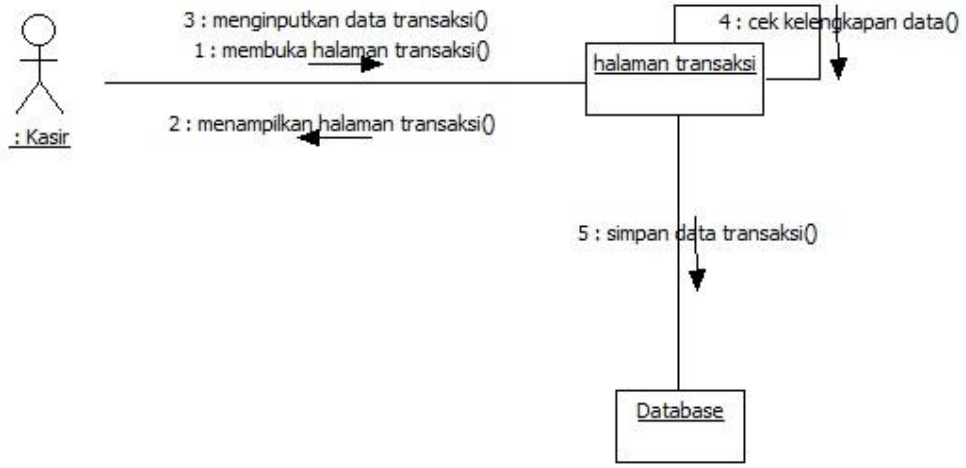
   Berikut merupakan collaboration diagram Tambah member, adapun gambarnya sebagai berikut :



Gambar 4.14 Collaboration Diagram Tambah member

2. Collaboration Diagram transaksi yang Diusulkan

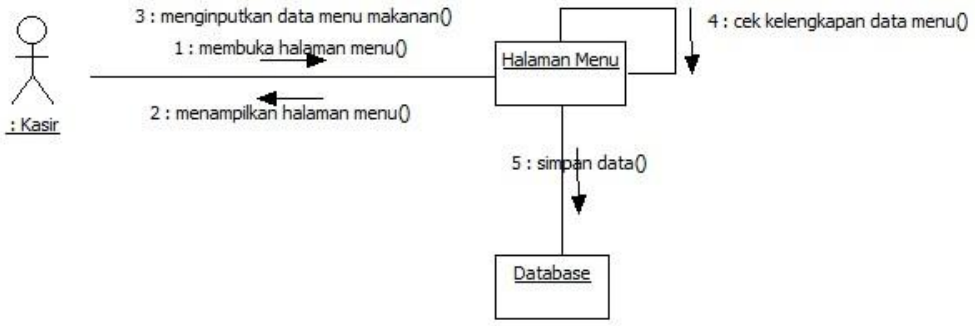
Berikut merupakan Collaboration Diagram transaksi yang diusulkan. Adapun gambarnya sebagai berikut :



Gambar 4.15 *Collaboration Diagram* Transaksi yang Diusulkan

3. Collaboration Diagram tambah menu yang Diusulkan

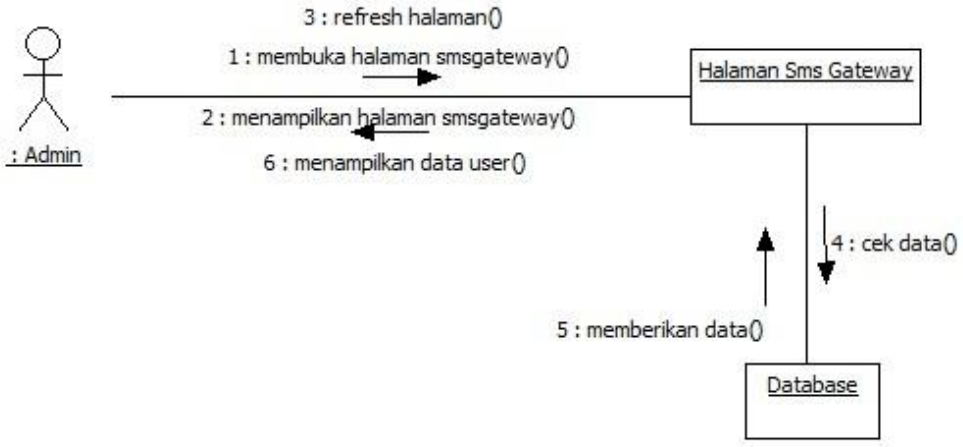
Berikut merupakan Collaboration Diagram tambah menu makanan yang diusulkan. Adapun gambarnya sebagai berikut :



