

**SISTEM INFORMASI LAYANAN PEMESANAN PADA RESTORAN
CINDELARAS KOTA MEDAN DENGAN MENGGUNAKAN
FRAMEWORK LARAVEL**

SKRIPSI

GEUBRINA RIZKA UTAMI SINAGA

72154060



**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2020 M/1441 H**

**SISTEM INFORMASI LAYANAN PEMESANAN PADA RESTORAN
CINDELARAS KOTA MEDAN DENGAN MENGGUNAKAN
FRAMEWORK LARAVEL**

SKRIPSI

**Diajukan Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Konsentrasi Sistem Informasi
Fakultas Sains dan Teknologi**

Oleh:

**GEUBRINA RIZKA UTAMI SINAGA
72154060**



**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2020 M / 1441 H**

PERSETUJUAN SKRIPSI

Hal : Surat Persetujuan Skripsi

Lamp : -

Kepada Yth :

Dekan Fakultas Sains dan Teknologi

UIN Sumatera Utara Medan

Assalamu'alaikum Wr, Wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan, maka kami selaku pembimbing berpendapat bahwa skripsi saudara:

Nama : Geubrina Rizka Utami Sinaga

Nomor Induk Mahasiswa : 72154060

Program Studi : Sistem Informasi

Judul : Sistem Informasi Layanan Pemesanan Pada Restoran Cindelaras Kota Medan Dengan Menggunakan Framework Laravel

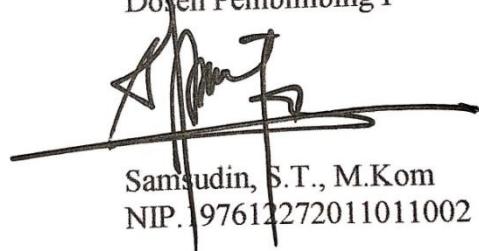
Dengan ini kami menilai skripsi tersebut dapat disetujui untuk dapat segera dimunaqasyahkan. Atas perhatiannya kami ucapkan terima kasih.
Wassalamu'alaikum Wr, Wb.

Medan, 10 Februari 2020

16 Jumadil Akhir 1441 H

Komisi Pembimbing

Dosen Pembimbing I



Samsudin, S.T., M.Kom
NIP. 197612272011011002

Dosen Pembimbing II



Muhammad Dedi Irawan, M.Kom
NIP. 199001312019031019

SURAT PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini:

Nama : Geubrina Rizka Utami Sinaga
NIM : 72154060
Program Studi : Sistem Informasi
Fakultas : Sains Dan Teknologi
Judul : **Sistem Informasi Layanan Pemesanan Pada Restoran Cindelaras Kota Medan Dengan Menggunakan Framework Laravel**

Menyatakan dengan ini sesungguhnya bahwa skripsi yang berjudul di atas adalah asli karya sendiri, kecuali kutipan-kutipan di dalamnya yang disebutkan sumbernya. Apabila dikemudian hari ditemukan plagiat dalam skripsi ini maka saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi lainnya sesuai dengan peraturan yang berlaku.

Demikianlah surat ini saya perbuat dengan sebenar-benarnya untuk dapat dipergunakan sebagaimana mestinya.

Medan, 10 Februari 2020

Penulis



Geubrina Rizka Utami Sinaga
NIM. 72154060



PENGESAHAN SKRIPSI

Nomor: B.103/ST/ST.V.2/PP.01.1/06/2021

Judul : Sistem Informasi Layanan Pemesanan Pada Restoran
Cindelaras Kota Medan Dengan Menggunakan Framework
Laravel

Nama : Geubrina Rizka Utami Sinaga

Nomor Induk Mahasiswa : 72154060

Program Studi : Sistem Informasi

Fakultas : Sains dan Teknologi

Telah dipertahankan di hadapan Dewan Pengaji Skripsi Program Studi Sistem Informasi
Fakultas Sains dan Teknologi UIN Sumatera Utara Medan dan dinyatakan **LULUS**.

Pada hari/tanggal : Senin, 13 Februari 2020

Tempat : Ruang Sidang Fakultas Sains dan Teknologi

Tim Ujian Munaqasyah,

Ketua

Samsudin, ST., M.Kom
NIP. 197612272011011002

Dewan Pengaji,

Pengaji I,

Samsudin, ST., M.Kom
NIP. 197612272011011002

Pengaji III,

M.Irwan Padli Nasution ST.,MM.,M.Kom
NIP. 197502132006041003

Pengaji II,

Triase S.T., M.Kom
NIB. 1100000122

Pengaji IV,

Ali Ikhwan., M.Kom
NIB. 1100000109

Mengesahkan,
Dekan Fakultas Sains dan Teknologi
UIN Sumatera Utara Medan,

Dr. H. M. Jamil, M.A.
NIP. 196609101999031002

ABSTRAK

Restoran merupakan salah satu jenis usaha pangan dilengkapi dengan penyajian dan penjualan makanan dan minuman. Reservasi merupakan aktifitas memesan tempat atau barang, reservasi dan restoran sering dikaitkan karena pemesanan tempat dan menu mempermudah pelanggan restoran sehingga pilihan reservasi menjadi pilihan yang tepat untuk pelanggan yang ingin memesan terlebih dahulu dan akan datang setelahnya. Restoran Ayam Penyet Cindelaras merupakan salah satu restoran yang populer bagi masyarakat jika ingin melakukan reservasi, masyarakat atau pelanggan yang melakukan reservasi pada restoran cindelaras masih menggunakan cara manual, yaitu dengan mendatangi restoran. Hal ini masih belum efektif dan efisien mengingat tidak semua pelanggan yang melakukan reservasi dapat meluangkan waktu untuk sekedar datang melakukan reservasi. Oleh sebab itu, diperlukan sebuah sistem informasi pemesanan yang mampu mewujudkan reservasi yang efektif dan efisien, penerapan sistem ini menggunakan pengembangan *Waterfall* dan menggunakan *tools UML (Unified Modeling Language)* dibangun dengan bahasa pemograman PHP serta MySQL sebagai *database server* dan Laravel sebagai *frameworknya*. Sistem Informasi Layanan Pemesanan Pada Restoran Cindelaras Kota Medan ini menyajikan cara reservasi tempat dan menu yang ada pada restoran dan cara pembayaran melalui *money transfer / m-banking*.

Kata Kunci : Restoran, Reservasi, Cindelaras, Laravel, Waterfall, UML, M-Banking

ABSTRACT

Restaurant is a type of food business equipped with serving and selling food and beverages. Reservation is an activity to order places or goods, reservations and restaurants are often associated because ordering places and menus makes it easier for restaurant customers so that reservation options are the right choice for customers who want to order in advance and will come afterwards. Cindelaras Restaurant is one of the popular restaurants for the public if you want to make a reservation, people or customers who make reservations at Cindelaras Restaurant still use the manual method, namely by visiting the restaurant. This is still not effective and efficient considering that not all customers who make reservations can take the time to just come to make a reservation. Therefore, we need an ordering information system that is able to realize effective and efficient reservations, the application of this system uses Waterfall development and uses UML (Unified Modeling Language) tools built with the PHP programming language and MySQL as the database server and Laravel as the framework. The Information System for Ordering Services at Cindelaras Restaurant, Medan City, presents a way to reserve a place and a menu at a restaurant and a method of payment via money transfer / m-banking.

Keyword : Restaurant, Reservation, Cindelaras, Laravel, Waterfall, UML, M-Banking

KATA PENGANTAR



Assalamu'alaikum Wr. Wb.

Alhamdulillah, puji syukur kehadirat Allah SWT, yang telah melimpahkan rahmat, hidayah serta karunia-Nya, sehingga penulis dapat menyelesaikan Skripsi dengan judul : **“Sistem Informasi Layanan Pemesanan pada Restoran Cindelaras Kota Medan Dengan Menggunakan Framework Laravel”**. Shalawat beriring salam semoga selalu tercurahkan kepada Rasulullah SAW beserta keluarga, para sahabat dan pengikutnya termasuk kita semua yang senantiasa menantikan syafa'atnya di hari akhir kelak.

Skripsi ini disusun sebagai salah satu syarat untuk menyelesaikan program pendidikan Strata-1 Program Studi Sistem Informasi di Universitas Islam Negeri Sumatera Utara Medan. Pada kesempatan ini penulis ingin mengucapkan banyak terima kasih kepada:

1. Bapak Prof. Dr. Saidurrahman, M.Ag selaku rektor UIN Sumatera Utara Medan.
2. Bapak Dr. H.M. Jamil, selaku Dekan Fakultas Sains dan Teknologi UIN Sumatera Utara Medan.
3. Bapak Samsudin, S.T., M.Kom selaku Kaprodi Sistem Informasi UIN Sumatera Utara Medan sekaligus Pembimbing I (satu) yang telah memberikan arahan dalam menentukan judul skripsi, serta membimbing penulis selama menyelesaikan skripsi ini.
4. Ibu Triase, S.T., M.Kom selaku Sekretaris Program Studi Sistem Informasi UIN Sumatera Utara Medan.
5. Bapak Muhammad Dedi Irawan, M.Kom selaku Pembimbing II (dua) yang telah membimbing penulis selama menyelesaikan skripsi ini.

6. Kepada Bapak dan Ibu Dosen Program Studi Sistem Informasi yang telah membimbing penulis selama mengikuti perkuliahan.
7. Kepada pihak Restoran Ayam Penyet Cindelaras khususnya cabang adam malik dan cabang marelant yang telah bersedia memberikan data-data yang dibutuhkan oleh penulis dalam menyelesaikan penelitian sampai dengan selesai.
8. Kepada Orang Tua saya tercinta, Bapak R. Sinaga dan Ibu Suryati serta adik saya Muhammad Fathan Qariby Sinaga yang telah memberikan dukungan dan semangat dalam mengerjakan skripsi ini.
9. Kepada teman-teman Sistem Informasi I (satu) dan II (dua) yang telah memberikan semangat tiada henti kepada penulis dalam menyelesaikan skripsi ini.

Tak lupa pula penulis memohon maaf kepada semua pihak jika dalam penulisan skripsi ini ada bahasa dan tulisan yang tidak sesuai dengan aturan yang telah ditetapkan, semua itu tidak penulis lakukan dengan sengaja, melainkan kesilapan penulis dalam mengerjakan skripsi ini.

Akhir kata penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dan penulis berharap semoga skripsi ini dapat bermanfaat bagi kita semua dan menjadi bahan masukan dalam dunia pendidikan.

Semoga Allah SWT selalu memberikan rahmat dan hidayah-Nya kepada kita semua, *Amin YaaRabbal'Alamin.*

Wassalamu'alaikum Wr. Wb.

Medan, 13 Februari 2020

Penulis,



Geubrina Rizka Utami Sinaga

NIM. 72154060

DAFTAR ISI

HALAMAN JUDUL.....	i
COVER.....	ii
LEMBAR PERSETUJUAN SKRIPSI	iii
SURAT PERNYATAAN KEASLIAN SKRIPSI.....	iv
LEMBAR PENGESAHAN SKRIPSI.....	v
ABSTRAK.....	vi
ABSTRACT	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xv
DAFTAR LAMPIRAN	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	3
BAB II PEMBAHASAN	4
2.1 Sistem.....	4
2.1.1 Karakteristik Sistem.....	4
2.1.2 Klasifikasi Sistem	5
2.1.3 Tujuan Sistem	6
2.2 Informasi	6
2.2.1 Kualitas Informasi.....	6
2.2.2 Nilai Informasi	7

2.3	Sistem Informasi	8
2.3.1	Komponen Sistem Informasi	9
2.4	Pelayanan	10
2.4.1	Kualitas Pelayanan	11
2.5	Pemesanan	12
2.6	Restoran	12
2.6.1	Tentang Restoran Ayam Penyet Cindelaras	12
2.7	Metode Waterfall	13
2.7.1	Fase-Fase Metode Waterfall	13
2.7.2	Karakteristik Metode Waterfall	15
2.8	UML	16
2.8.1	Jenis-jenis Diagram UML dan Beberapa Contoh Diagram	17
2.9	Framework	22
2.10	Laravel	23
2.10.1	Sejarah Laravel	24
2.10.2	Konsep MVC	24
2.10.3	Kelebihan Laravel	24
2.10.4	Fitur Laravel	25
2.11	Database	26
2.11.1	DBMS	27
2.12	PHP	27
2.13	HTML	27
2.14	MySQL	28
2.15	XAMPP	29
2.16	Web Browser	29
2.17	Studi Sejenis	30
	BAB III METODE PENELITIAN	32

3.1	Metode Penelitian	32
3.2	Tempat dan Waktu Penelitian	32
3.3	Cara Kerja	34
	3.3.1 Metode Pengumpulan Data.....	34
	3.3.2 Metode Pengembangan Sistem.....	36
3.4	Kerangka Berfikir Penelitian	39
	BAB IV HASIL DAN PEMBAHASAN.....	40
4.1	Struktur Organisasi	40
	4.1.1 Visi dan Misi.....	41
4.2	Analisis Yang Diperlukan.....	41
4.3	Perancangan Sistem	42
	4.3.1 Use Case Diagram.....	43
	4.3.2 Activity Diagram	51
	4.3.3 Sequences Diagram.....	55
	4.3.4 Class Diagram.....	57
4.6	Desain Database.....	58
4.7	Desain Interface	64
4.8	Implementasi dan Pengkodean	68
4.9	Testing.....	69
	4.9.1 Interface	69
	4.9.2 Blackbox Testing	79
	BAB V KESIMPULAN DAN SARAN	81
5.1	Kesimpulan	81
5.2	Saran	81
	DAFTAR PUSTAKA	82
	LAMPIRAN I	85
	LAMPIRAN II	114
	LAMPIRAN III	116

DAFTAR GAMBAR

Gambar	Nama Gambar	Hal
2.1	Fase-Fase Metode Waterfall	15
2.2	Contoh Use Case Diagram.....	18
2.3	Contoh Activity Diagram.....	19
2.4	Contoh Sequences Diagram	21
2.5	Contoh Class Diagram	22
3.1	Kerangka Berpikir Penelitian.....	39
4.1	Struktur Organisasi Restoran APCL Cabang Adam Malik....	40
4.2	Struktur Organisasi Restoran APCL Cabang Marelan.....	40
4.3	Use Case Diagram Sistem Reservasi Cindelaras	51
4.4	Activity Diagram Login Admin	52
4.5	Activity Diagram Menu	53
4.6	Activity Diagram Konfirmasi Reservasi Admin.....	54
4.7	Activity Diagram User	55
4.8	Sequences Diagram Reservasi Admin	56
4.9	Sequences Diagram Reservasi User.....	57
4.10	Class Diagram	58
4.11	Form Home Admin	64
4.12	Form Login Admin	65
4.13	Form Home	66
4.14	Form Reservasi	67
4.15	Interface Menu Home	69
4.16	Interface Menu	70
4.17	Interface Reservasi	71
4.18	Interface Daftar Menu	72

4.19	Interface Detail Reservasi	73
4.20	Interface Konfirmasi Pembayaran.....	74
4.21	Interface Selesai Reservasi.....	75
4.22	Interface Login Admin.....	75
4.23	Interface Dashboard	76
4.24	Interface Data Meja.....	76
4.25	Interface Data Reservasi	77
4.26	Interface Konfirmasi DP	77
4.27	Interface Daftar Menu Admin	78
4.28	Interface Tambah Daftar Menu.....	78
4.29	Interface Laporan	79

DAFTAR TABEL

Tabel	Nama Tabel	Hal
2.1	Simbol Use Case Diagram.....	17
2.2	Simbol Activity Diagram	18
2.3	Simbol Sequences Diagram.....	20
2.4	Simbol Class Diagram.....	21
2.5	Referensi Studi Sejenis.....	30
3.1	Waktu Penelitian & Pengerjaan Sistem.....	33
3.2	Analisis Dokumen	35
4.1	Tabel User	58
4.2	Tabel Reservasi Menu	59
4.3	Tabel Reservasi	60
4.4	Tabel Menu.....	61
4.5	Tabel Meja.....	61
4.6	Tabel Konfirmasi Bayar	62
4.7	Tabel Admin.....	62
4.8	Tabel Cabang.....	63
4.9	Tabel Kategori Menu.....	64
4.10	Tabel Blackbox Testing.....	79

DAFTAR LAMPIRAN

Lampiran I Hasil Pertanyaan Wawancara

Lampiran II Source Code

Lampiran III Laporan Data Penelitian Cindelaras

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dewasa ini teknologi telah berkembang dan menghasilkan teknologi informasi berbasis internet, dalam konsep ini dunia bisnis juga telah merambah ke dalam dunia teknologi di berbagai bidang termasuk pada bidang kuliner.

Restoran atau rumah makan ayam penyet cindelaras merupakan salah satu bisnis restoran yang cukup diminati oleh masyarakat kota medan. Namun saat ini dicindelaras masih melakukan pemesanan tempat (*booking*) dan pembayaran dengan sistem manual sehingga pelanggan masih harus mendatangi lokasi rumah makan atau menghubungi melalui nomor telepon restoran untuk melakukan pemesan tempat (*booking*) dan juga masih melakukan pembayaran secara tunai. Sistem manual ini adakalanya dapat menimbulkan beberapa masalah. Masalah yang biasa terjadi ialah meja dan kursi tidak sesuai pesanan, kesalahan menu makanan dan minuman, kesalahpahaman pelanggan dan lain-lain, sistem manual sangat tidak efisien dan praktis apalagi pada era teknologi sekarang yang mengedepankan kemajuan teknologi banyak rumah makan yang telah menyediakan layanan *take-outdining* atau *deliveryservice*.

Pada kajian permasalahan di atas peneliti akan merancang sebuah sistem dimana pelanggan tidak perlu mendatangi rumah makan atau restoran tersebut dan cukup dengan menggunakan *smartphone* maka aplikasi pemesanan tempat dan makanan bisa digunakan. Untuk pembayaran uang muka/panjar peneliti akan menggunakan cara non-tunai yaitu dengan menggunakan nomor rekening atau *m-banking* yang dimana pelanggan bisa menyelesaikan pembayarannya di *ATM* terdekat atau dengan *mobilebanking*. Inilah yang menjadi pokok utama dalam melakukan penyusunan skripsi yang berjudul **“Sistem Informasi Layanan Pemesanan Pada Restoran Cindelaras Kota Medan Dengan Menggunakan Framework Laravel”**.

1.2 Rumusan Masalah

Adapun rumusan masalah pada penelitian ini ialah:

1. Bagaimana merancang serta membangun sistem informasi layanan pemesanan pada restoran cindelaras berbasis web ?
2. Bagaimana penerapan pembayaran rekening/*m-banking* pada sistem informasi layanan pemesanan pada restoran cindelaras ?

1.3 Batasan Masalah

Menurut latar belakang diatas, ruang lingkup penelitian dapat dibatasi menjadi:

1. *Website* ini akan dibangun menggunakan bahasa pemrograman PHP 7, HTML 5, dan *Database MySQL*.
2. *Website* ini dibangun dengan menggunakan *framework* PHP laravel dan *booster* sebagai *framework* HTML-nya.
3. Pembuatan *website* ini akan difokuskan pada 2 cabang restoran ayam penyet cindelaras yaitu pada cabang marelan dan cabang adam malik.
4. Pembuatan *website* ini akan difokuskan pada pemesanan (*booking*) tempat dan menu di restoran cindelaras.
5. Batas waktu pembayaran *bookingfee* ialah 1 jam setelah dilakukannya pemesanan.
6. Promo/diskon berlaku kepada pelanggan yang telah setia, dan ditentukan oleh admin/pihak restoran.

1.4 Tujuan Penelitian

Pada penelitian ini memiliki tujuan sebagai berikut:

1. Menghasilkan sistem aplikasi pelayanan restoran dalam pemesanan tempat dan makanan di restoran ayam penyet cindelaras.
2. Menerapkan pembayaran rekening pada sistem transaksi pemesanan tempat dan menu di restoran cindelaras.

1.5 Manfaat Penelitian

Manfaat yang akan dicapai pada penelitian ini:

1. Manfaat kepada pelaku peneliti ialah:
 - a. Membuat peneliti paham bagaimana pengembangan system informasi berbasis *website*.
 - b. Untuk syarat kelulusan untuk mendapatkan gelar sarjana (S1) di Program Studi Sistem Informasi Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara Medan.
2. Manfaat kepada institusi ialah:
 - a. Penelitian ini bertujuan sebagai bahan referensi penelitian mahasiswa lain.
 - b. Mengetahui kompetensi mahasiswa dalam menguasai teori yang pelajari selama berkuliah.
3. Manfaat bagi restoran tersebut ialah:
 - a. Membantu dalam kegiatan sistem informasi (*website*) pada restoran ayam penyet cindelaras.
 - b. Mendapatkan sistem yang dapat digunakan untuk mendukung proses layanan orderan pada Restoran Ayam Penyet Cindelaras.