Gambar 4.16 *Collaboration Diagram* Tambah menu yang Diusulkan

4. Collaboration Diagram SmsGateway yang Diusulkan

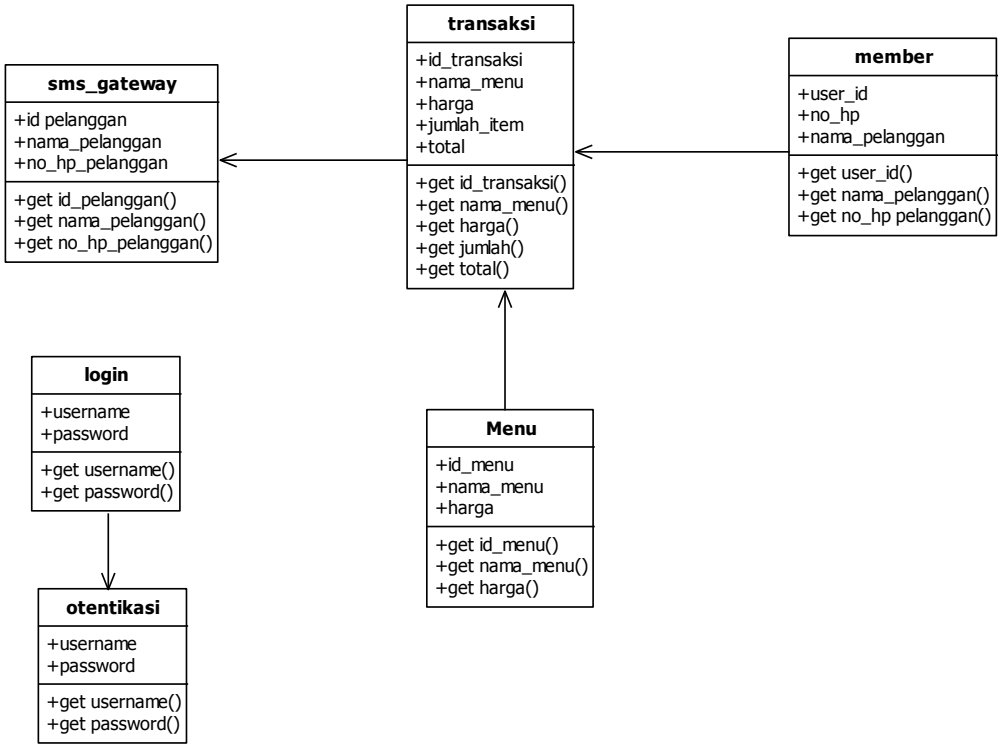
Berikut merupakan Collaboration Diagram pemasaran yang diusulkan. Adapun gambarnya sebagai berikut :