BAB II

TINJAUAN PUSTAKA

2.1 Sistem

Sistem memiliki makna sebagai himpunan dari komponen berupa data, dan cara yang saling terhubung, baik *SDM*, *hardware* dan *software* saling berinteraksi untuk mencapai suatu tujuan (Maniah, 2017).

(Hutahaean, 2014) menurut pendapat dari beberapa para ahli bahwa sistem ialah :

1. Menurut Murdick, R.G., sistem adalah sekumpulan komponen yang mempola himpunan atau himpunan *procedure* pengolahan yang mencari arah tertentu.
2. Menurut Davis, G.B., sistem secara fisik adalah himpunan dari komponen yang bekerja secara bersama untuk mencapai suatu tujuan.

2.1.1 Karateristik Sistem

Sistem memiliki beberapa ciri atau sifat-sifat tertentu, antara lain (Kusrini&Koniyo, 2007):

1. Component

Sistem dibangun dari beberapa elemen yang saling bersinggungan yang akan bekerjasama mempola bagian-bagian sistem.

2. Boundary

Area yang akan saling membatasi sebuah sistem kepada ekosistem kerjanya.

3. Subsistem

Irisan elemen sistem yang saling aktif dan berhubungan untuk mencapai tujuan masing-masing.

4. Environment

Luaran sekat sistem yang dipengaruhi oleh operasi sistem.

5. Penghubung Sistem (*Interface*)

Media yang menghubungkan subsistem kepada subsistem yang lain.

Keterkaitan ini akan meniscayakan sumber daya berjalan dari satu subsistem ke subsistem yang lain.

6. **Masukkan Sistem (*Input*)**

Input berupa energi yang masuk kedalam sistem, energi tersebut dimasukkan agar dapat berhubungan.

7. **Keluaran Sistem (*Output*)**

Keluaran yang dihasilkan dari energi diproses dan dibagi-bagi menjadi *output* yang bermanfaat dan sisa pembuangan.

8. **Pengolahan Sistem (*Process*)**

Sistem dapat memproses dan mengalihkan *input* menjadi *output*.

9. **Sasaran Sistem (*Object*)**

Sasaran yang akan dicapai oleh sistem akan dikategorikan sukses apabila telah mencapai sasaran atau sasaran tersebut.

2.1.2 Klasifikasi Sistem

Klasifikasi sistem adalah sebagai berikut:

1. **Sistem Abstrak dan Sistem Fisik**

Merupakan sistem yang memiliki pemikiran nonfisik, sedangkan sistem fisik merupakan sistem yang memiliki atau ada secara fisik.

2. **Sistem Alamiah dan Sistem Buatan Manusia**

Merupakan sistem terbentuk melalui teknik alam, sedangkan sistem buatan merupakan sistem yang dirancang oleh manusia.

3. **Sistem Tertentu dan Sistem Tak Tertentu**

Sistem yang dapat diprediksi secara tepat dan sistem yang tidak dapat diprediksi secara tidak tepat.

4. **Sistem Tertutup dan Sistem Terbuka**

Sistem yang dipengaruhi oleh sistem luar dan sistem yang berhubungan dengan ekosistem luar (Kusrini&Koniyo, 2007).

2.1.3 Tujuan Sistem

Susanto (2002), “Tujuan sistem adalah sasaran atau tujuan akhir yang ingin dicapai oleh suatu sistem”. Supaya tujuan ini berhasil, kriteria terlebih dahulu yang perlu diketahui. Kriteria juga dapat dijadikan tolak ukur untuk menilai ketercapaian sebuah sistem dan menjadi rujukan untuk melakukan suatu pengendalian. Jadi, kriteria tujuan bersifat absolut (Djahir&Pratita, 2014).

Berdasarkan uraian di atas, bisa disimpulkan tentang tujuan sistem adalah komponen terakhir dalam elemen-elemen yang saling berhubungan didalam suatusistem. Tanpa tujuan sistem, maka sistem menjadi tidak sistematis dan tidak terkendali. Perlu diketahui juga bahwa tujuan sebuah sistem antara sistem yang lain berbeda.

2.2 Informasi

Banyak definisi informasi yang kita ketahui dan juga dikemukakan oleh para ahli tentang apa itu informasi, sebenarnya apa itu informasi? Berikut pendapat salah satu ahli:

Edhy Sutanta (2011), “Informasi merupakan hasil pengolahan data, sehingga menjadi bentuk yang penting bagi penerimanya dan memiliki kegunaan sebagai dasar pengambilan keputusan yang dapat dirasakan secara langsung pada saat itu maupun tidak langsung dimasa yang akan datang” (Muslihudin& Oktafianto, 2016). Selain itu, pendapat lain menjelaskan bahwa informasi adalah keterangan yang sudah diproses menjadi wujud yang bermakna kepada pemakainya, yang berguna dalam penentuan keputusan terkini atau lumbung informasi pendukung. Data tidak mempunyai *value*, sedangkan informasi pasti mempunyai *value*. Informasi akan bernilai jika manfaatnya lebih besar daripada biaya produksinya (Kusrini & Koniyo, 2007).

2.2.1 Kualitas Informasi

Sutarbi (2005) kualitas dari sebuah informasi terkait dengan tiga (3) aspek, yaitu:

1. *Accurate*

Informasi wajib terlepas dari kekeliruan yang tidak dapat diterima atau menyimpang. Akurat dimaksudkan yaitu informasi wajib secara benar menginterpretasikan maksudnya.

2. *Timelines*

Informasi harus segera diterima oleh penerima, informasi yang telah lusuh tidak akan mempunyai *value*. Karena informasi adalah hal yang mendasar pengambilan kebijakan.

3. *Relevance*

Informasi ini memiliki kebaikan bagi pemakainya. Relevansi pada setiap individu itu berbeda (Muslihudin & Oktafianto, 2016).

2.2.2 Nilai Informasi

Nilai dari informasi dibagi menjadi 2 bagian, yaitu manfaat yang didapat dan biaya untuk produksinya. Informasi dapat diklasifikasikan bernilai jika manfaat yang didapatkan lebih dari *cost* yang dikeluarkan. Nilai informasi tidak bisa dikalkulasikan dengan nilai uang, tetapi nilai keefektifannya bisa diperkirakan. Nilai informasi dikaitkan dengan telaah *costeffectiveness* atau *costbenefits* (Sutabri, 2004). Terdapat sepuluh karakteristik nilai informasi, yaitu:

1. Mudah didapat

Sifat ini menunjukkan kemudahan dan kecepatan mendapatkan informasi. Kecepatannya bisa diukur, misalnya 1 menit versus 24 jam. Namun, seberapa berharga informasi bagi pengguna sulit untuk diukur.

2. Luas dan lengkap

Sifat ini menunjukkan kelengkapan isi informasi. Ini bukan hanya tentang volume, tetapi juga tentang keluaran informasi. Sifat ini sangat kabur dan oleh karena itu sulit diukur.

3. Ketelitiannya

Ciri ini terkait dengan derajat kebebasan dari kekeliruan dalam *output* informasi. terdapat dua jenis kegalatan dalam volumen data yang besar yaitu galat dalam perhitungan dan galat dalam pencatatan.

4. Kecocokan

Sifat ini menampilkan seberapa bagus *output* informasi terkait dengan *request* pengguna. Isi dari informasi wajib berhubungan dengan masalah yang dihadapi sementara semua keluaran lainnya tidak bermanfaat. Sifat ini sulit terukur.

5. Ketepatan waktu

Merupakan shift yang berkaitan dengan waktu yang dilalui.

6. Kejelasan

Merupakan tingkatan yang menunjukkan kejelasan suatu informasi yang bebas dari kata yang rancu.

7. Keluwesan

Sifat tidak memiliki keinginan untuk mengubah informasi yang berguna untuk mencapai kesimpulan.

8. Bisa dibuktikan

ciri ini menampilkan sejauh apa informasi dapat telaah oleh beberapa *user* untuk sampai pada kesimpulan yang sama.

9. Tidak ada prasangka

Merupakan ciri yang berkaitan dengan ada atau tidaknya hasrat untuk merubah informasi yang diarahkan sebelumnya.

10. Bisa diukur

Ciri ini adalah inti dari informasi yang produksi oleh sistem informasi formal.

2.3 Sistem Informasi

Sistem informasi adalah himpunan tahapan organisasi yang diterapkan dalam bentuk terkomputerisasi dan otomatis untuk mencapai hasil, yaitu memberikan informasi kepada pengambil kebijakan dan mengendalikan sistem. (Sulistyo, 2017). Selain itu, sistem informasi yaitu himpunan sub sistem fisik maupun nonfisik yang saling berinteraksi satu dengan yang lain dan bergiatan secara bersama dengan serasi untuk menggapai hasilnya yaitu mengolah data sehingga menjadi informasi yang berfaedah (Azhar, 2017).

Oleh karena itu, berdasarkan kajian diatas maka dapat dipahami bahwa sistem informasi adalah gabungan interaksi antara data, teknologi serta aktivitas masyarakat yang memanfaatkan teknologi tersebut.

2.3.1 Komponen Sistem Informasi

Untuk menyusun sistem informasi diperlukannya elemen sistem informasi, dalam sistem informasi terdapat beberapa komponen (Anggraeni Yunaeti&Irviani, 2017) yaitu:

1. *Hardware*

Termasuk perangkat fisik seperti Pc dan alat cetak.

2. *Software*

Serangkaian perintah yang diberikan ke *Hardware* untuk dapat mengolah data.

3. *Procedure*

Seperangkat tahapan yang digunakan untuk mencapai pengolahan data dan menghasilkan *output* yang diinginkan.

4. *Programmer*

Orang-orang yang bertanggung jawab atas pengembangan dari sistem informasi, pengolahan, serta penggunaan *input* sistem informasi.

5. *Database*

Seperangkat hubungan, tabel, serta hal lain yang terkait dengan penyimpanan dari data.

6. Komunikasi data dan Jaringan komputer

Sistem perantara yang meniscayakan sumber daya untuk dibagikan atau dimanfaatkan oleh pengguna.

Sementara itu, Burch dan Grudnitski (1986) mengemukakan bahwa suatu sistem informasi yang tersusun dari elemen yang disebut *buildingblock*, *modelblock*, *outputblock*), *technologyblock*, dan *controlblock*. Sebagai sebuah sistem, masing-masing dari enam blok di atas saling berhubungan membentuk satu interaksi untuk mencapai hasilnya (Al Fatta, 2007) keenam blok yang terdapat dalam sistem informasi, antara lain:

1. Blok Masukan

Input hanya dimasukkan di sistem informasi. *Block Input* meliputi metode pengambilan data yang dapat diinputkan dalam bentuk dokumen dasar.

2. Blok Model

Blok ini hanya tersusun dari beberapa kombinasi proses, model matematis, dan logika yang akan menampilkan ilusi data masukan dan data yang disimpan untuk menghasilkan *input* yang dikehendaki.

3. Blok Keluaran

Informasi berkualitas seperti dokumentasi yang bermanfaat untuk setiap tingkatan manajemen.

4. Blok Teknologi

Teknologi adalah kotak alat didalam sistem informasi. Teknologi dimanfaatkan untuk menyerap masukan, menyimpan data, menjalankan model, dan mengakses data, memproduksi dan mengirimkan *output* dan membantu mengontrol sistem keseluruhan.

5. Blok Basis Data

Data yang terhubung dan disimpan di perangkat keras PC untuk memanipulasinya membutuhkan *software*.

6. Blok Kendali

Pengendalian perlu direncanakan dan diimplementasikan agar memastikan bahwa hal yang bias menghancurkan sistem dapat diantisipasi atau jika terjadi kegalatan bisa segera diatasi dengan cepat.

2.4 Pelayanan

Armstrong (2008) menyatakan bahwa pelayanan ialah setiap giat atau kebaikan yang bisa dilakukan oleh masing-masing pihak yang pada konsepnya tidak berwujud dan tidak berdampak pada pemilihan suatu hal. Selain dari itu, produksi dapat atau tidaknya dapat di relevansikan dengan produk fisik (Rangkuti, 2017). Berdasarkan kajian di atas, dapat simpulkan bahwa pelayanan merupakan suatu bentuk kegiatan yang menguntungkan dari satu pihak kepihak lain.

Adapun beberapa pendapat dari para ahli, sebagai berikut (Rangkuti, 2017):

1. Menurut Atep Adya Brata, “Pelayanan adalah segala upaya untuk memberikan kemudahan guna mewujudkan kepuasan calon pembeli atau pelanggan sebelum atau sesudah terjadi transaksi”.
2. Menurut A.S.Moenir, “Pelayanan adalah suatu proses pemenuhan kebutuhan melalui kegiatan orang lain yang langsung diterima. Dengan kata lain dapat dikatakan bahwa pelayanan adalah suatu tindakan yang dilakukan oleh orang lain sehingga masing-masing mendapat yang diharapkan. manfaat dan kepuasan.”

Berdasarkan pengertian diatas dapat kita pahami bahwa pelayanan merupakan kegiatan atau kegiatan yang saling menguntungkan yang dilakukan oleh satu sisi kepada sisi lainnya sehingga dapat menghasilkan kepuasan bagi kedua belah sisi.

2.4.1 Kualitas Pelayanan

Pengukuran dari kinerja layanan bisa dilakukan dengan memanfaatkan alat ukur kinerja layanan yang dirancang oleh Parasuraman, Zeithaml, dan Berry dalam bukunya yang berjudul Delivery Quality Service. Menurut Parasuraman, Zeithaml, dan Berry (1999) terdapat sebelas aspek kualitas pelayanan yaitu (Nogi, 2005):

1. *Tangible* (Kenampakan fisik)
2. *Reliability* (Reliabilitas)
3. *Responsiveness* (Responsivitas)
4. *Competence* (Kompoetensi)
5. *Courtesy* (Kesopanan)
6. *Credibility* (Kredibilitas)
7. *Security* (Keamanan)
8. *Access* (Akses)
9. *Communication* (Komunikasi)
10. *Understanding the customer* (Pengertian Pelanggan)

11. *Accountability* (Akuntabilitas)

2.5 Pemesanan

Pemesanan merupakan kegiatan yang dilakukan pelanggan sebelum membeli. Demi mencapai kepuasan pelanggan, perusahaan wajib memiliki sistem *order* yang baik. Menurut KBBI di situs resmi Kementerian Pendidikan dan Kebudayaan (Kemendikbud, 2019) yang diakses pada 21 September 2019, pemesanan adalah pembuatan, proses, cara pemesanan (pembuatan, tempat, pembuatan, dan lain-lain) kepada pemesan.

Berdasarkan pengertian diatas, kita ketahui bahwa pemesanan adalah aktifitas yang dilakukan untuk pesan tempat, item dan lain-lain terhadap orang lain.

2.6 Restoran

Menurut Direktorat Jenderal. Par. 1990/1991:2, restoran adalah salah satu jenis usaha makanan yang sebagian atau seluruhnya berada pada suatu bangunan permanen, dilengkapi dengan peralatan untuk menyimpan, menyajikan, dan menjual makanan dan minuman kepada masyarakat di tempat usahanya dan memenuhi kebutuhan persyaratan yang ditetapkan dalam keputusan tersebut (Wiwoho, 2008). Singkatnya, restoran adalah usaha makanan yang melayani berbagai kebutuhan seputar makanan.

2.6.1 Tentang Restoran Ayam Penyet Cindelaras

Dalam situs resmi *e-book* Amerika Scribd, Inc yang diunggah pada 5 April 2015 dan diakses pada 27 September 2019 (Akbar, 2015), Usaha Ayam Cindelaras Penyet pada awalnya bermula dengan usaha bakso keliling yang telah dirintis oleh Pak Sumarsono sejak tahun 1994. Seirama berjalannya waktu, ketika Pak Sumarsono berjibaku mengembangkan mimpiinya, ia mengigikan kelak bisnis bakso kelilingnya bisa berkembang menjadi restoran skala nasional yang besar. Berkat kerja keras bersama rekan, gerai pertama akhirnya terwujud, ditandai dengan diresmikannya “Bakso Urat ADS” yang berlokasi di Jalan Setia Budi Medan. Sejak momen itu usaha Pak Sumarsono terus berkembang sampai

akhirnya di tahun 2007 beliau meresmikan “Ayam Penyet Cindelaras” yang sampai sekarang telah berkembang seperti yang kita kenal sekarang.

Berikut cabang/outlet Restoran Ayam Penyet Cindelaras yang telah tersebar di wilayah Kota Medan dan sekitarnya:

1. APCL Teladan, Jln. Sisingamangaraja No. 123, Teladan Barat, Medan
2. APCL Sei Belutu, Jln. Sei Belutu Medan (Samping Kampus UMA Setia Budi, Medan)
3. APCL Adam Malik, Jln. AdamMalik Medan
4. APCL Ring Road, Jln. Gagak Hitam Medan (Samping SPBU Pertamina)
5. APCL Lauchi, Jln. Jamin Ginting KM. 13,5 Lauchi, Medan
6. APCL Marelan, Jln. Marelan Raya No. 55 (Di Sebelah Alfamart Marelan 2), Tanah Enam Ratus, Medan
7. APCL Tandem, Jln. Teuku Amir HamzahNo. 700, Tandem, Binjai
8. APCL Sei Rampah, Jln. Medan – Tebing Tinggi
9. APCL Binjai, Jln. Jenderal Gatot Subroto, Suka Ramai, Kota Binjai
10. APCL Kualanamu, Jln. Batang Kuis, Bangun Sari, Tanjung Morawa
11. APCL Stabat, Jln. Proklamasi No. 9, Kwala Bingai, Stabat
12. APCL Tebing Tinggi, Jln. Gn. Leuseur, Tj. Marulak, Kota Tebing Tinggi
13. APCL Simalungun, Sei Mangkei, Bosar Maligas, Kabupaten Simalungun, Sumatera Utara

2.7 Metode *Waterfall*

Menurut (Yurindra, 2017) *Waterfall* merupakan sebuah metode dalam SDLC yang memiliki ciri spesifik pekerjaan yaitu masing-masing tahapan pada *waterfall* wajib diselesaikan dahulu sebelum meneruskan ke tahapan berikutnya. Maksudnya fokus pada setiap bagian dapat terlaksana secara optimal karena jarang ada pekerjaan paralel padahal bisa terjadi paralelisme di *waterfall*.

2.7.1 Fase-Fase Metode *Waterfall*

Tahapan metode *waterfall* adalah sebagai berikut (Yurindra, 2017):

1. *Requirement Analysis*

Fase ini membutuhkan *software*, informasi tersebut umumnya dapat didapatkan melalui survey, dan wawancara. Informasi akan dianalisis untuk memperoleh informasi dan didokumentasikan untuk proses selanjutnya.

2. *System Design*

Bagian ini dilakukan sebelum *coding*. Bagian ini berguna agar memberikan gambaran yang harus dilakukan dan tentang tampilannya. Tahap ini menolong dalam menentukan persyaratan *Hardware* dan sistem serta menentukan arsitektur sistem keseluruhan.

3. *Implementation*

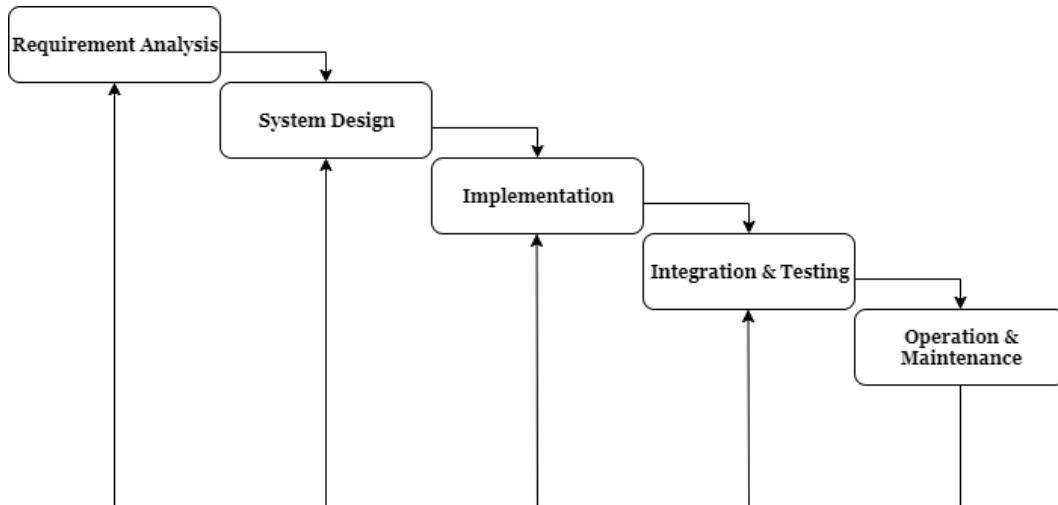
Pada bagian ini dilakukan perancangan program. Pengembangan perangkat lunak dibagi menjadi modul kecil yang akan dikumpulkan pada bagian selanjutnya. Selain itu, pada tahap ini juga dilakukan pengujian terhadap modul yang sedang dibuat, apakah telah memenuhi fungsi yang harapkan atau belum.

4. *Integration and Testing*

Pada bagian ini modul yang telah dibangun digabung dan dilakukan pengujian agar dapat mengetahui *software* yang dibangun sudah sesuai dengan desain dan memiliki celah atau tidak.

5. *Operation and Maintenance*

Bagian ini merupakan Bagian akhir akhir dari model *waterfall*. *Software* yang telah selesai dioperasikan dan dilakukan pemeliharaan. Pemeliharaan yang dilakukan yaitu memperbaiki kegalatan yang tidak dapatkan pada tahap pengujian. Meningkatkan penerapan unit sistem dan mengoptimalkan layanan sistem untuk persyaratan baru.



Gambar 2.1 Fase-fase Metode *Waterfall* (Yurindra, 2017)

2.7.2 Karakteristik Model *Waterfall*

1. Persyaratan-persyaratan teori model *waterfall* dapat dilihat dibawah ini, (Yurindra, 2017):
 - a. Ketika semua persyaratan dipahami dengan baikdi awal pengembangan.
 - b. Definisi produk stabil dan tidak berubah selama pengembangan karena kilah apa pun seperti perubahan eksternal, perubahan tujuan, perubahan anggaran, atau perubahan teknologi. Untuk itu, teknologi yang dipakai harus dipahami dengan maksimal.
 - c. Menghasilkan produk baru, atau versi terbaru dari produk yang sudah ada. Pada dasarnya, jika anda menghasilkan versi terbaru akan masuk *incremental development*, dimana tiap stage sama seperti *Waterfall* kemudian direpetitif.
 - d. Posting produk yang sudah ada ke *platform*.
2. Kelebihan Metode *Waterfall*
 - a. Proses menjadi lebih sistematis, urutan pemrosesan dengan metode ini akan lebih sistematis dari satu tahap ketahap berikutnya.

- b. Dari sisi pengguna juga lebih menguntungkan karena bisa merancang dan menyiapkan semua data dan kebutuhan pengolahan yang akan dibutuhkan.
 - c. Jadwal menjadi pasti, jadwal setiap proses bisa dipastikan dengan pasti. Sehingga bisa terlihat dengan jelas target penyelesaian pengembangan program. Dengan urutan yang pasti, seseorang juga dapat melihat progres yang benar untuk setiap tahapannya.
3. Kelemahan *Waterfall*
- a. Sifatnya preskriptif, sehingga sukar untuk melakukan perubahan ditengah proses.
 - b. Jika ada kekurangan proses dari tahap sebelumnya, maka tahap pengembangan harus dilakukan dari awal. Ini akan memakan banyak waktu. Karena jika proses sebelumnya belum selesai hingga selesai, maka tahapan selanjutnya tidak akan berjalan. Jadi, jika ada kekurangan dalam permintaan pengguna, proses *development* harus dimulai dari awal.
 - c. Membutuhkan urutan kebutuhan yang lengkap diawal, tetapi jarang sekali pelanggan dapat menyediakan kebutuhan yang lengkap sebelumnya.

2.8 UML

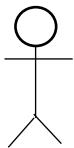
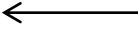
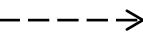
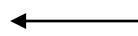
UML Kepanjangannya “*Unified Modeling Language*” ialah metode yang dilakukan secara visual untuk merancang sistem berorientasi objek (Hai & Setiawan, 2019). Pengertian lain dari UML ialah bahasa pemodelan yang dapat menjelaskan secara detail dalam menganalisa dan mengeksekusi sesuatu yang diperlukan sistem (Suendri, 2018). Sistem *blueprint* ialah sebuah sistem standar penulisan yang disuguhkan UML yaitu meliputi konsep bisnis proses, penulis kelas-kelas kedalam bahasa program yang individual, skema database, dan elemen yang dibutuhkan pada sistem *software* (Zufria, 2013).

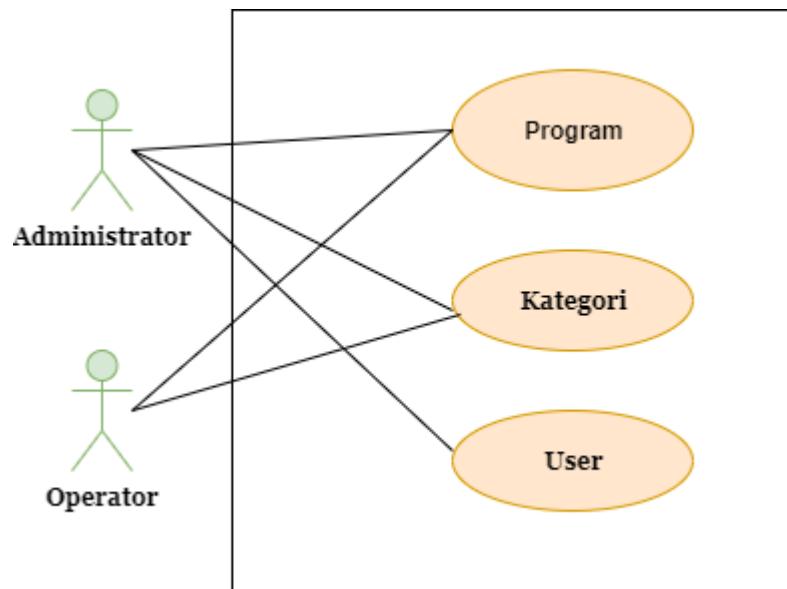
2.8.1 Jenis-jenis Diagram UML dan Beberapa Contoh Diagram

1. Use Case Diagram

Use case diagram biasa menjelaskan tipe interaksi antara si pemakai sistem dengan sistemnya. (Hai & Setiawan, 2019).

Table 2.1 Simbol-simbol Use Case Diagram

No.	Gambar	Nama	Keterangan
1		<i>Actor</i>	Mengkhususkan himpunan peran yang dimainkan user ketika berhubungan dengan <i>use case</i>
2		<i>Dependency</i>	Hubungan yang jika perubahan suatu elemen independen terjadi maka akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri
3		<i>Generalization</i>	Hubungan yang mana objek anak berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk
4		<i>Include</i>	Mengkhususkan bahwa <i>use case</i> sumber secara eksplisit
5		<i>Extend</i>	Mengkhususkan bahwa <i>use case</i> tujuan memperluas perilaku dari <i>use case</i> dasar ke suatu titik yang diberikan
6		<i>Association</i>	Yang mengkoneksikan objek satu dengan objek lainnya
7		<i>Use case</i>	Keterangan dari susunan aksi-aksi yang ditampilkan sistem dan menciptakan suatu hasil yang terukur bagi <i>actor</i>
8		<i>Collaboration</i>	Hubungan aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan komponen (sinergi)



Gambar 2.2 Contoh Use Case Diagram (Suendri, 2018)

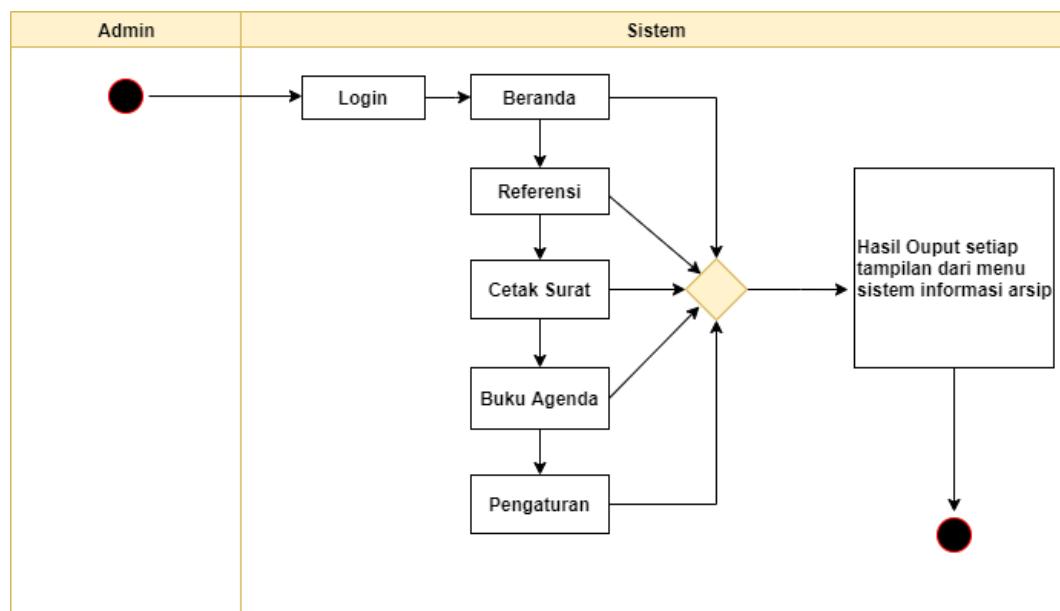
2. Activity Diagram

Activity Diagram merupakan diagram yang mengilustrasikan keseluruhan aliran dari aktivitas (Sulistyo, 2017).

Table 2.2 Simbol-simbol Activity Diagram

No.	Gambar	Nama	Keterangan
1.		<i>Activity</i>	Menunjukkan bagaimana setiap kelas saling berhubungan satu sama lain
2.		<i>Action</i>	State yang mencerminkan eksekusi dari sebuah aksi di dalam sistem
3.		<i>Initial Node</i>	Objek dicetak atau diawali.

No.	Gambar	Nama	Keterangan
4.		Activity Final Node	Objek dicetak dan diakhiri
5.		Decision	Dilambangkan untuk mendeskripsikan suatu kebijakan yang harus diputuskan pada kondisi tertentu.
6.		Line Connector	Dilambangkan untuk mengkoneksikan satu simbol dengan simbol lainnya.

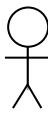
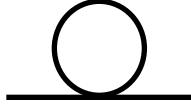
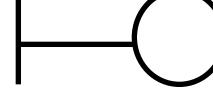


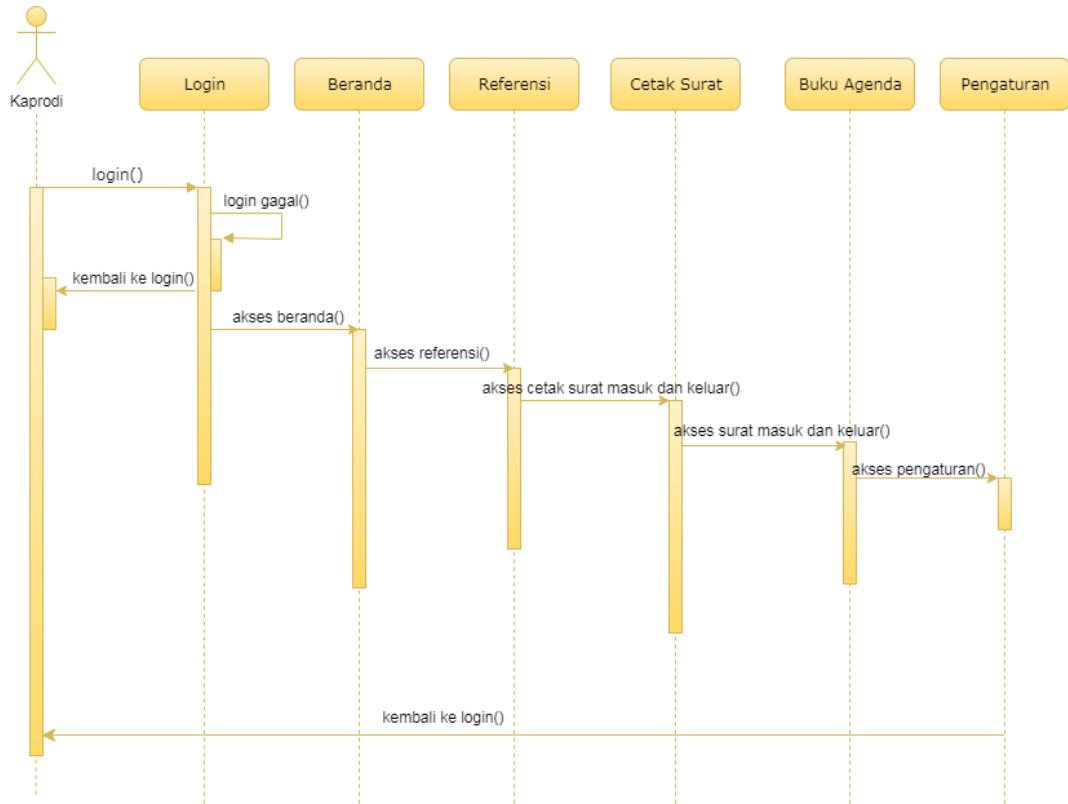
Gambar 2.3 Contoh Activity Diagram (Irawan & Aprillia, 2018)

3. Sequences Diagram

Sequences Diagram merupakan diagram yang menunjukkan *sequence* dari aktivitas dan hubungan antar *class* (Sulistyo, 2017).

Table 2.3 Simbol-simbol Sequences Diagram

No.	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Gambaran orang yang sedang bersinergi dengan sistem
2.		<i>Entity Class</i>	Gambaran relasi yang akan dilakukan
3.		<i>Boundary Class</i>	Menunjukkan gambaran dari form
4.		<i>Control Class</i>	Gambarkan antara <i>boundary</i> ke tabel
5.		<i>A focus of Control & Life Line</i>	Gambarkan tempat memulai dan mengakhiri <i>massage</i>
6.		<i>A Message</i>	Gambarkan pengiriman pesan



Gambar 2.4 Contoh Sequences Diagram (Irawan & Aprillia, 2018)

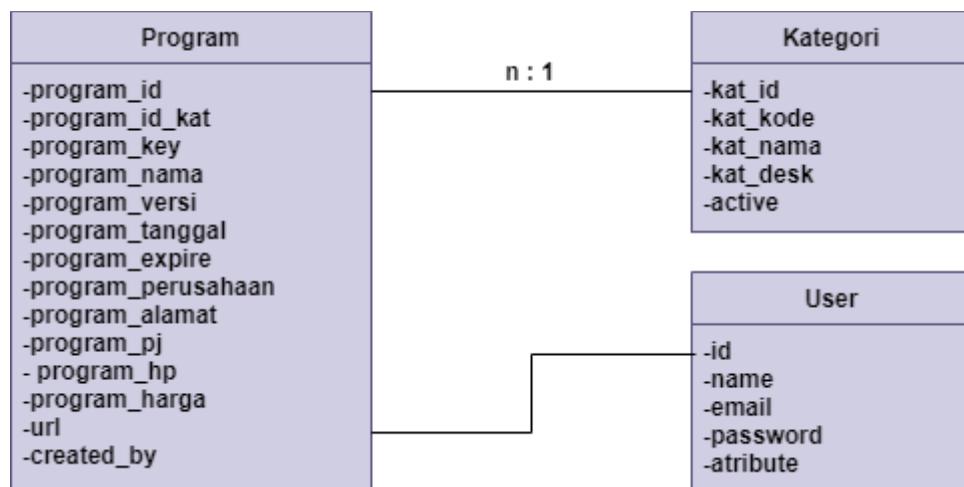
4. Class Diagram

Class Diagram merupakan diagram yang menunjukkan *relationship* antar *class* (Sulistyo, 2017)

Tabel 2.4 Simbol-simbol Class Diagram

No .	Gambar	Nama	Keterangan
1.		<i>Boundary Lifeline</i>	Gambaran diagram untuk suatu elemen yang berbeda, secara khas merupakan pengkoneksi <i>actor</i> dan layar.
2.		<i>Entity Lifeline</i>	Gambaran sebuah tempat atau operasi yang menangkap pengetahuan atau informasi dalam suatu sistem

No .	Gambar	Nama	Keterangan
3.		<i>Control Lifeline</i>	Gambaran dari suatu pengendalian yang menyusun dan menjadwalkan aktivitas komponen.
4.		<i>Message</i>	Perilaku sistem yang menandakan adanya suatu alur informasi atau perpindahan terarah antar elemen
5.		<i>Actor</i>	Gambaran seorang user system yang memulai jalan peristiwa/kejadian
6.		<i>Activation Bar</i>	Menggambarkan lamanya suatu pesan diproses
7.		<i>Note</i>	Menunjukkan catatan untuk komentar dari suatu pesan antar elemen



Gambar 2.5 Contoh Class Diagram (Suendri, 2018)

2.9 *Framework*

Framework diterjemahkan sebagai ‘kerangka kerja’. Secara garis besar *framework* adalah menyediakan kerangka kerja untuk membangun suatu sistem

sehingga pembangun sistem tidak harus merancang sistem dari awal. Dalam konsep *web framework*, *framework* menyediakan berbagai fungsi, sintaksis, *library*, *extension*, dan *template* siap pakai untuk mempercepat pekerjaan membangun *website*. Banyak jenis kerangka kerja ditemukan termasuk: Django, Yii, Zend, dan juga Laravel (S.ST, 2019). Aplikasi situs web, harus mengikuti aturan kerangka kerja. *framework* (dalam hal ini PHP framework), tidak wajib memikirkan kode perintah dasar / fungsi aplikasi website (S.Hut, 2010)

Menurut (S. Hut, 2010), beberapa keuntungan yang diperoleh dari penggunaan framework antara lain:

1. Waktu perancangan aplikasi web lebih efisien.
2. *Code* aplikasi *website* menjadi lebih gampang dibaca, karena sedikit dan bersifat pokok. Detailnya adalah kode dari *framework* dan ini mungkin tidak wajib dipikirkan, terjamin.
3. *Website* menjadi lebih mudah *di-maintenance*, karena tidak perlu fokus ke semua komponen kode *website*, paling utama kode sistem *framework*.
4. Tidak wajib lagi menciptakan kode penunjang aplikasi *website* seperti hubungan database, validasi form, GUI, dan *security*.
5. Pikiran akan lebih fokus ke kode jalan permasalahan laman, apa yang dimunculkan dan layanan yang diberikan dari aplikasi *website* tersebut.

2.10 Laravel

Laravel merupakan *framework* PHP dengan ciri *open source*. Dengan desain *Model-View-Controller* (MCV) yang dipakai agar membangun aplikasi *website*. *Framework* ini pertama sekali diciptakan oleh Taylor Otwell pada tanggal 22 Februari 2012 (Putih, 2014) selain itu ada juga yang beranggapan bahwa, Laravel ialah pengembangan *website* berlandaskan MVP yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas *software* dengan mengurangi *cost* pengembangan awal, biaya perawatan, serta untuk mengoptimalkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks ekspresif, jelas dan efisien (S.ST, 2019).

2.10.1 Sejarah Laravel

Laravel sendiri diciptakan oleh Taylor Otwell, dia memulai proyek laravel pada tahun 2011 dibulan April. Proyek ini didasari karena Otwell tidak menemukan *framework* yang mempunyai kelebihan update sejalan dengan versi PHP, menurutnya mengembangkan *framework* yang telah ada tidak mudah dengan keterbatasan sumber daya. Oleh karena itu Otwell membuat *framework* laravel dengan memberi ketentuan yaitu PHP versi 5.4 ke atas (Aminudin, 2015).

2.10.2 Konsep MVC

Model-View-Controller atau MVC merupakan suatu metode dalam proses pembuatan atau pembangunan sebuah sistem aplikasi dengan cara memisahkan data (*Model*) dari tampilan (*View*) dan cara bagaimana proses berlangsung (*Controller*) (Putih, 2014)

1. *Model*, model itu sendiri merepresentasikan struktur data. Dalam beberapa kasus model memuat fungsi-fungsi yang membantu seseorang dalam pengelolaan database seperti memasukkan data ke dalam database, memutakhirkan data dan lain-lain.
2. *View*, *View* adalah bagian yang mengontrol tampilan atau interface untuk pengguna. Dapat dikatakan merupakan halaman depan dari suatu sistem baik yang berbasis web, android maupun desktop.
3. *Controller*, *controller* itu sendiri merupakan bagian yang menghubungkan atau menjembatani antara model dan tampilan.
4. *Controller* berisi semua perintah yang terdapat dalam aplikasi yang memiliki fungsi untuk mengolah data dan mengirimkannya ke halaman *interface* pada sistem aplikasi.

2.10.3 Kelebihan Laravel

Adapun kelebihan framework Laravel adalah sebagai berikut (Aminudin, 2015):

1. Ekspresif

Laravel merupakan *framework* PHP ekspresif, artinya ketika melihat sintaks Laravel, seorang programmer diharapkan dapat segera mengetahui kegunaan sintaks tersebut walaupun belum pernah mempelajarinya apalagi menggunakannya.

2. Simple

Salah satu hal yang membuat Laravel begitu sederhana adalah *Elouent* ORM. Misalnya, kami ingin mengambil semua data di *tabel users*, jadi yang perlu dilakukan hanya membuat kelas model bernama *user*.

3. Accessible

Laravel dibuat dengan dokumentasi selengkap mungkin. *Code Developer* dari Laravel sendiri berkomitmen untuk selalu menyertakan dokumentasi yang lengkap setiap kali versi terbaru dirilis.