Gambar 4.17 *Collaboration Diagram* Sms Gateway yang Diusulkan

4.2.3.8. Class Diagram

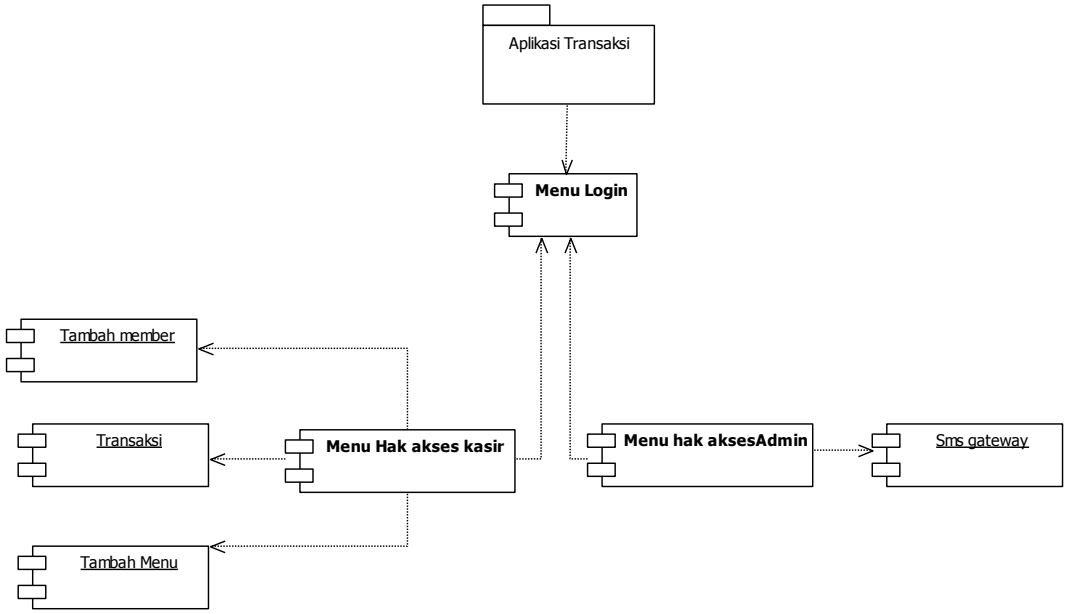
Class diagram adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket paket yang ada didalam sistem/perangkat lunak yang sedang kirakembangkan. Diagram kelas memberi gambaran/diagram statis tenaga sistem/perangkat lunak dan relasi-relasi yang ada didalamnya. Adapun diagram class yang diusulkan adalah sebagai berikut:



Gambar 4.18 class diagram

4.2.3.9. Component Diagram

Component diagram adalah modul fisik program. Komponen – komponen yang dimaksud adalah pustaka – pustaka kode dan berkas – berkas yang ada saat program dijalankan. Adapun Component diagram yang di usulkan adalah sebagai berikut:



Gambar 4.19 Component Diagram

4.2.4. Perancangan Antar Muka

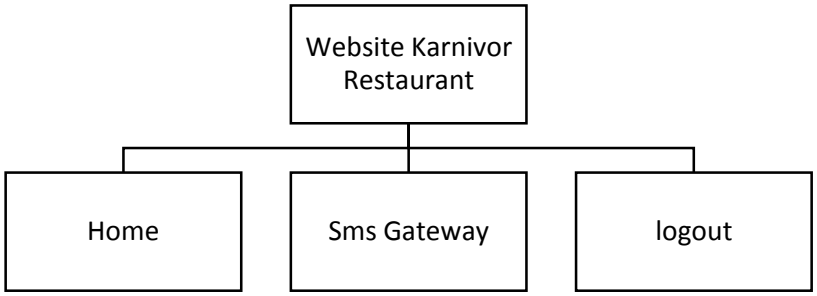
Pada perancangan antarmuka ini akan emnggambarkan rancangan tampilan yang ada dalam sistem yang diusulkan. Adapun yang akan dibahas adalah struktur menu, perancangan input dan output.

1. Struktur Menu

Struktur menu adalah bentuk umum dari satu rancangan program untuk memudahkan pemakai dalam menjalankan program sehingga pada saat menjalankan program, user tidak mengalami kesulitan dalam memilih menu-menu yang diinginkan.