2.10.4 Fitur Laravel

Berikut adalah fitur-fitur yang tersedia pada framework laravel, (Putih, 2014) antara lain:

1. *Bundles*, Ikatan merupakan fitur yang mendukung sistem kemasan modular. Dengan fitur ini kita dapat dengan mudah untuk melakukan penambahan paket aplikasi ke dalam project kita. Laravel versi 4.x menggunakan *composer* sebagai manajer aplikasi.
2. *Elouent ORM (Object-RelationalMapping)*, merupakan implementasi dari PHP lanjutan dari *activerecord* yang menyediakan metode tersendiri dalam mengatur *relationship* antar obyek di database. Laravel *Query builder* adalah salah satu fitur yang disupport *Elouent*.
3. *Application logic*, merupakan fitur pengembangan aplikasi secara umum, baik dengan *controller* atau pendeklarasian *route*.
4. *Reverse routing*, merupakan fitur yang tepat dalam mendefinisikan hubungan antara *link* dan *route*, sehingga memungkinkan perubahan *link* dan *route* atau tanpa melakukan pengubahan *view*.

5. *Restfull controllers*, merupakan salah satu cara yang dapat digunakan untuk memisahkan logika antara HTTP GET dan POST.
6. *Class auto loading*, menyediakan fitur untuk *load* PHP *class* tanpa perlu melakukan *include*, *On-demand loading* hanya akan *me-load class* yang diperlukan.
7. *View composers*, merupakan kode *logic* yang dieksekusi ketika *view* di*load*.
8. *IoCcontainer*, memungkinkan obyek baru yang akan dihasilkan sesuai prinsip kontrol, dengan instansiasi opsional dan referensi dari obyek baru.
9. *Migrations*, merupakan fitur menyediakan sistem kontrol untuk skema *database*, sehingga memungkinkan untuk menghubungkan antara perubahan kode aplikasi dengan layout database, memudahkan *deploy* dan *update* aplikasi.
10. *Unit testing*, menyediakan fitur testing untuk mendeteksi atau mencegah kode ganda atau berulang (regresi) unit test ini dapat dijalankan melalui perintah *commandline*.
11. *Automatic pagination*, fitur yang memungkinkan pembuatan halaman/*paging* secara otomatis dengan metode yang sudah diintegrasikan ke laravel.

2.11 *Database*

Database merupakan kumpulan informasi, informasi ini disimpan dengan cara yang sangat terstruktur. Dengan mengeksplorasi struktur tersebut, anda bisa mengakses dan memodifikasi informasi dengan cepat dan tepat (Siahaan & Sianipar, 2019). Pendapat lain juga mengatakan bahwa *database* atau basis data adalah kumpulan data yang saling terhubung yang disimpan bersama pada suatu media tanpa adanya duplikasi data, sehingga mudah digunakan kembali dan tidak mengalami ketergantungan pada program yang akan menggunakannya dan dapat diakses oleh satu atau lebih program aplikasi secara optimal (Mandar, 2017).

Database adalah kumpulan file, tabel dan arsip yang saling berhubungan dan disimpan dalam media elektronik. Data tidak hanya disimpan begitu saja, tetapi dikelola dan ditata sehingga dikenal dengan *Database Management System* (DBMS). Dengan begitu, data yang tersimpan dalam database dapat tersusun rapi dan terstruktur sehingga memudahkan kita untuk mendapatkan informasi terkait *database*.

2.11.1 Database Management System (DBMS)

DBMS sendiri merupakan sebuah perangkat lunak (*software*) yang berfungsi untuk mengelola basis data. DBMS akan memudahkan pengguna dalam proses pembuatan database, mulai dari pembuatan, hingga proses yang berlaku di database, baik *berupa entry, edit, delete, Query data*, pembuatan laporan dan lain sebagainya secara efektif dan efisien (Yuhefizard, 2008)

2.12 PHP

PHP adalah singkatan dari “*HypertextPreprocessor*”, yang merupakan bahasa pemrograman web *server-side* yang bersifat *open source*. PHP adalah *script* yang terintegrasi dengan HTML dan berada di server (*server side HTML embedded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman *website* dinamis. Dinamis artinya halaman yang akan ditampilkan dibuat ketika halaman tersebut diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu *up-to-date*. Semua *script* PHP dijalankan di server tempat *script* dijalankan (Anhar, 2010).

2.13 HTML

HTML adalah singkatan dari *HyperTextMarkupLanguage*. Disebut *hypertext* karena dalam HTML sebuah teks biasa dapat berfungsi secara berbeda, kita dapat menjadikannya sebagai tautan yang dapat berpindah dari satu halaman ke halaman lain hanya dengan mengklik teks tersebut. Kemampuan teks ini disebut *HyperText*, meskipun dalam implementasinya nanti tidak hanya teks yang dapat digunakan sebagai *link*. Disebut *MarkupLanguage* karena bahasa HTML menggunakan tanda (*mark*), untuk menandai bagian-bagian teks. Misalnya, teks di

antara tanda tertentu akan dicetak tebal, dan jika di antara tanda lain akan tampak besar. Tanda ini dikenal sebagai tag HTML. HTML adalah bahasa dasar pembuatan web. Disebut *basic* karena dalam pembuatan web, jika hanya menggunakan HTML maka tampilan web terasa hambar. Ada banyak bahasa pemrograman web yang ditujukan untuk memanipulasi kode HTML, seperti JavaScript dan PHP, memahami HTML adalah hal yang paling awal. HTML bukanlah bahasa pemograman (bahasa pemograman), melainkan bahasa markup (*markup language*), hal ini kedengarannya sedikit aneh, namun jika Anda sudah terbiasa dengan bahasa pemograman lain, HTML tidak akan menemukan struktur yang biasa terdapat pada bahasa pemograman seperti IF, LOOP, atau variabel, HTML hanyalah bahasa struktural yang fungsinya untuk menandai bagian-bagian dari suatu halaman (Rerung, 2018).

2.14 MySQL

MySQL adalah salah satu jenis *data-base* server yang paling terkenal diantara *data-base* server lainnya dan saat ini banyak digunakan untuk membangun sebuah sistem aplikasi baik itu berbasis web maupun dekstop (Harianto et al., 2019). Selain itu, pendapat lain juga menjelaskan MySQL adalah perangkat lunak sistem manajemen basis data SQL yang dikenal sebagai DBMS (*Database Management System*), basis data ini *multithread, multi-user*. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah GNU *General Public License* (GPL), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus khusus (Anhar, 2010).

MySQL memiliki beberapa kelebihan dan keuntungan dibanding database lain (Wahana Komputer, 2010), di antaranya adalah:

1. Banyak ahli berpendapat MySQL merupakan server tercepat.
2. MySQL merupakan system manajemen database yang *Open Source* (kode sumbernya terbuka), yaitu software ini bersifat *free* atau bebas digunakan oleh perseorangan atau instansi tanpa harus membeli atau membayar kepada pembuatnya.
3. MySQL mempunyai performa yang tinggi tapi simpel.

4. Database MySQL mengerti bahasa SQL (*Structured Query Languange*).
5. MySQL dapat diakses melalui protocol ODBC (*Open Database Connectivity*) buatan Microsoft. Ini menyebabkan MySQL dapat diakses oleh banyak *software*.
6. Semua klien dapat mengakses server dalam satu waktu, tanpa harus menunggu yang lain untuk mengakses *database*.
7. Database MySQL dapat diakses dari semua tempat di internet dengan hak akses tertentu.
8. MySQL merupakan *database* yang mampu menyimpan data berkapasitas besar, sampai berukuran Gigabyte.
9. MySQL dapat berjalan di berbagai *operating system* seperti Linux, Windows, Solaris, dan lain lain.

2.15 XAMPP

XAMPP merupakan *software web server apache* yang sudah tersedia di *database server MySQL* dan *support php programming*. XAMPP adalah perangkat lunak gratis dan mudah digunakan yang mendukung instalasi di linux dan windows. Kelebihan lain yang Anda dapatkan dari xampp adalah dengan sekali install *Apache web server*, *MySQL database server*, *phpsupport* (php4 php5) dan beberapa modul lainnya hanya berbeda jika versi windows selalu berupa instalasi grafis dan linux. dalam bentuk file terkompresi tar. .gz. Keunggulan lain yang berbeda dengan versi untuk *Windows* adalah memiliki fitur untuk mengaktifkan server secara grafis, sedangkan Linux masih menjadi perintah di konsol. Oleh karena itu versi linux sulit untuk dioperasikan (Puspitasari, 2011).

2.16 Web Browser

Web Browser merupakan perangkat lunak atau *software* yang berfungsi menampilkan dan melakukan interaksi dengan dokumen-dokumen yang disediakan oleh *server web*(Puji et al., 2015). Jadi kesimpulannya *web browser* adalah tempat mengakses informasi *web* atau melakukan transaksi *via web*. *Web Browser* yang saat ini sudah banyak dan sangat sering dipakai ialah Opera, UC Browser, Safari dan lain-lain.

2.17 Studi Sejenis

Berikut adalah penelitian-penelitian yang selaras/sejenis dengan penelitian yang dilakukan peneliti sebagai bahan referensi dalam melakukan penelitian ini :

Tabel 2.5 Referensi Studi Sejenis

No .	Nama	Judul	Kelebihan	Kelemahan
1	Adi Putra Nugraha, Kodrat Imam Satoto, Kurniawan Teguh Martono (Jurnal Teknologi dan Sistem Komputer – Vol 2 No. 2, April 2014)	Aplikasi Pemesanan Makanan Berbasis Mobile Pada Rumah Makan “Lek Nonong”	Sistem ini sudah berbasis mobile android dan befokus pada pemesanan menu	Sistem ini tidak menyediakan pemesanan tempat atau meja
2	Novhirtamely Kahar, Reny Wahyuning Astuti, Reni (Jurnal Informatika Vol 7 No. 2, Juli 2013)	Aplikasi Pemesanan Makanan Online Berbasis Web pada Rumah Makan Pagi Sore Sipin Jambi	Sistem ini tersedia 3 jenis pesanan yaitu Delivery Service, Catering Service (pesan menu paket dengan jumlah banyak), dan Catering Ruang Minang (pesan menu paket tempat khusus di ruang minang)	Pada sistem ini pembayaran masih menggunakan cash dan belum menggunakan bank transfer atau sejenisnya

No .	Nama	Judul	Kelebihan	Kelemahan
3	Yuda Irawan, Herianto, Refni Wahyuni (Jurnal Teknologi dan Open SourceVol 2 No. 2, Desember 2019 Hal : 108-115)	Aplikasi Pemesanan Makanan Berbasis Web di CateringCindel aras Pekanbaru	Sistem ini mempermudah konsumen khususnya pelanggan catering cindelaras di pekanbaru untuk melakukan pesanan	Sistem ini belum menjelaskan metode pembayaran
4	Debbie Defrina, Dewi Putrie Lestari (Jurnal Ilmiah Informatika dan Komputer Vol 22 No. 3, Desember 2017)	Aplikasi Pemesanan Makanan dan Minuman Online Berbasis Mobile Browser Pada Restoran Tiga Saudara	Sistem ini memberikan tipe yaitu pelayanan melalui internet, sistem ini menggunakan WLAN, sehingga untuk bisa melakukan pemesanan perlu terhubung dengan Wi-Fi yang disediakan	

Dari referensi studi sejenis yang telah dikumpulkan diatas, dapat dilihat bahwa tidak ada penelitian yang membahas layanan pemesanan pada restoran cindelaras kota medan dengan menggunakan framework laravel, sehingga pelaku peneliti mengambil topik pembahasan tersebut.

BAB III

METODE PENELITIAN

3.1 Metode Penelitian

Metode penelitian yang diterapkan dalam penelitian ini ialah metodologi kualitatif, metodologi kualitatif adalah metode penelitian yang menciptakan gambaran berupa aplikasi yang didapat dari catatan, *interview*, dokumen-dokumen hasil observasi terkait tulisan maupun gambar (Samsudin et al., 2014). Metode kualitatif ialah jenis penelitian yang hasil temuannya tidak berasal dari statistik atau bentuk hitungan lainnya.

Terdapat dua pengolahan data dalam penelitian ini, yaitu reduksi data dan koding data.

1. Reduksi data, ialah menggolongkan dan mengorganisasi data sinkron dengan topik yang dimana data itu dihasilkan dari penelitian.
2. Koding data, ialah pengolahan materi atau informasi dengan memberi kode-kode terhadap segmen-segmen data.

Analisis yang digunakan ialah analisis kualitatif yang mana didalamnya terdapat upaya bekerja dengan data, mengorganisasikan data, dan memilah untuk menemukan apa yang penting dari data untuk suatu penelitian.

3.2 Tempat dan Waktu Penelitian

Tempat penelitian ini dilaksanakan di Restoran Ayam Penyet Cindelaras yang beralamat di Jln. Marelan Raya No. 55, Tanah Enam Ratus, Kec. Medan Marelan, Kota Medan. Penelitian ini dilaksanakan pada bulan Juni 2019 sampai September 2019.

Tabel 3.1 Waktu Penelitian dan Pengerjaan Sistem

<i>Kegiatan</i>	<i>Jadwal</i>				<i>September</i>				<i>Oktober</i>				<i>November</i>				<i>Desember</i>			
	M I	M II	M III	M IV	M I	M II	M III	M IV	M I	M II	M III	M IV	M I	M II	M III	M IV	M I	M II	M III	M IV
Riset Kebutuhan Institusi/ Perusahaan	■																			
Pengumpulan Data		■■■■																		
Pengajuan Proposal					■■■■															
Seminar Proposal						■■■■														
Analisis Data Sistem							■■■■		■■■■											
Perancangan Sistem										■■■■										
Implementasi Sistem											■■■■									
Pengujian Sistem												■■■■								
Dokumentasi Sistem													■■■■							

Adapun tahap yang akan dilakukan sebagai berikut :

1. Penelitian Kebutuhan Kelembagaan / Perusahaan

Peneliti akan melakukan penelitian terhadap permasalahan yang ada pada rumah makan Ayam Penyet Cindelaras guna mencari solusi untuk mengatasi permasalahan yang ada.

2. Pengajuan Proposal Skripsi

Setelah menemukan solusi yang dirasa efektif, peneliti mengajukan sebuah judul sebagai syarat pengajuan proposal skripsi.

3. Penjadwalan

Dalam melakukan penelitian, peneliti membutuhkan penjadwalan yang akan membantu dalam menentukan batasan waktu dalam merancang sistem pelayanan pemesanan pada Restoran Cindelaras Penyet Ayam.

4. Seminar Proposal

Diselenggarakan seminar proposal untuk menyatakan judul penelitian yang diajukan diterima atau tidak dengan mencantumkan proposal skripsi.

5. Pengumpulan Data

Setelah judul penelitian diterima, maka peneliti akan mengumpulkan data untuk keperluan perancangan sistem.

3.3 Cara Kerja

3.3.1 Metode Pengumpulan Data

Metode ini dilakukan dengan cara mengumpulkan data yang diperlukan untuk penelitian pada Restoran Cindelaras Kota Medan. Berikut ini metode pengumpulan data yang digunakan dalam penelitian :

1. Observasi

Tahap ini dilakukan dengan cara mengumpulkan data melalui pengamatan langsung terhadap objek penelitian untuk melihat secara dekat kegiatan yang dilakukan untuk memperkuat data pada penelitian ini. Pelaksanaan observasi ini dilakukan dengan mendatangi lokasi Restoran Ayam Penyet Cindelaras yaitu Cabang Adam Malik dan juga Marelan.

2. Wawancara

Tahap ini dilakukan dengan cara mengumpulkan data melalui pihak cindelaras seperti bertanya secara langsung sehingga informasi yang kita inginkan jelas, caranya seperti melakukan wawancara kepada pihak restoran ayam penyet cindelaras yaitu bapak Robby Iksandi, selaku penanggung jawab restoran ayam penyet cindelaras cabang adam malik dan bapak Rian Eko Syahputra selaku penanggung jawab restoran ayam penyet cindelaras cabang tanah 600 marelan. Wawancara ini dilakukan bertempat di dua lokasi cabang ayam penyet cindelaras

dengan memberikan pertanyaan yang meliputi seputar restoran ayam penyet cindelaras dan seputar menu dan reservasi.

Wawancara ini dilakukan pada :

Waktu : 1. Jum'at, 3 Januari 2020

2. Sabtu, 4 Januari 2020

Tempat : 1. AP Cindelaras Adam Malik, Jln. Adam Malik Medan

2. AP Cindelaras Marelan, Jln. Marelan Raya No. 55 (Di Sebelah Alfamart Marelan), Tanah Enam Ratus, Medan

Berdasarkan wawancara yang telah dilakukan, hasil dari wawancara ini berupa data-data maupun informasi spesifik terkait reservasi dan menu di restoran ayam penyet cindelaras dikota medan.

3. Analisis Dokumen

Pada tahap ini, peneliti menggunakan analisis dokumen untuk menjelaskan mengenai dokumen-dokumen yang akan digunakan dalam sistem pelayanan pemesanan pada restoran ayam penyet cindelaras.

Tabel 3.2 Analisis Dokumen

1	Nama Dokumen	Data Menu
	Sumber	Manager Cabang APCL
	Rangkap	1
	Fungsi	Sebagai penentu harga makanan dan minuman pada Restoran APCL
	Item Data	Nama makanan, nama minuman, harga, jumlah
2	Nama Dokumen	Data Reservasi
	Sumber	Manager Cabang APCL
	Rangkap	1

	Fungsi	Digunakan sebagai data pemesanan tempat dan menu
	Item Data	Id restoran, nama user, nominal, email, waktu, tanggal
3	Nama Dokumen	Data Restoran
	Sumber	Manager Cabang APCL
	Rangkap	1
	Fungsi	Digunakan sebagai data laporan
	Item Data	Alamat, struktur organisasi, no_rek, email, password, image

3.3.2 Metode Pengembangan Sistem

Pengembangan sistem yang digunakan pada aplikasi ini adalah Metode *Waterfall* dari model pendekatan SDLC (*System Development Life Circle*) dalam Sistem Informasi Layanan Pemesanan Pada Restoran Cindelaras Kota Medan dengan Menggunakan *Framework* Laravel.

Metode *Waterfall* adalah metode pengembangan yang umum digunakan oleh para peneliti sistem, adapun tahapan yang dapat diuraikan ialah :

1. Analisis Kebutuhan Perangkat Lunak

Pada tahap ini penulis akan melakukan wawancara (*interview*) pada pihak cindelaras yaitu *manager/penanggung jawab* di cabang masing-masing, wawancara ini bertujuan untuk mengambil data yang akan digunakan untuk kebutuhan membuat *field-field* pada pengembangan sistem.

a. Data

Pada tahap ini, peneliti akan mencari data dan informasi melalui buku atau *e-book* yang berkaitan dengan reservasi/pemesanan.

b. Perangkat Keras

Adapun perangkat keras yang digunakan dalam penelitian ini ialah laptop dengan spesifikasi sebagai berikut:

- 1) Layar 14”
- 2) Memiliki tingkat resolusi 1366 x 768 pixels
- 3) Prosesor Intel® Celero® CPU B830
- 4) RAM 2 GB DDR3
- 5) Harddisk 500 GB

c. Perangkat Lunak

Adapun perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut:

- 1) SistemOperasi, Windows 7
- 2) XAMPP 3.2.2
- 3) MySqL 5.1.37
- 4) PHP 7
- 5) Bootstrap 4.3.1
- 6) Sublime Text
- 7) Chrome
- 8) Microsoft Office 2007
- 9) Draw IO
- 10) Laravel 6.0

2. Desain

Sebelum pengkodean program, data yang diperoleh diubah *menjadi field-field* yang akan disusun secara sistematis sehingga membentuk suatu tabel. Adapun tabel yang akan dibuat yaitu: tabel_admins, tabel_admin_role, tabel_cabangs, tabel_kategori_menus, tabel_konfirmasi_bayars, tabel_mejas, tabel_menus,tabel_reservas, tabel_reservas_menus, tabel_roles, tabel_users.

Dalam tahap desain ini juga terdapat rancangan *Class Diagram*, *Activity Diagram*, *Use Case Diagram*, dan *Sequence Diagram*.

3. Pembuatan Kode Program

Ditahap ini, aplikasi dikembangkan sesuai desain yang telah disusun secara sistematis dalam kode program PHP, HTML dan database MySQL serta framework laravel.

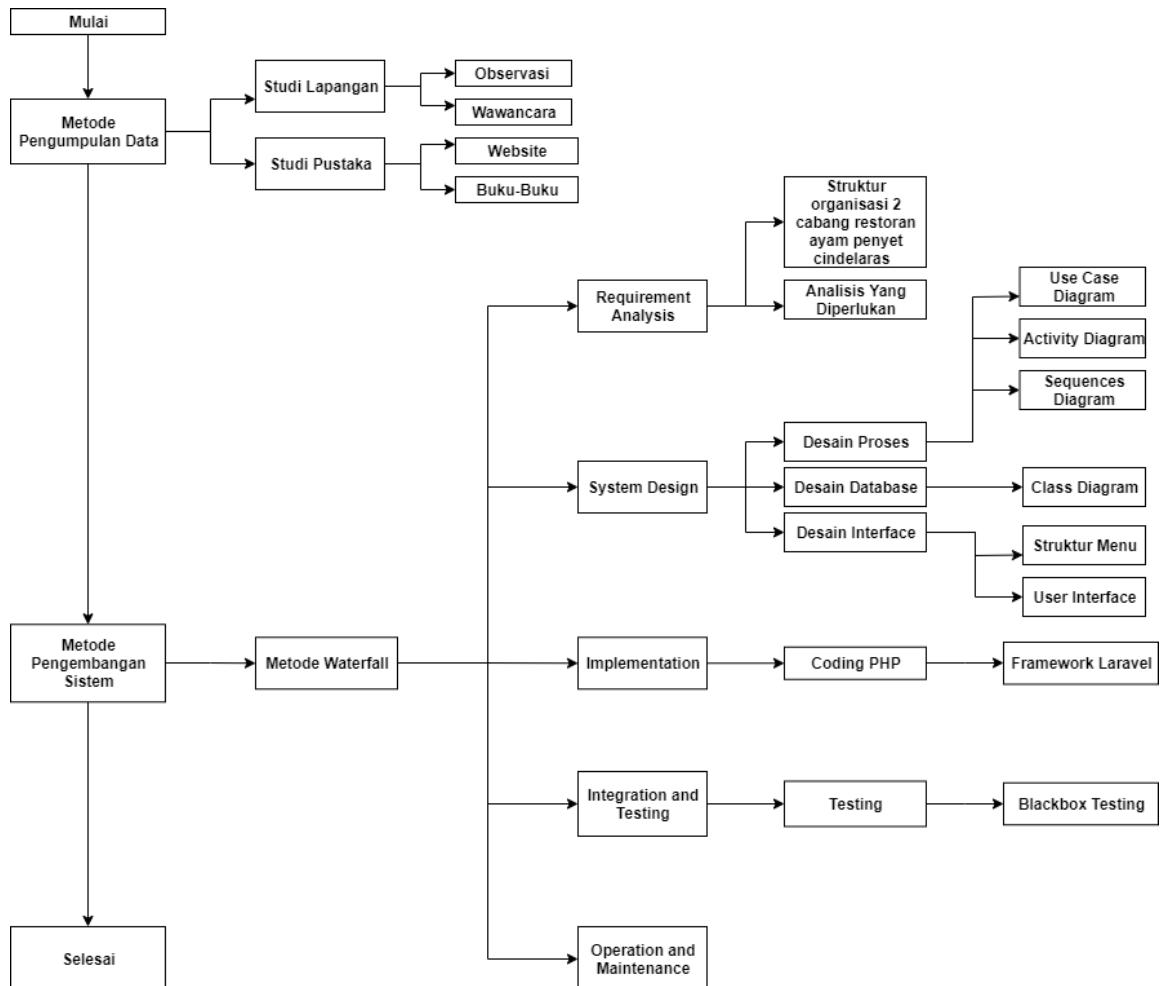
4. Pengujian

Setelah selesai tahap coding, aplikasi akan dilakukan pengujian kepada *user* pengguna sehingga dapat diketahui apakah telah berjalan sesuai dengan yang diharapkan atau belum.

5. Pemeliharaan

Setelah tahap pengujian dilakukan beberapa revisi sesuai kebutuhan, aplikasi sudah dapat di pergunakan oleh *user* yang dalam hal ini adalah konsumen Rumah Makan Cindelaras dan juga *staff* sebagai admin pada cabang-cabang Ayam Penyet Cindelaras di Kota Medan. Apabila aplikasi sudah dipergunakan, maka diperlukan untuk melakukan perawatan.

3.4 Kerangka Berpikir Penelitian



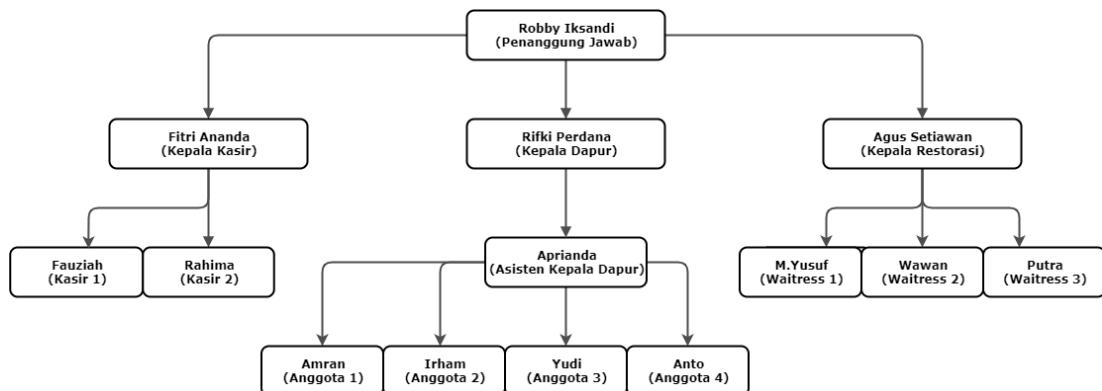
Gambar 3.1 Kerangka Berpikir Penelitian

BAB IV

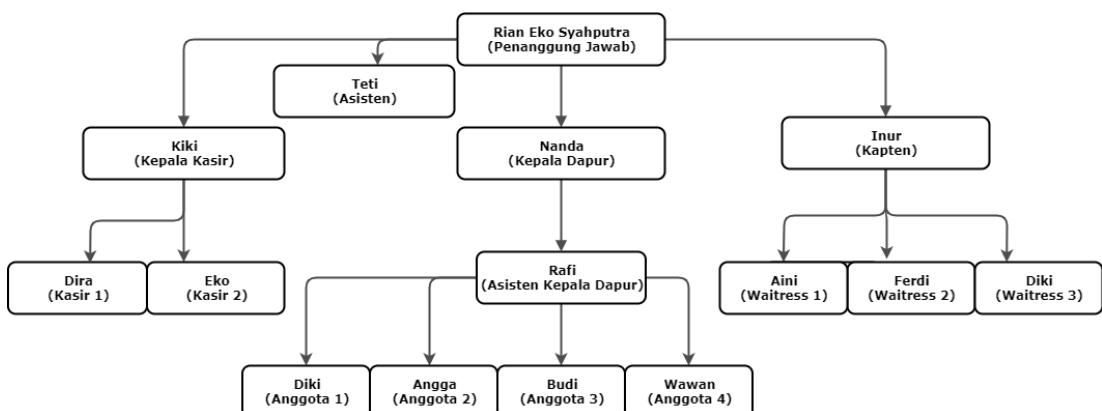
HASIL DAN PEMBAHASAN

4.1 Struktur Organisasi

Struktur organisasi Restoran Ayam Penyet Cindelaras Kota Medan dibentuk untuk dikelompokkan, dibagi, dan dikoordinasikan sesuai pekerjaan atau posisi masing-masing *staff* untuk mencapai tujuan yang di harapkan. Berikut 2 struktur organisasi Restoran Ayam Penyet Cindelaras Kota Medan Cabang Adam Malik dan Marelan.



Gambar 4.1 Struktur Organisasi Restoran APCL Cabang Adam Malik



Gambar 4.2 Struktur Organisasi Restoran APCL Cabang Marelan

4.1.1 Visi dan Misi

Visi :

“Ayam Penyet CINDELARAS, Restoran Tradisional Terbesar, Cita Rasa Khas Indonesia”.

Misi :

1. Menciptakan merek Ayam Penyet CINDELARAS sehingga memiliki kekuatan yang luar biasa.
2. Membentuk Restoran Ayam Penyet CINDELARAS sebagai bisnis kuliner berbasis jaringan yang bersih dan hygienis sehingga mampu memberikan kepuasan bagi para pelanggannya.
3. Mengutamakan cita rasa tradisional khas Indonesia dengan konsep yang modern.
4. Memajukan bisnis hingga menjangkau seluruh kota yang ada di Indonesia.

4.2 Analisis Yang Diperlukan

Sebelum ke tahap perancangan sistem yang baru, peneliti akan melakukan analisis terhadap sistem yang sedang berjalan saat ini. Tujuannya untuk membandingkan kinerja sistem yang telah ada dengan sistem yang akan diusulkan. Adapun sistem yang sedang berjalan pada sistem pelayanan pemesanan tempat dan menu di restoran Ayam Penyet Cindelaras di dua cabang ini ialah masih dengan mendatangi restoran, lalu pelanggan yang ingin melakukan reservasi harus memesan minimal sehari sebelumnya dengan membayar minimal lima puluh ribu untuk panjar yang nantinya akan dipotong total makanan dan minuman yang dipesan, menu dan tempat dipesan secara bersamaan, pada kedua cabang ini sistem pembayaran masih menggunakan cara tunai. Sistem yang sedang berjalan ini tidak efisien dan hemat waktu maka dari itu sistem informasi akan memberikan efek terhadap eksistensi restoran ayam penyet cindelaras kota medan sehingga minat masyarakat diharapkan akan terus meningkat sesuai dengan perkembangan teknologi.

Berikut kekurangan sistem yang sedang berjalan :

1. Proses reservasi masih dilakukan secara manual
2. Proses transaksi reservasi masih dilakukan secara tunai
3. Sistem informasi belum sepenuhnya mendukung untuk sistem reservasi di restoran ayam penyet cindelaras kota medan

Adanya sistem usulan yang akan dibangun terdiri dari beberapa pengguna, diantaranya:

1. User/Pelanggan
 - a. User masuk ke halaman website
 - b. User memasukkan data diri, email, jumlah meja , beserta tanggal dan jam untuk reservasi
 - c. User memilih menu makanan dan minuman
 - d. User melihat laporan reservasi dan melakukan pembayaran lewat rekening
 - e. User mengupload bukti pembayaran
 - f. Setelah selesai user dapat keluar dari website
2. Admin
 - a. Admin melakukan *log in* dengan memasukkan *username* dan *password*
 - b. Admin mengatur meja untuk konsumen yang melakukan reservasi
 - c. Admin dapat menambahkan meja dan menghapus meja
 - d. Admin mengkonfirmasi bukti pembayaran konsumen yang melakukan reservasi
 - e. Admin menambahkan atau menghapus menu
 - f. Admin menambahkan cabang restoran
 - g. Setelah selesai admin dapat keluar (*log out*)

4.3 Perancangan Sistem

Pada pembahasan ini akan menjelaskan proses perancangan sistem yang dituangkan dalam bentuk UML (*Unified Modeling Language*). UML ialah bahasa grafis yang standar untuk pemodelan perangkat lunak berorientasi objek.

Terdapat empat diagram yang akan dibuat yaitu *Use Case Diagram*, *Activity Diagram*, *Sequences Diagram* dan *Class Diagram*.

4.3.1 *Use Case Diagram*

Adapun *use case diagram* yang digunakan dalam pembangunan sistem ini adalah sebagai berikut:

1. *Actor*
 - a. *User*, bisa melakukan reservasi, bisa memilih cabang, bisa melihat menu makanan dan minuman, bisa memesan makanan dan minuman, bisa melakukan pembayaran dan konfirmasi pembayaran, dimana user bisa memilih sesuai keinginan di restoran cindelaras mana yang diinginkan agar user dapat melakukan reservasi.
 - b. *Admin*, dalam sistem ini admin bisa mengatur meja user, mengatur menu, menambah cabang cindelaras, menerima pembayaran, mengkonfirmasi DP, secara garis besar admin memiliki semua akses atas sistem ini.
2. *Use Case Description*

Deskripsi untuk setiap use case diagram ialah sebagai berikut:

- a. *Home*

<i>Use Case Name</i>	Home
<i>Scenario</i>	User membuka halaman utama pada website
<i>Triggering Event</i>	-
<i>Brief Description</i>	Membuka website dan user dapat melihat halaman utama sistem reservasi
<i>Actors</i>	User
<i>Related Use Case</i>	-
<i>Stakeholders</i>	-

<i>Pre-conditions</i>	User mencari halaman website “Ayam Penyet Cindelaras”	
<i>Post conditions</i>	User telah masuk pada halaman website “Ayam Penyet Cindelaras”	
<i>Flow of Activities</i>	Melihat Home	
	<i>Actors</i>	Sistem
	1. Akses website ayam penyet cindelaras	2. Muncul form halaman utama / home website ayam penyet cindelaras

b. Reservasi

<i>Use Case Name</i>	Reservasi
<i>Scenario</i>	User melakukan pemesanan meja secara online di website
<i>Triggering Event</i>	-
<i>Brief Description</i>	Melakukan reservasi atau pemesanan meja untuk ketersediaan supaya pada saat pelanggan datang, meja sudah disiapkan.
<i>Actors</i>	User,Admin
<i>Related Use Case</i>	-
<i>Stakeholders</i>	-
<i>Pre-conditions</i>	Memasukkan data berupa nama, email, no.hp, tanggal, waktu dan total orang serta pilih cabang
<i>Post conditions</i>	Data diri telah disimpan

<i>Flow of Activities</i>	Melakukan Reservasi	
	<i>Actors</i>	Sistem
	3. Akses menu reservasi 3. Masukkan data diri beserta waktu, tanggal, jumlah orang, dan pilih cabang store lalu klik “booking”	4. Muncul form data diri untuk membuat reservasi 4. Permintaan user sudah terpenuhi dan meja berhasil di booking

c. Pemesanan Makanan dan Minuman

<i>Use Case Name</i>	Pemesanan Makanan dan Minuman	
<i>Scenario</i>	User melakukan pemesanan menu secara online di website	
<i>Triggering Event</i>	-	
<i>Brief Description</i>	Melakukan pemesanan menu untuk ketersediaan supaya pada saat pelanggan datang, menu sudah disiapkan.	
<i>Actors</i>	User	
<i>Related Use Case</i>	-	
<i>Stakeholders</i>	-	
<i>Pre-conditions</i>	Memilih menu makanan dan minuman apa saja yang akan <i>user</i> pesan.	
<i>Post conditions</i>	Data menu makanan dan minuman telah disimpan	
<i>Flow of Activities</i>	Pemesanan Makanan dan Minuman	
	<i>Actors</i>	Sistem

	<ol style="list-style-type: none"> 1. Akses menu makanan dan minuman. 5. Memilih menu dengan men-klik “tambah” 	<ol style="list-style-type: none"> 2. Muncul form tampilan makanan dan minuman. 4. Permintaan user sudah terpenuhi dan menu berhasil ditambah
--	--	---

d. Upload Bukti Pembayaran

<i>Use Case Name.</i>	Upload Bukti Pembayaran.	
<i>Scenario</i>	User melakukan mengunggah bukti/screenshot ke dalam sistem	
<i>Triggering Event</i>	-	
<i>Brief Description</i>	Mengupload bukti ke halaman konfirmasi sehingga admin bisa memproses reservasi	
<i>Actors</i>	User	
<i>Related Use Case</i>	-	
<i>Stakeholders</i>	-	
<i>Pre-conditions</i>	Mengupload bukti pembayaran	
<i>Post conditions</i>	Reservasi telah selesai dilaksanakan	
<i>Flow of Activities</i>	Upload Bukti Pembayaran	
	<i>Actors</i>	Sistem

	<p>1. Akses menu konfirmasi pembayaran</p> <p>3. Menginput bukti transfer dan nominal pembayaran DP, No.Rek User lalu klik “selesai”</p>	<p>2. Muncul form tampilan konfirmasi pembayaran</p> <p>4. Permintaan user sudah terpenuhi dan pembayaran akan diverifikasi oleh admin</p>
--	--	--

e. Bukti Booking

<i>Use Case Name</i>	Bukti Booking	
<i>Scenario</i>	Admin dapat melihat bukti booking user sebagai verifikasi	
<i>Triggering Event</i>	-	
<i>Brief Description</i>	Bukti booking yang terkirim kepada admin menandakan bahwa verifikasi user untuk reservasi harus dilaksanakan	
<i>Actors</i>	Admin,User	
<i>Related Use Case</i>	-	
<i>Stakeholders</i>	-	
<i>Pre-conditions</i>	Status reservasi masih pending	
<i>Post conditions</i>	Status reservasi sudah terkonfirmasi	
<i>Flow of Activities</i>	Bukti Booking	
	<i>Actors</i>	Sistem

	<p>1. Akses ke menu reservasi</p> <p>3. Memeriksa status pembayaran pelanggan</p> <p>5. Admin akan memasukkan user ke dalam list konfirmasi DP</p>	<p>2. Muncul form tampilan menu reservasi</p> <p>4. Tampilan akan berubah jika pelanggan telah melakukan pembayaran</p> <p>6. Permintaan admin sudah terpenuhi</p>
--	--	--

f. Konfirmasi DP

<i>Use Case Name</i>	Konfirmasi DP	
<i>Scenario</i>	Admin melakukan pemeriksaan untuk melihat pelanggan yang sudah membayar atau belum	
<i>Triggering Event</i>	-	
<i>Brief Description</i>	Pelanggan melakukan pembayaran agar admin dapat memverifikasi pembayaran pelanggan sehingga reservasi dapat diselesaikan	
<i>Actors</i>	Admin	
<i>Related Use Case</i>	-	
<i>Stakeholders</i>	-	
<i>Pre-conditions</i>	Status reservasi masih pending	
<i>Post conditions</i>	Status reservasi sudah terkonfirmasi	
<i>Flow of Activities</i>	Konfirmasi DP	
	<i>Actors</i>	Sistem

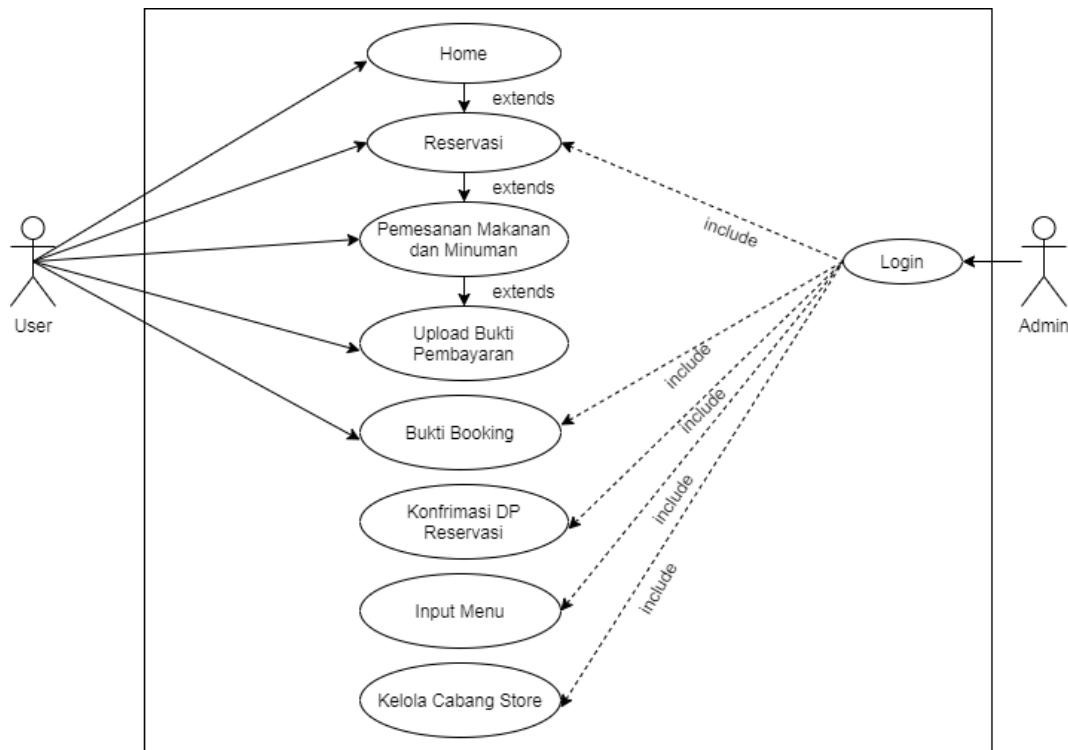
	<ol style="list-style-type: none"> 1. Akses konfirmasi DP 3. Klik “konfirmasi” 	<ol style="list-style-type: none"> 2. Muncul form tampilan konfirmasi DP 4. Pembayaran yang dilakukan user telah diverifikasi oleh admin
--	--	--

g. Input Menu

<i>Use Case Name</i>	Input Menu	
<i>Scenario</i>	Admin menginput menu makanan dan minuman	
<i>Triggering Event</i>	-	
<i>Brief Description</i>	Admin melakukan penambahan menu makanan dan minuman sesuai dengan pembaharuan yang ada pada masing-masing cabang cindelaras	
<i>Actors</i>	Admin	
<i>Related Use Case</i>	-	
<i>Stakeholders</i>	-	
<i>Pre-conditions</i>	Daftar menu sayuran belum tersedia	
<i>Post conditions</i>	Daftar menu sayuran telah tersedia	
<i>Flow of Activities</i>	Input Menu	
	<i>Actors</i>	Sistem
	<ol style="list-style-type: none"> 1. Akses input menu 3. Masukkan data yang diperlukan seperti nama menu, jenis menu, harga menu, lalu klik “tambah” 	<ol style="list-style-type: none"> 2. Muncul form tambah daftar menu 4. Permintaan admin sudah terpenuhi dan menu telah tersedia

h. Kelola Cabang *Store*

<i>Use Case Name</i>	Kelola Cabang Store	
<i>Scenario</i>	Admin menginput cabang cindelaras yang ada di sekitar kota medan	
<i>Triggering Event</i>	-	
<i>Brief Description</i>	Admin melakukan penambahan cabang-cabang cindelaras agar sistem reservasi dapat meluas penggunaannya	
<i>Actors</i>	Admin	
<i>Related Use Case</i>	-	
<i>Stakeholders</i>	-	
<i>Pre-conditions</i>	Sistem masih menampilkan dua cabang	
<i>Post conditions</i>	Sistem telah menampilkan tiga cabang	
<i>Flow of Activities</i>	Kelola Cabang Store	
	<i>Actors</i>	Sistem
	1. Akses menu cabang store 3. Memasukkan nama dan alamat cabang cindelaras	2. Menampilkan form untuk menambahkan cabang cindelaras 4. Permintaan admin sudah terpenuhi, dan cabang telah berhasil ditambah