1. Struktur Menu Administrator

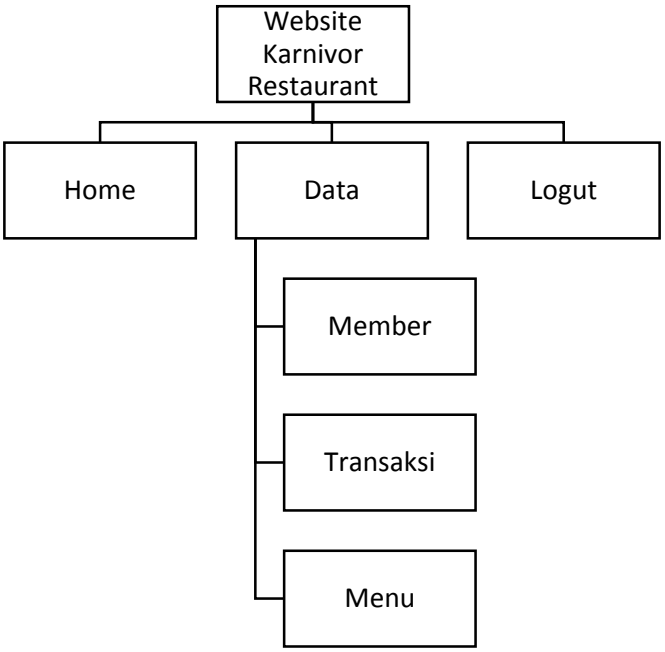
Berikut adalah rancangan menu administrator :



Gambar 4.20 Struktur menu admin yang Diusulkan

2. Struktur Menu Administrator

Berikut adalah rancangan menu kasir :



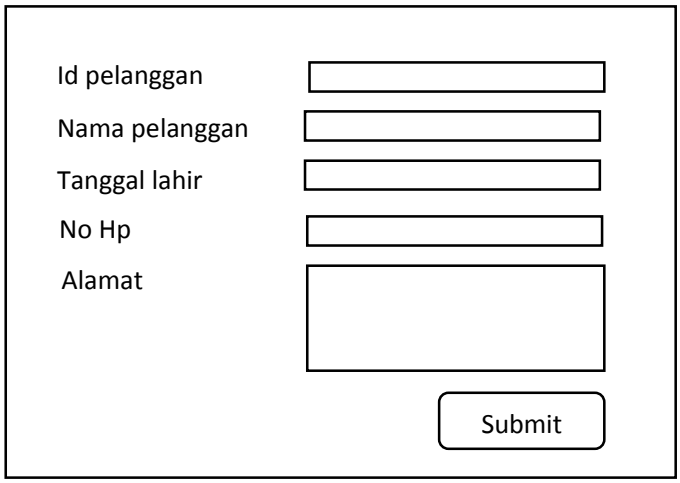
Gambar 4.21 Struktur menu kasir yang Diusulkan

2. Perancangan Input

Rancangan Tampilan ini dipergunakan bagi pengguna yang berkepentingan untuk menggunakan program aplikasi.

1. Input tambah member

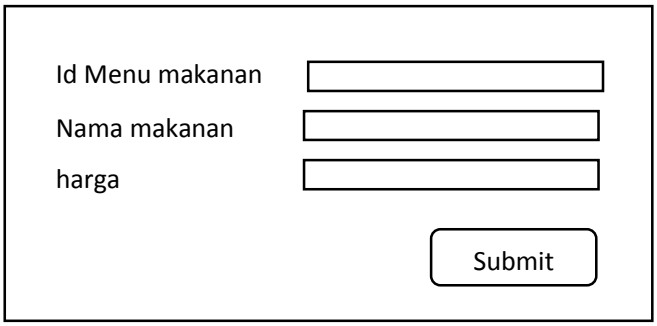
Berikut merupakan perancangan form tambah member.



Gambar 4.22 Rancangan Tampilan tambah member

2. Input Menu makanan

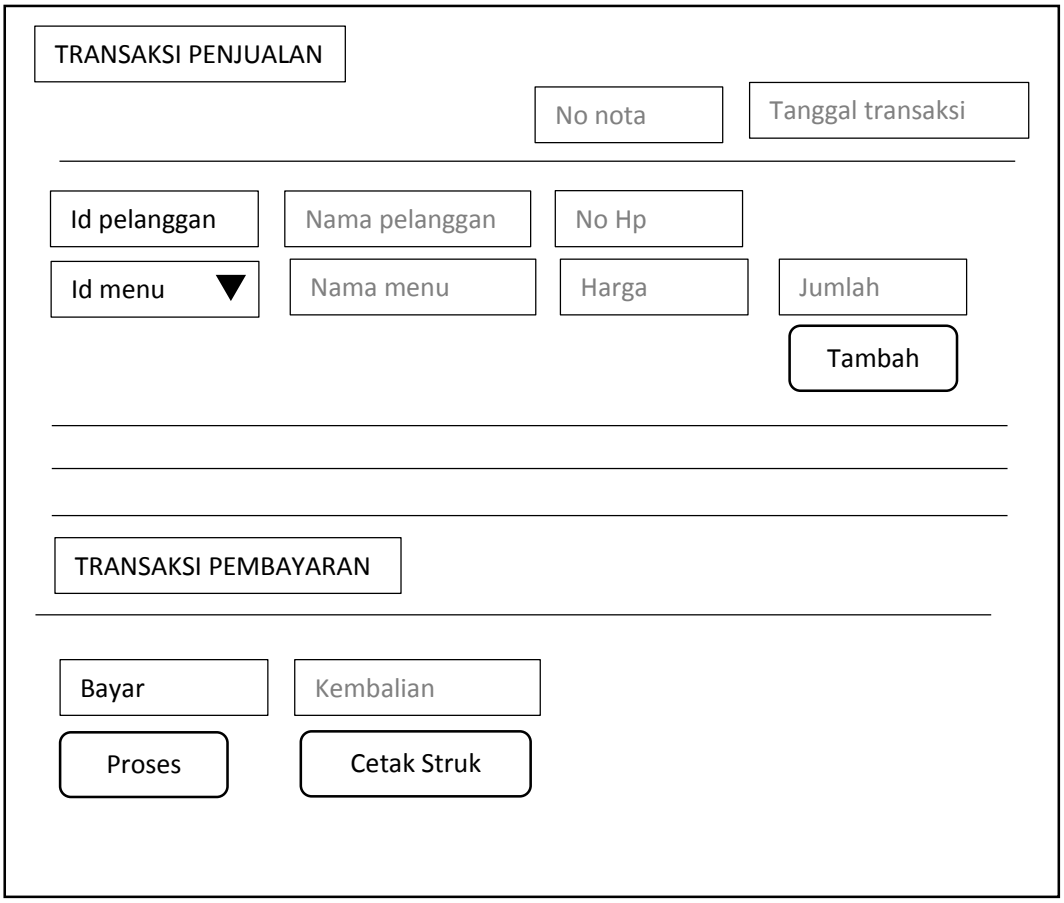
berikut merupkan rancangan Input Menu



Gambar 4.23 Rancangan Tampilan Input Menu makanan

3. Input member

Dibawah ini merupakan rancangan input transaksi oleh kasir:



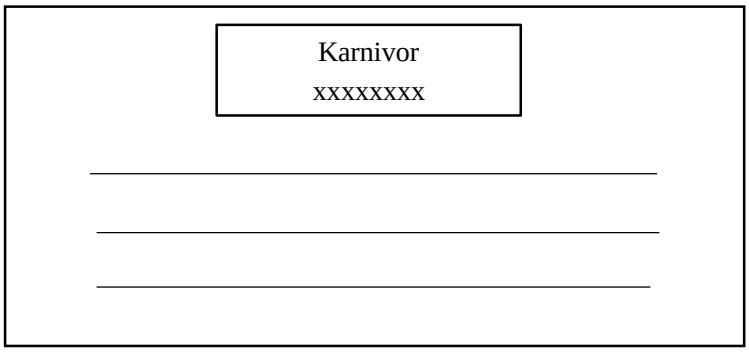
Gambar 4.24 form inputan untuk transaksi