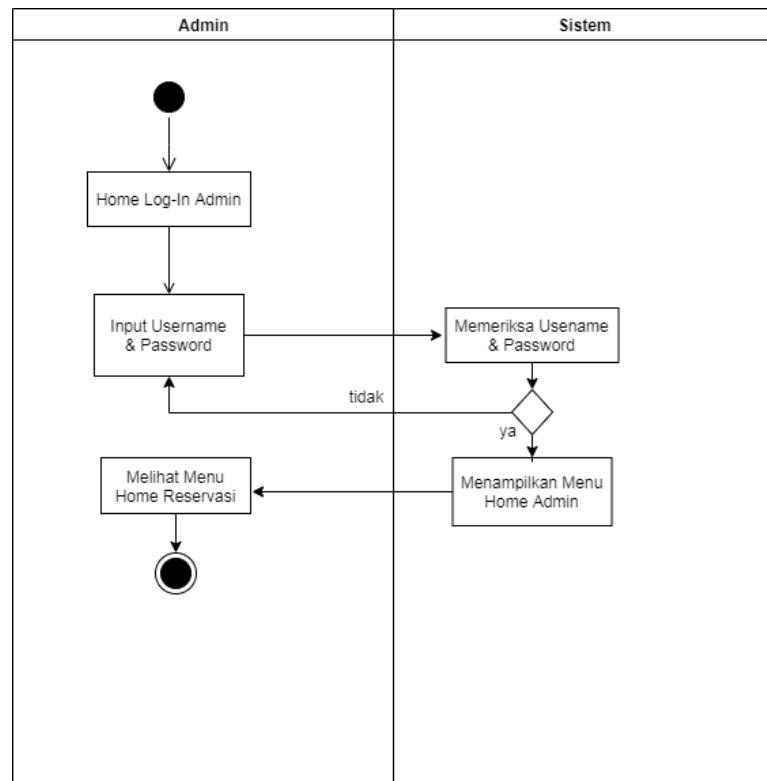
Gambar 4.3 Use Case Diagram Sistem Reservasi Cindelaras

4.3.2 Activity Diagram

Activity diagram adalah aktivitas yang mengilustrasikan keseluruhan langkah-langkah dalam perancangan sistem reservasi dari bagaimana alur itu dimulai sampai alur itu berakhir.

1. Activity Diagram Login

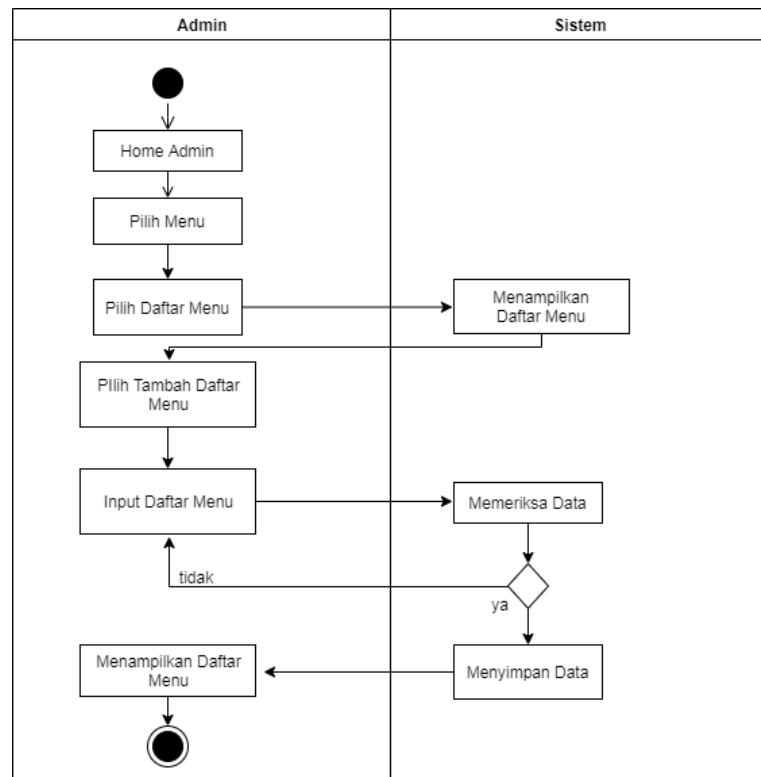
Aktifitas ini dilakukan pada saat login, dan aktivitas ini hanya bisa dilakukan oleh admin. Pada saat akan memasuk *home admin*, admin diwajibkan menginput username beserta *password*. Jika *username* dan *password* yang dimasukkan benar, maka sistem akan menampilkan *home admin*, kemudian admin dapat melihat *home reservasi*. Berikut gambar rancangan *activity diagram* login:



Gambar 4.4 Activity Diagram Login Admin

2. *Activity Diagram* Menu

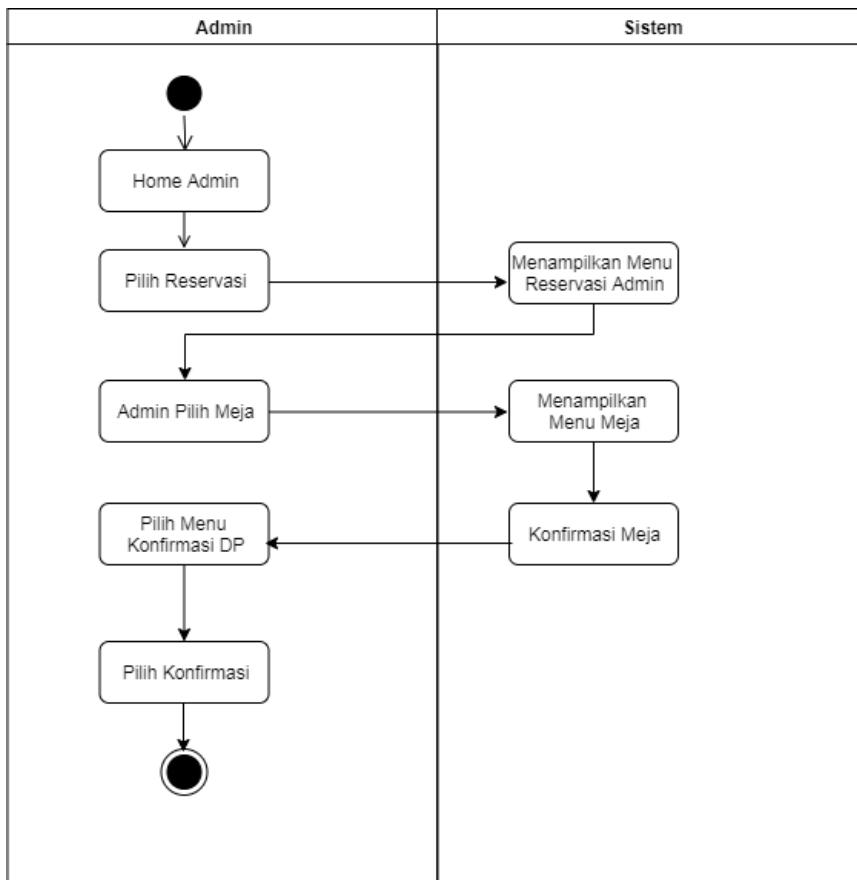
Aktivitas ini hanya bisa dilakukan oleh admin, dimana admin bisa menambahkan menu makanan dan minuman serta harga di masing-masing cabang cindelaras yang terdapat di kota medan. Berikut gambar rancangan *activity diagram* menu :



Gambar 4.5 Activity Diagram Menu

3. Activity Diagram Konfirmasi Reservasi Admin

Aktivitas konfirmasi reservasi ini hanya bisa dilakukan oleh admin. Dimulai dengan admin memasuki halaman utama admin, lalu admin membuka form reservasi dan data para pelanggan yang melakukan reservasi pun muncul, setelah itu admin melakukan pemilihan meja untuk user yang melakukan reservasi serta telah melakukan pembayaran, lalu kemudian data user masuk ke *form* konfirmasi DP dan dinyatakan telah selesai melakukan reservasi. Berikut gambar perancangan activity diagram konfirmasi reservasi:

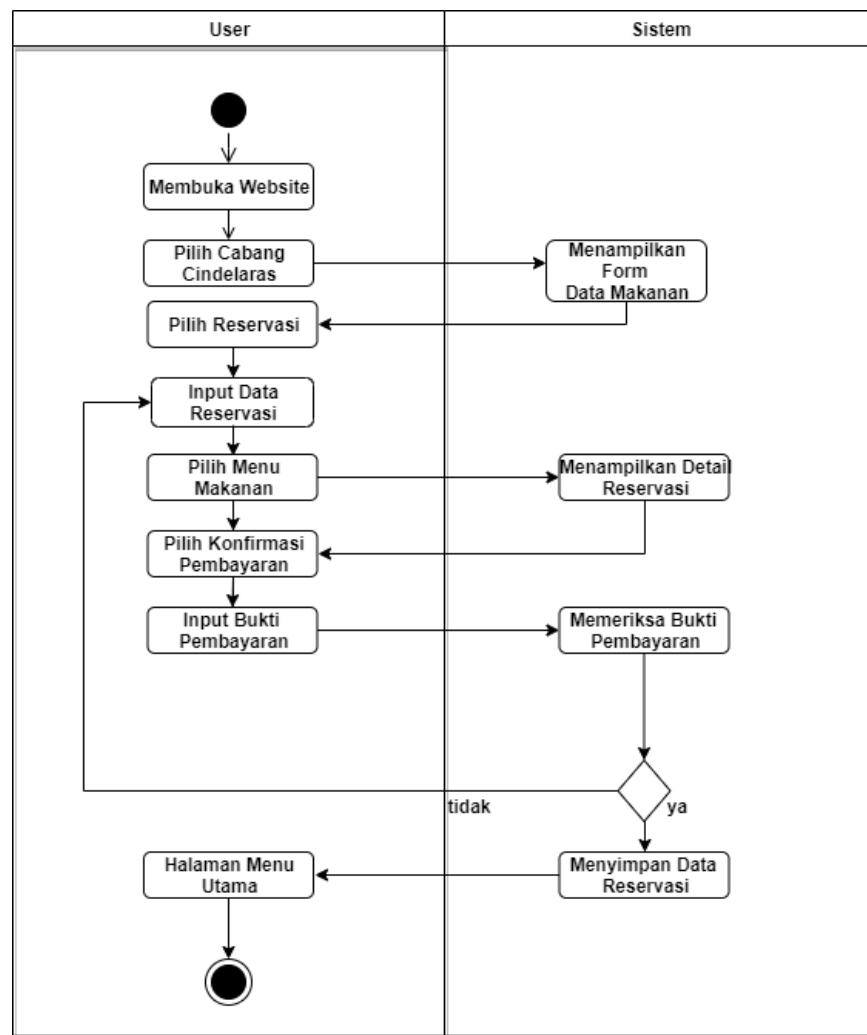


Gambar 4.6 Activity Diagram Konfirmasi Reservasi Admin

4. Activity Diagram User

Aktivitas ini dilakukan oleh pelanggan sebagai *user* tanpa melakukan login, disini user dapat melakukan reservasi dengan membuka website lalu user melihat cabang cindelaras mana yang akan dilakukan reservasi kemudian sistem akan menampilkan menu-menu yang terdapat pada cabang cindelaras tersebut, setelah itu user melakukan reservasi sesuai keinginannya dengan menginput data yaitu nama, email, nomor telepon,tanggal reservasi, jam reservasi, jumlah orang dan cabang nya lalu beralih ke halaman menu makanan, disini user memilih menu makanan dan minuman dan beralih ke halaman detail reservasi, disini data lengkap yang di input user tertera beserta jumlah uang yang harus dibayar sebagai DP, user diharuskan melakukan pembayaran sebelum melewati waktu yang ditentukan lalu user wajib melakukan konfirmasi pembayaran pada halaman

berikutnya dengan mengupload bukti pembayaran, dan sistem akan menyimpan data yang nantinya akan di verifikasi oleh admin, setelah semua selesai maka user bisa kembali ke menu utama website. Berikut gambar perancangan *activity diagram user*:



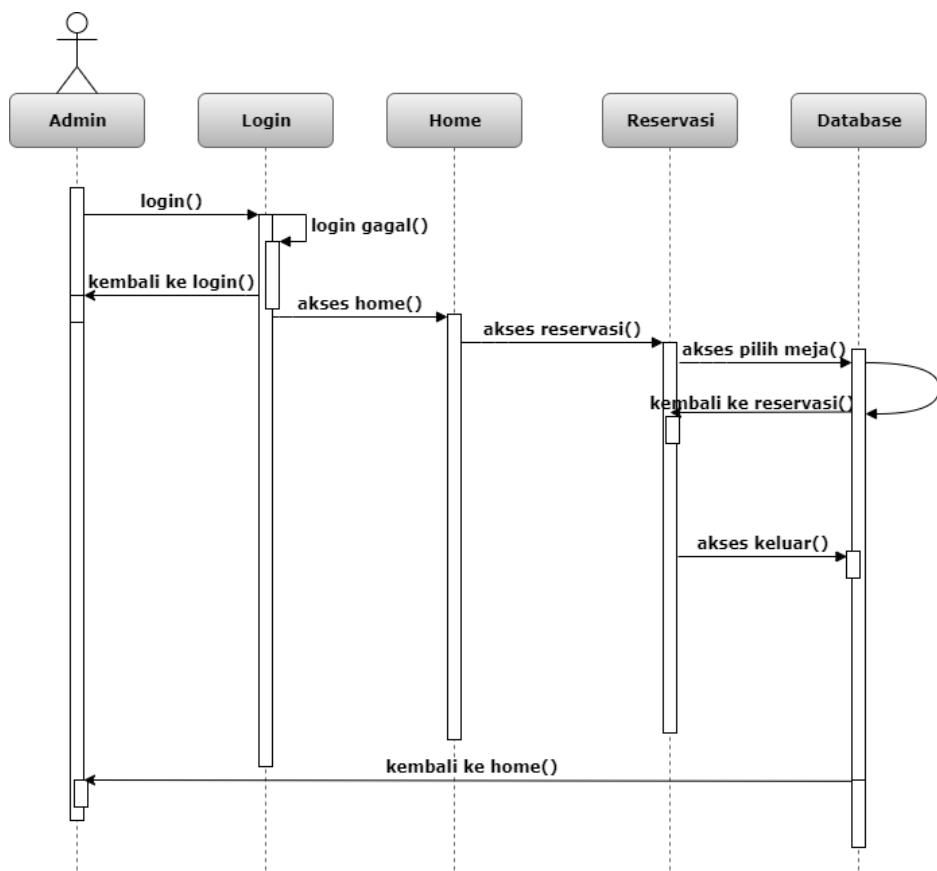
Gambar 4.7 Activity Diagram User

4.3.3 Sequences Diagram

Sequences diagram adalah gambaran interaksi objek melalui tanda atau petunjuk untuk komunikasi diantara objek-objek tersebut.

1. Sequences Diagram Reservasi Admin

Untuk menampilkan form reservasi, disini admin sebagai aktor melakukan login pada sistem admin, jika berhasil maka admin bertemu dengan halaman home, dan memilih *button* reservasi setelah itu admin melihat data user yang telah melakukan pembayaran dan mengorganisir data user ke meja-meja restoran, lalu admin kembali ke halaman home.

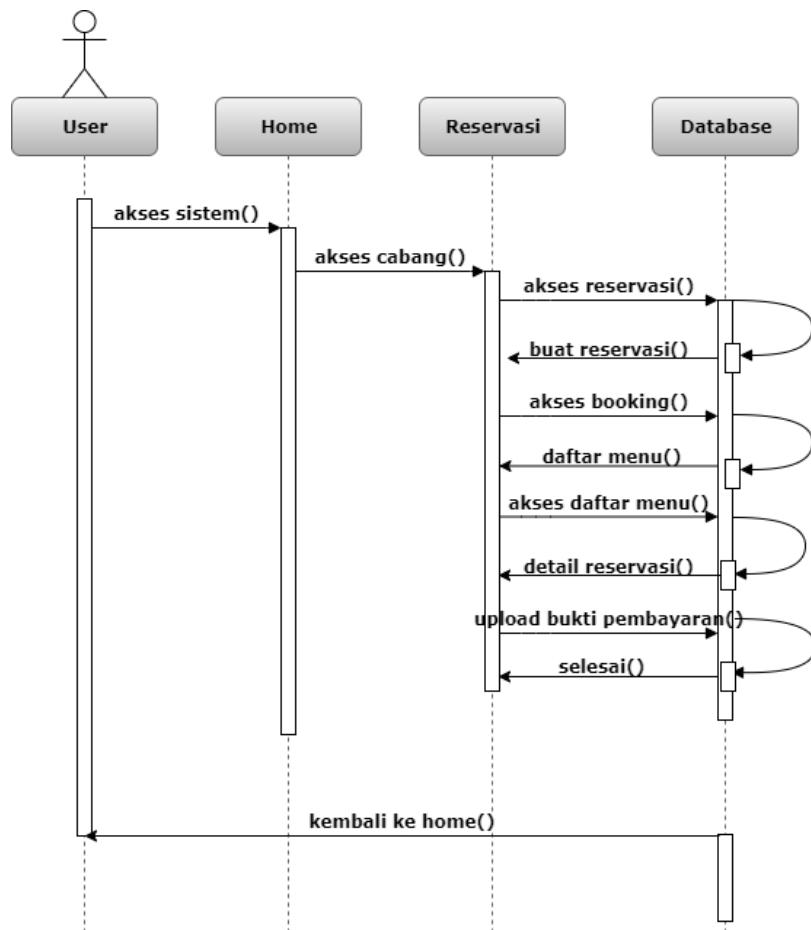


Gambar 4.8 Sequences Diagram Reservasi Admin

2. Sequences Diagram Reservasi User

Untuk menampilkan form reservasi *user*, disini user sebagai aktor membuka website untuk melihat menu *home*, di halaman home user memilih cabang setelah itu user membuka menu reservasi dan memberikan keterangan data diri untuk reservasi, kemudian *user* juga memilih menu yang disediakan dan detail

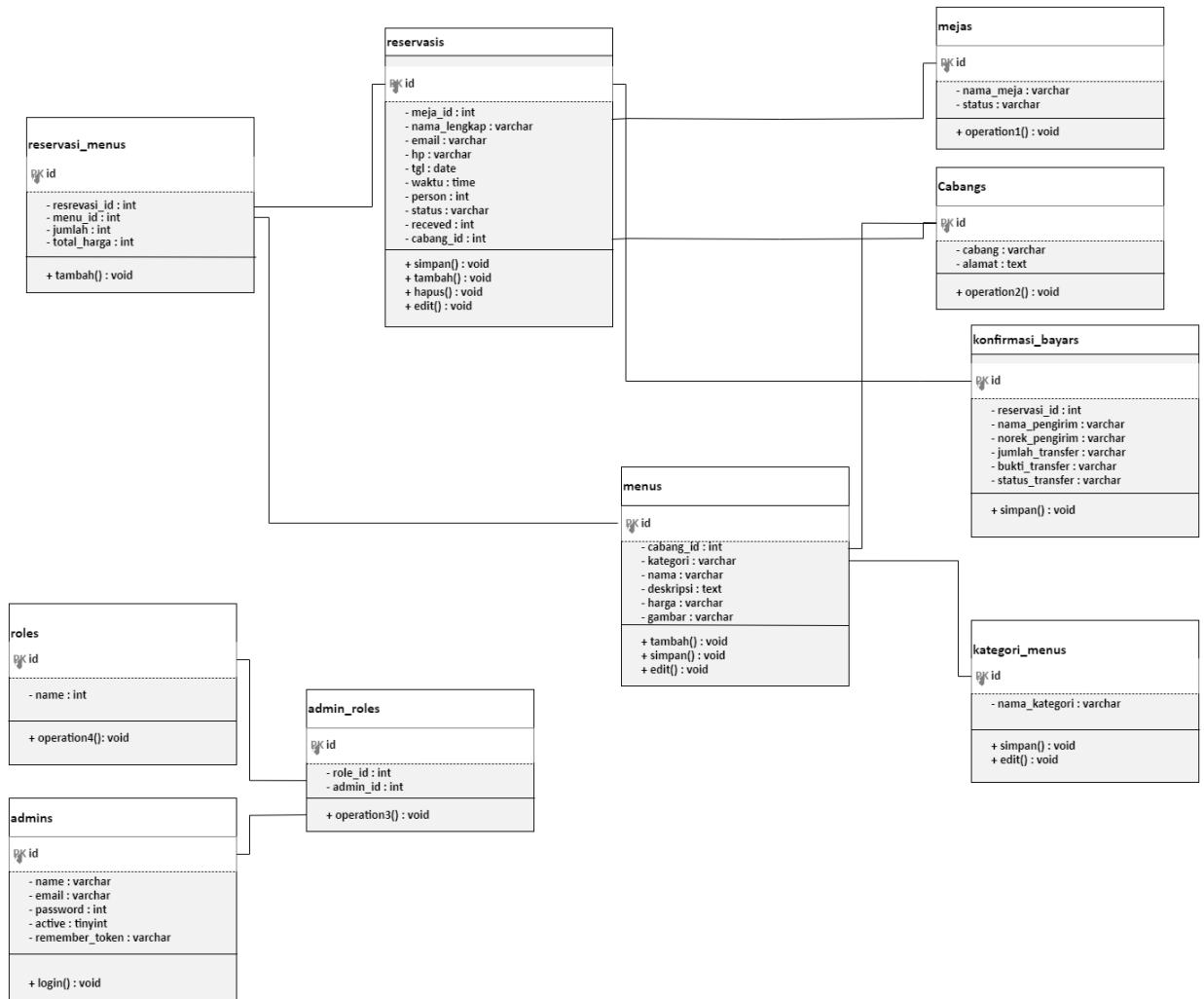
reservasi akan muncul, terakhir user akan mengupload bukti pembayaran dan selesai.