3. Perancangan Output

Rancangan output adalah produk dari sistem informasi yang dihasilkan setelah input dilaksanakan. Hasil dari rancangan input data terlihat dari media keluaran. Dalam rancangan output, sistem informasi ini juga memanfaatkan file database mengandung data-data yang terbilang cukup banyak sehingga membutuhkan tempat yang cukup luas untuk menggambarkan bagaimana sebenarnya rancangan output pada aplikasi yang akan dibuat.

a. Tampilan Struk

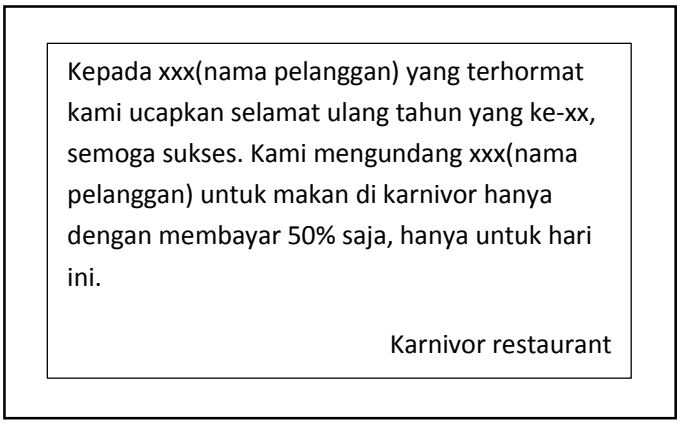
Perancangan *output* ini bertujuan untuk menampilkan hasil transaksi yang dilakuakn pelanggan dengan konsumen.



Gambar 4.25 Rancangan Tampilan Struk

b. Tampilan penanganan pelanggan

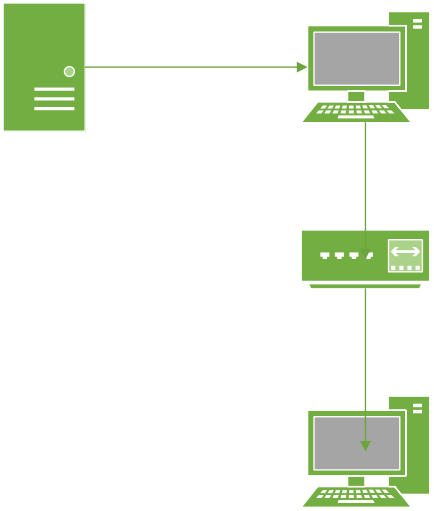
Perancangan *output* ini bertujuan untuk menjaga pelanggan supaya tetap loyal terhadap perusahaan dan outputnya berupa sms yg dikirim melalui *Sms Gateway*.



Gambar 4.26 Rancangan Tampilan *Sms Gateway*

4.2.5. Perancangan Arsitektur Jaringan

Arsitektur jaringan bisa dikatakan sebagai gambaran secara fisik dari pola hubungan antara komponen-komponen jaringan, yang meliputi server, workstation, hub dan pengkabelannnya. Adapun Rancangan Arsitektur jaringan yaitu sebagai berikut :



Gambar 4.27 Arsitektur jaringan

Arsitektur jaringan ini berskala WAN (Wide Area Network). Admin dan server mengirimkan informasi berupa aplikasi web ke internet, dan untuk para user terhubung pada sebuah media penghubung yang dikenal sebagai modem, dimana modem tersebut biasanya berupa lewat saluran telepon yang telah terkoneksi internet