Gambar 4.9 Sequences Diagram Reservasi User

4.3.4 Class Diagram

Class diagram adalah gambaran struktur sistem yang di deskripsikan dengan menggunakan class-class. *Class diagram* juga memberikan pandangan lebih luas mengenai suatu sistem dengan cara menunjukkan kelas serta hubungannya satu sama lain.



Gambar 4.10 Class Diagram

4.6 Desain Database

1. Tabel User

Tabel user menyimpan data diri user pada sistem, berikut merupakan daftar *field*, *type* dan ukuran data pada tabel user:

Tabel 4.1 Tabel User

No.	Nama Field	Type	Ukuran
1	Id	bigint	20

No.	Nama Field	Type	Ukuran
2	Name	varchar	255
3	Email	varchar	255
4	email_verified_at	timestamp	-
5	Password	varchar	255
6	remember_token	varchar	100
7	created_at	timestamp	-
8	updated_at	timestamp	-

2. Tabel Reservasi Menu

Tabel reservasi menu menyimpan data pesanan makanan dan menu beserta jumlah harga pada sistem, berikut merupakan daftar *field*, *type* dan ukuran data pada tabel reservasi menu:

Tabel 4.2 Tabel Reservasi Menu

No.	Nama Field	Type	Ukuran
1	Id	bigint	11
2	reservasi_id	int	11
3	menu_id	int	11
4	Jumlah	int	11
5	total_harga	int	11
6	created_at	timestamp	-
7	updated_at	timestamp	-

3. Tabel Reservasi

Tabel reservasi menyimpan seluruh data user pada sistem, berikut merupakan daftar *field, type* dan ukuran data pada tabel reservasi:

Tabel 4.3 Tabel Reservasi

No.	Nama Field	Type	Ukuran
1	Id	bigint	20
2	meja_id	int	11
3	nama_lengkap	varchar	255
4	Email	varchar	255
5	Hp	varchar	255
6	Tgl	date	100
7	Waktu	time	-
8	Person	int	11
9	Status	varchar	255
10	Received	int	11
11	cabang_id	int	11
12	reservasi_token	varchar	255
13	created_at	timestamp	-
14	updated_at	timestamp	-

4. Tabel Menu

Tabel menu menyimpan data menu makanan, minuman, harga serta gambar makanan dan minuman pada sistem, berikut merupakan daftar *field, type* dan ukuran data pada tabel menu:

Tabel 4.4 Tabel Menu

No.	Nama Field	Type	Ukuran
1	Id	bigint	20
2	cabang_id	int	11
3	Kategori	varchar	255
4	Nama	varchar	255
5	Deskripsi	text	-
6	Harga	varchar	255
7	Gambar	varchar	255
8	created_at	timestamp	-
9	updated_at	timestamp	-

5. Tabel Meja

Tabel meja menyimpan data jumlah meja pada sistem, berikut merupakan daftar *field*, *type* dan ukuran data pada tabel meja:

Tabel 4.5 Tabel Meja

No.	Nama Field	Type	Ukuran
1	Id	bigint	20
2	nama_meja	varchar	255
3	Status	int	11
4	created_at	timestamp	-
5	updated_at	timestamp	-

6. Tabel Konfirmasi Bayar

Tabel konfirmasi bayar menyimpan data dari user berupa nomor rekening, jumlah transfer, dan bukti transfer pada sistem, berikut merupakan daftar *field*, *type* dan ukuran data pada tabel konfirmasi bayar:

Tabel 4.6 Tabel Konfirmasi Bayar

No.	Nama Field	Type	Ukuran
1	Id	bigint	20
2	reservasi_id	int	11
3	nama_pengirim	varchar	255
4	norek_pengirim	varchar	255
5	jumlah_transfer	varchar	255
6	bukti_transfer	varchar	255
7	status_transfer	varchar	255
8	created_at	timestamp	-
9	updated_at	timestamp	-

7. Tabel Admin

Tabel admin menyimpan data admin berupa username dan password admin pada sistem, berikut merupakan daftar *field*, *type* dan ukuran data pada tabel admin:

Tabel 4.7 Tabel Admin

No.	Nama Field	Type	Ukuran
1	Id	int	10
2	Name	varchar	255

No.	Nama Field	Type	Ukuran
3	Email	varchar	255
4	Password	varchar	255
5	Active	tinyint	1
6	remember_token	varchar	100
7	created_at	timestamp	-
8	updated_at	timestamp	-

8. Tabel Cabang

Tabel cabang menyimpan data cabang-cabang restoran cindelaras pada sistem, berikut merupakan daftar *field*, *type* dan ukuran data pada tabel cabang:

Tabel 4.8 Tabel Cabang

No.	Nama Field	Type	Ukuran
1	Id	bigint	20
2	Cabang	varchar	255
3	Alamat	text	-
4	created_at	timestamp	-
5	updated_at	timestamp	-

9. Tabel Kategori Menu

Tabel kategori menu menyimpan data menu makanan dan minuman sesuai dengan kategori nya pada sistem, berikut merupakan daftar *field*, *type* dan ukuran data pada tabel kategori:

Tabel 4.9 Tabel Kategori Menu

No.	Nama Field	Type	Ukuran
1	Id	bigint	20
2	nama_kategori	varchar	255
3	created_at	timestamp	-
4	updated_at	timestamp	-

4.7 Desain *Interface*

Desain interface ialah desain perancangan antar muka pada sistem yang akan dibangun.

1. Form Home Admin

The interface design for the Home Admin form is structured as follows:

- Header:** A top navigation bar containing a "Log Out" button.
- Left Sidebar:** A vertical sidebar on the left side with the following navigation links:
 - Dashboard
 - Meja
 - Reservasi
 - Konfirmasi DP
 - Menu
 - Laporan
- Right Main Area:** The main content area displays several data elements:
 - Table Data:** A table showing the status of 8 tables (Meja 1 to Meja 8). The table has two columns: "Logo" and "Meja". The data is as follows:

Logo	Meja Kosong 8
Logo	Meja Terisi 0
Logo	Reservasi 0
Logo	Menu 20
 - Table Layout:** Below the table data, there is a 2x4 grid of boxes labeled Meja 1 through Meja 8.

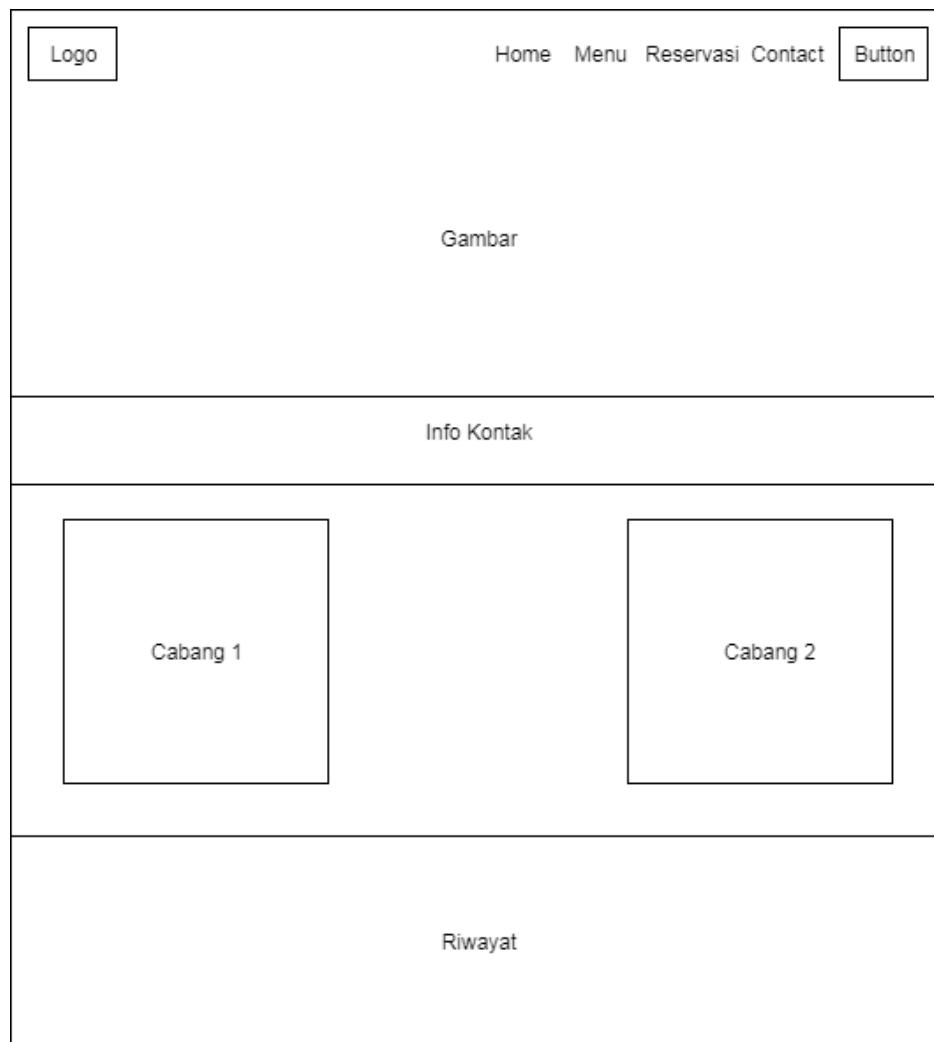
Gambar 4.11 Form Home Admin

2. *Form Login Admin*

The diagram illustrates a wireframe for an 'Admin Login' form. At the top, the title 'Admin Login' is centered. Below the title, there are two input fields: 'Email Address' on the left and a corresponding empty rectangular box on the right for entering the email. Below these, the label 'Password' is on the left, followed by an empty rectangular box for entering the password. At the bottom center of the form is a single button labeled 'Log In'.

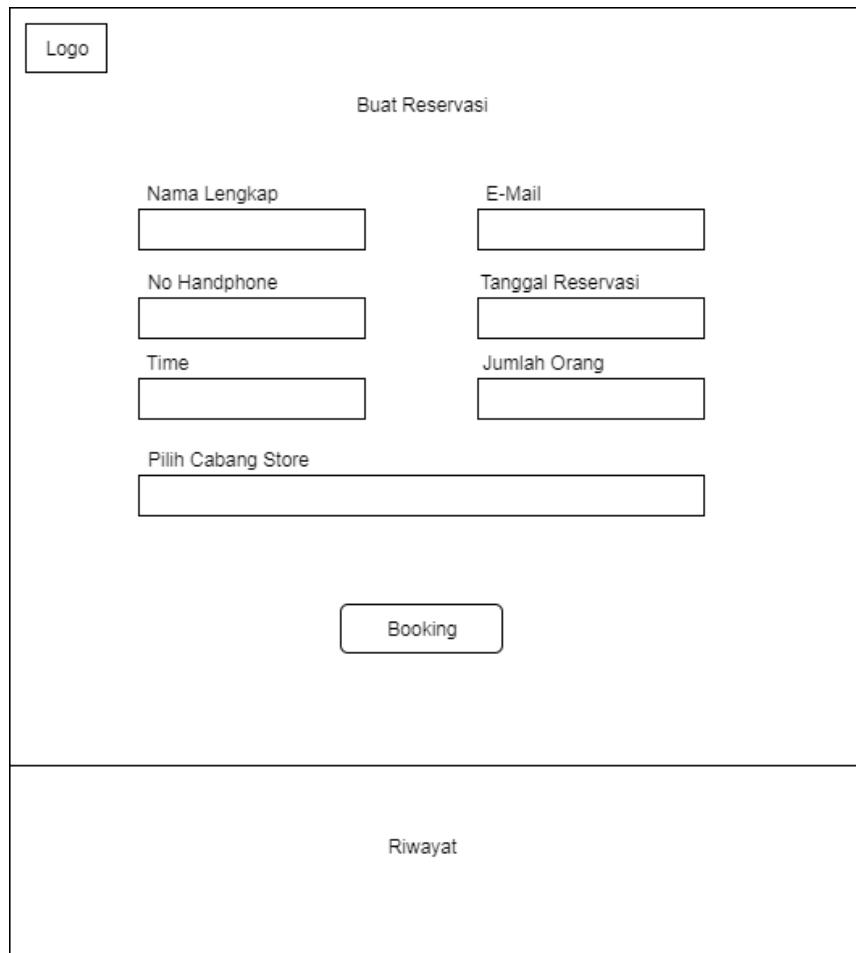
Gambar 4.12 Form Login Admin

3. *Form Home*



Gambar 4.13 Form Home

4. Form Reservasi



The image shows a user interface for a reservation form. At the top left is a placeholder for a 'Logo'. Below it is a button labeled 'Buat Reservasi'. The form is divided into two columns. The left column contains fields for 'Nama Lengkap' (Full Name) and 'No Handphone' (Phone Number). The right column contains fields for 'E-Mail' (Email) and 'Tanggal Reservasi' (Reservation Date). Below these are fields for 'Time' and 'Jumlah Orang' (Number of People). A section labeled 'Pilih Cabang Store' (Select Branch Store) contains a long input field. At the bottom center is a button labeled 'Booking'. A large, empty rectangular area below the form is labeled 'Riwayat' (History).

Logo

Buat Reservasi

Nama Lengkap

No Handphone

E-Mail

Tanggal Reservasi

Time

Jumlah Orang

Pilih Cabang Store

Booking

Riwayat

Gambar 4.14 Form Reservasi

4.8 Implementasi dan Pengkodean

Dalam tahap ini akan di terapkan rancangan yang telah di buat menjadi sebuah sistem *website* atau aplikasi yang membantu para pelanggan yang ingin melakukan pemesanan tempat dan menu pada restoran cindelaras di Kota Medan. Dalam pengkodeannya, aplikasi sistem pelayanan ini dibangun menggunakan:

1. Kebutuhan Perangkat Keras

Adapun *hardware* yang digunakan dalam pembuatan sistem aplikasi layanan pemesanan ini ialah laptop dengan spesifikasi sebagai berikut:

- a) Layar14”
- b) Memiliki tingkat resolusi 1366 x 768 pixels
- c) Prosesor Intel.® Celero® CPU B830
- d) RAM 2 GB DDR3
- e) Harddisk 500 GB

2 Kebutuhan Perangkat Lunak.

Adapun *software* yang digunakan dalam pembuatan aplikasi ini ialah sebagai berikut:

- a) Sistem Operasi, Windows 7
- b) XAMPP 3.2.2
- c) MySQL 5.1.37
- d) PHP 7
- e) Bootstrap 4.3.1
- f) Sublime Text
- g) Chrome
- h) Microsoft Office 2007
- i) Draw IO
- j) Laravel 6.0

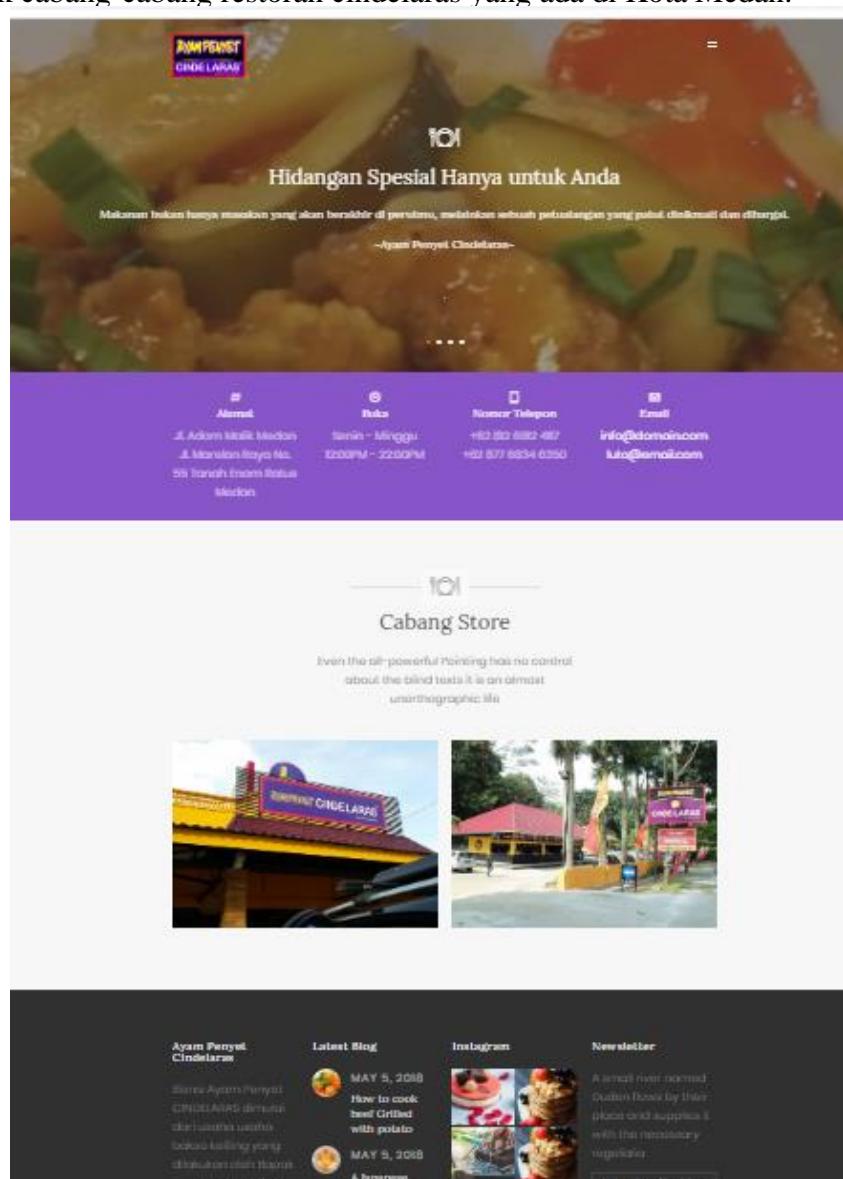
4.9 Testing

4.9.1 Interface

Tahap pengujian sistem ialah dimana tahap pemeriksaan apakah sistem sudah berjalan sesuai kehendak atau tidak.

1. Home

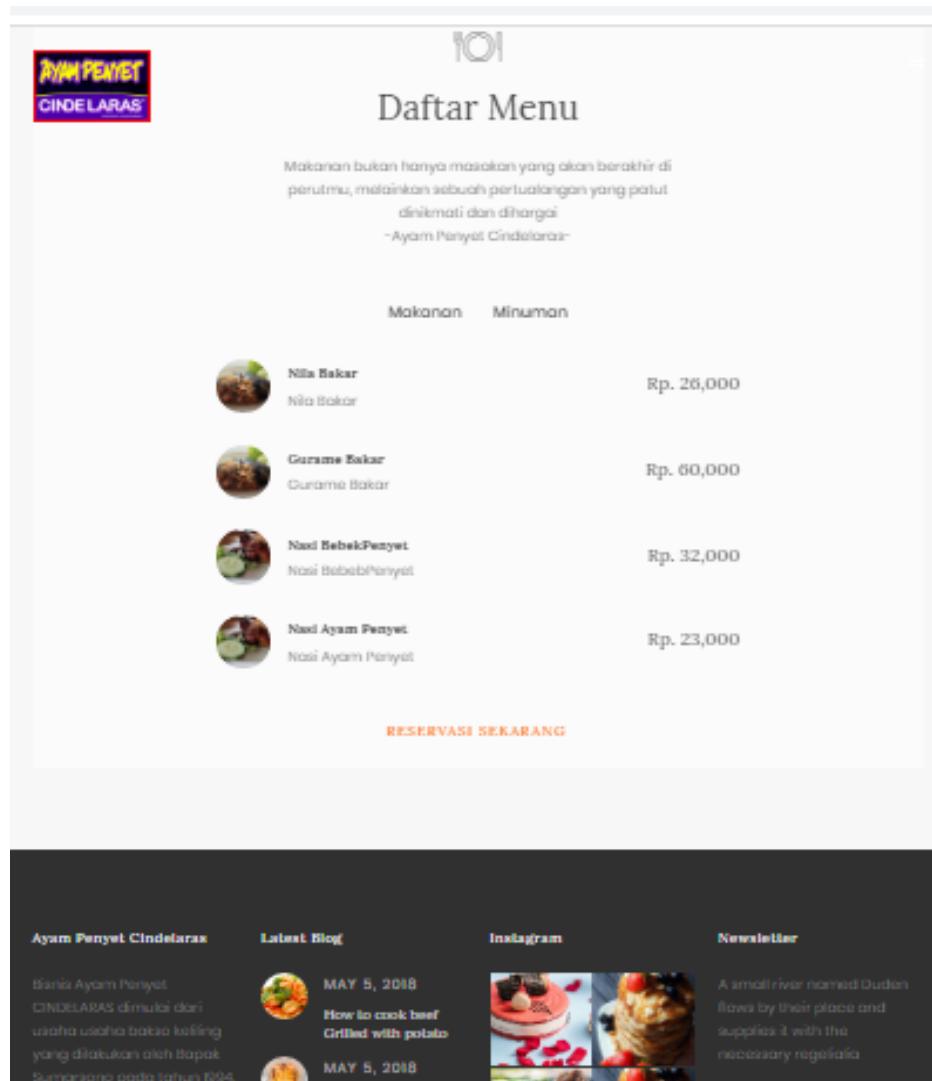
Menu home ini adalah tampilan utama dalam sistem layanan pemesanan, di dalamnya berisi keterangan alamat, jam buka restoran, nomor telepon restoran, dan letak cabang-cabang restoran cindelaras yang ada di Kota Medan.



Gambar 4.15 Interface Menu Home

2. Menu

Interface menu ini adalah tampilan untuk user yang ingin melihat menu makanan dan minuman yang disediakan serta harga masing-masing menu.



Daftar Menu

Makanan bukan hanya masakan yang akan berakhir di perutmu, melainkan sebuah pertulungan yang patut dinikmati dan dihargai
-Ayam Penyet Cindelaras-

Makanan	Minuman
 Nasi Bakar Nasi Bakar	Rp. 26,000
 Gurame Bakar Gurame Bakar	Rp. 60,000
 Nasi Bebek Penyet Nasi Bebek Penyet	Rp. 32,000
 Nasi Ayam Penyet Nasi Ayam Penyet	Rp. 23,000

RESERVASI SEKARANG

Ayam Penyet Cindelaras

Banua Ayam Penyet
CINDELARAS dimulai dari usaha usaha bahan keling yang diakuisisi oleh Bapak Sumarsano pada tahun 1994.

Latest Blog

 MAY 5, 2018
How to cook beef
Grilled with potato

 MAY 5, 2018

Instagram



Newsletter

A small river named Duduk flows by their place and supplies it with the necessary resources.

Gambar 4.16 *Interface* Menu

3. Reservasi

Menu reservasi ini adalah tampilan yang berisi keperluan-keperluan yang harus di lakukan user jika ingin melakukan reservasi.

Buat Reservasi

Restoran Tradisional Terbesar, Cita Rasa Khas Indonesia

Nama Lengkap Email

No Handphone Tanggal Reservasi mm/dd/yyyy

Waktu Jumlah Orang

Pilih Cabang Store

BOOKING

Ayam Penyet Cindelaras

Bisnis Ayam Penyet CINDELARAS dimulai dari usaha usaha bakso keliling yang dilakukan oleh Bapak Sumarsono pada tahun 1994. Seiring waktu berjalan, Bapak Sumarsono berusaha untuk mengembangkan impianinya.

Latest Blog

- MAY 5, 2018 How to cook beef Grilled with potato
- MAY 5, 2018 A Japanese Master Chef
- MAY 5, 2018

Instagram

Newsletter

A small river named Duden flows by their place and supplies it with the necessary rogalialia

Enter Email address

Gambar 4.17 Interface Reservasi

4. Daftar Menu

Interface daftar menu ini adalah langkah selanjutnya setelah user melakukan reservasi maka user harus memilih menu makanan dan minuman.

Daftar Menu

Makanan bukan hanya masakan yang akan berakhir di perutmu, melainkan sebuah pertualangan yang perut dinikmati dan dihargai
-Ayam Penyet Cindelaras-

Makanan Minuman

	Tempe Tahu Penyet Tempe Tahu Penyet	Rp. 15,000
	Nasi Nila Bakar + Tumis Kangkung Nasi Nila Bakar + Tumis Kangkung	Rp. 30,000
	Ayam Bakar Ayam Bakar	Rp. 25,000
	Nasi Goreng Ori Nasi Goreng Ori	Rp. 15,000

Even the all-powerful Pointing has no control about the blind texts it is on almost
SELANJUTNYA

Ayam Penyet Cindelaras
Bantuan Ayam Penyet CINDELARAS dimulai dari usaha usaha bapak keliling yang dilakukan oleh Bapak

Latest Blog
 MAY 5, 2018
How to cook beef Grilled with potato

Instagram

Newsletter
A small river named Duden flows by their place and supplies it with the necessary irrigation

Gambar 4.18 Interface Daftar Menu

5. Detail Reservasi

Menu tampilan detail reservasi ini menampilkan data diri user dan menu pesanan user serta cara membayar DP.

#	Nama Menu	Jumlah	Harga	Sub Total
1	Es Teler CL	2	Rp. 15,000	Rp. 30,000
2	Tempe Tahu Penyet	2	Rp. 15,000	Rp. 30,000
			Total	Rp. 60,000

Data Pribadi

Nama	Cuby
Email	aluhan@rocketmail.com
Hp	08128901417

Data Reservasi

Biaya Reservasi	Rp. 50,000
Waktu	31 Jan 2020, 13:30
Jumlah	4 Orang

• Reservasi akan dikenakan biaya sebesar Rp. 50,000 sebagai pembayaran DP.
 • Silahkan transfer pembayaran DP tersebut ke Rek. 5300011434531 a/n Gebrina Rukta.
 • Konfirmasi pembayaran Anda di [sini](#).
 • Reservasi Anda akan kami batalkan jika dalam waktu 2 jam tidak konfirmasi pembayaran DP.
 • Total pesanan akan di kurangi dengan biaya reservasi saat Anda membayar di kasir.

[SELESAI](#) [DOWNLOAD](#)

Ayam Penyet Cindelaras
Banyak Ayam Penyet CINDELARAS dimulai dari usaha usaha baba keliling yang dilakukan oleh Bapak

Latest Blog
 MAY 5, 2018 [How to cook beef Grilled with potato](#)

Instagram
 

Newsletter
A small river named Duden flows by their place and supplies it with the necessary irrigation

Gambar 4.19 Interface Detail Reservasi

6. Konfirmasi Pembayaran

Menu tampilan konfirmasi pembayaran ini menampilkan kolom-kolom yang diisi oleh user untuk melakukan verifikasi pembayaran DP.

Konfirmasi Pembayaran

Please use this self-photograph for verifying your payment about this order
Please it on an off-white surface for better result

Nama Lengkap	Email
Puspita	puspita@gmail.com
Nama Panggilan	No. Rekening Panggilan
Nenek Panggilan	No. Rekening Panggilan
Jumlah Transfer	Bukti Transfer
50000	<input type="button" value="Choose File"/> No file chosen

KONFIRMASI

Ayam Penyet Cindelaras

masak Ayam Penyet
CINDELARAS dimulai dari
sebuah warung bahan basah yang dibuat oleh ibu Bapak
Sumarmi dan suami turun 3004
dariing warung berjalan, dipilih
Sumarmi berusaha untuk
mengembangkan
impian ayam Sumarmi saat
di banting dapur Sumarmi
masak Ayam Penyet

Latest Blog

 MAY 5, 2018 How to cook beef Grilled with potato
 MAY 5, 2018 A Japanese Master Chef
 MAY 5, 2018 Special Recipe for this month

Instagram



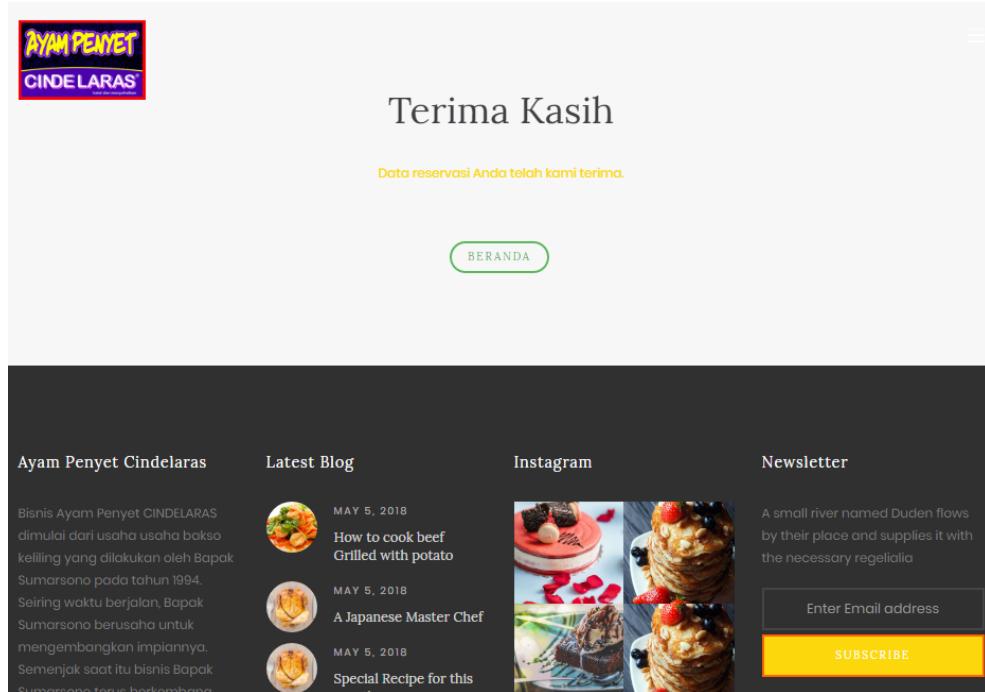
Newsletter

A small text: 'A small meal named Ondek Ondek
flew by their place and
supplied it with the
necessary vegetables.'

Gambar 4.20 Interface Konfirmasi Pembayaran

7. Selesai Reservasi

Tampilan ini memunculkan kata ‘terima kasih’ yang artinya telah selesainya kegiatan reservasi oleh user pada aplikasi.



Gambar 4.21 Interface Selesai Reservasi

8. Login Admin

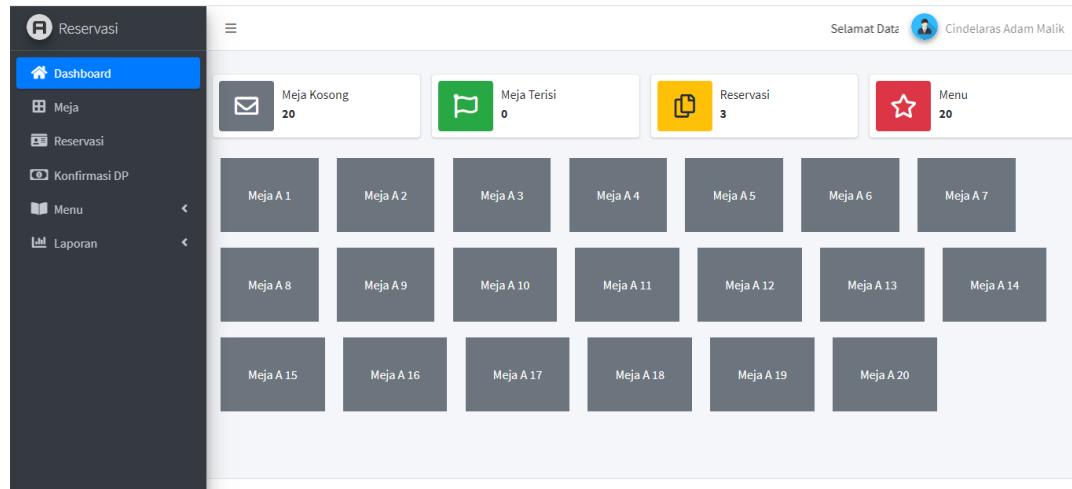
Tampilan menu login admin menampilkan kolom user dan password agar dapat masuk kedalam sistem reservasi admin.

 A screenshot of an 'Admin Login' form. The form is titled 'Admin Login' and includes fields for 'E-Mail Address' (containing 'cabang1@example.com'), 'Password' (containing '*****'), and a 'Remember Me' checkbox. Below the form are 'Login' and 'Forgot Your Password?' buttons. The top left corner shows 'Laravel Admin' and the top right corner shows 'Admin Login'.

Gambar 4.22 Interface Login Admin

9. Dashboard

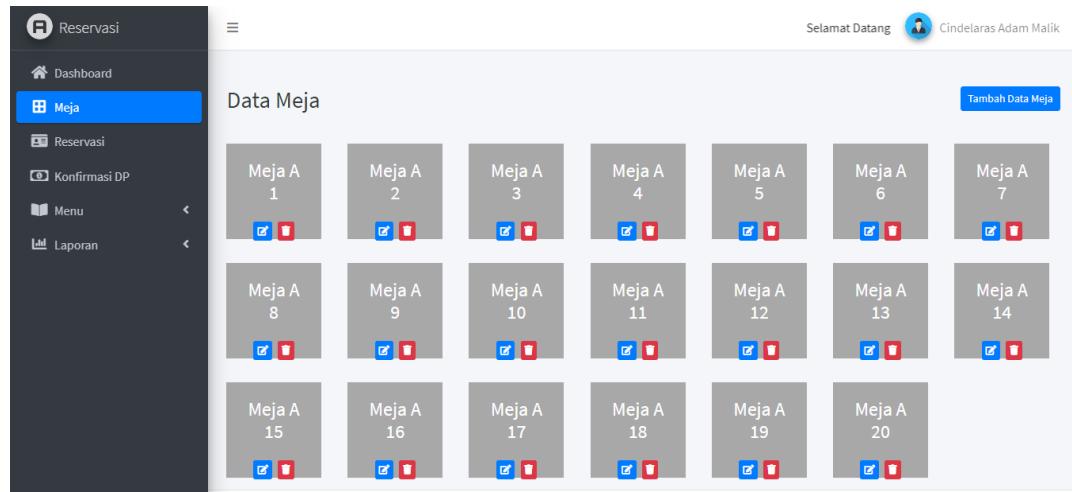
Tampilan menu dashboard menampilkan data meja yang kosong atau terisi.



Gambar 4.23 Interface Dashboard

10. Data Meja

Tampilan menu data meja menampilkan data meja sehingga admin bisa menambah meja atau menghapus meja.



Gambar 4.24 Interface Data Meja

11. Data Reservasi

Tampilan menu data reservasi ialah menampilkan data user yang melakukan reservasi sehingga admin bisa mengatur meja untuk user.

NO	Nama	Email	Handphone	Waktu Reservasi	Jumlah	Status DP	Aksi
1	sauki	qy4215@gmail.com	082302020202	08-Feb-2020, 15:00	4 Org	Terkonfirmasi	Pilih Meja Hapus
2	dwi	s.luhan@rocketmail.com	082294121670	08-Feb-2020, 16:00	2 Org	Selesai	Pilih Meja Hapus
3	Halimah	s.luhan@rocketmail.com	082245453213	15-Feb-2020, 15:00	4 Org	Selesai	Pilih Meja Hapus

Gambar 4.25 Interface Data Reservasi

12. Konfirmasi DP

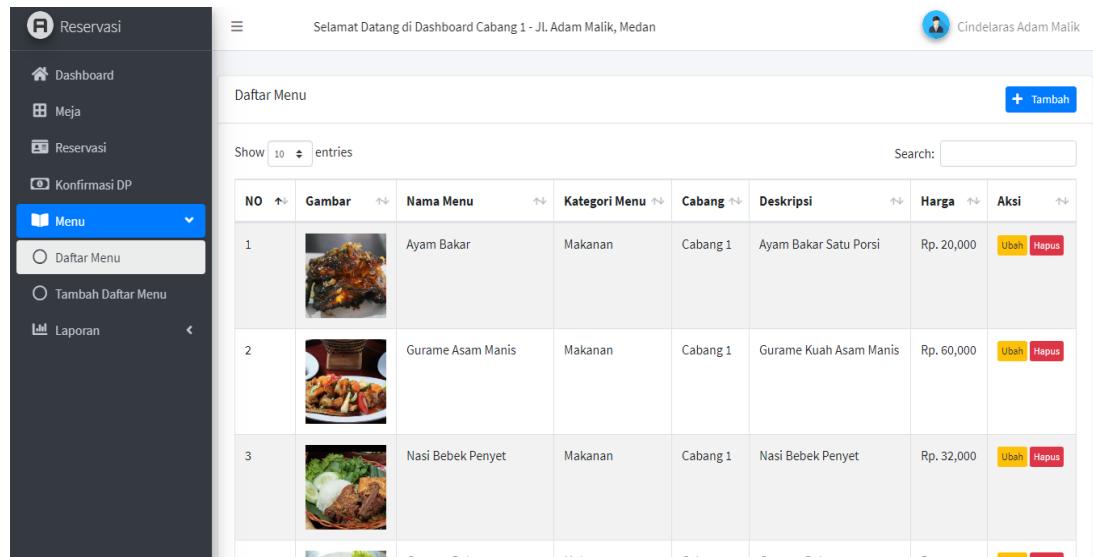
Tampilan menu konfirmasi DP ialah sistem menampilkan user yang telah melakukan pembayaran yang selanjutnya akan di cek oleh admin.

NO	Nama Reservasi	Nama Pengirim	NoRek Pengirim	Jumlah Transfer	Status	Bukti Transfer	Aksi
1	dwi	dwi	000006788532	Rp. 35,500	Terkonfirmasi	5e3948aa3345c.jpg	Hapus
2	Halimah	Halimah	787999766678000	Rp. 48,000	Terkonfirmasi	5e393f02a17dd.jpg	Hapus
3	sauki	sauki	00212123346790	Rp. 52,000	Terkonfirmasi	5e393c2f5b198.jpg	Hapus

Gambar 4.26 Interface Konfirmasi DP

12. Daftar Menu Admin

Tampilan daftar menu admin ialah sistem menampilkan daftar menu makanan dan minuman khusus admin.

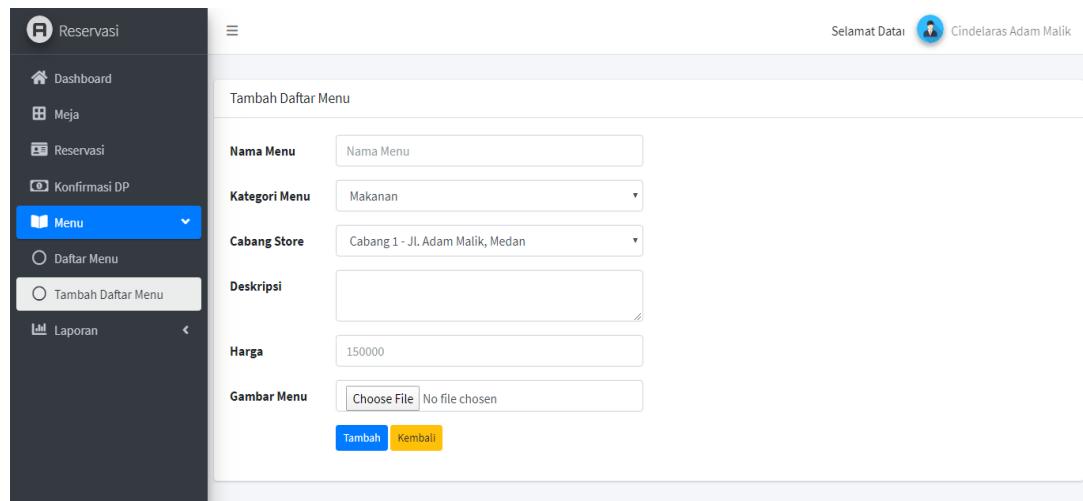


NO	Gambar	Nama Menu	Kategori Menu	Cabang	Deskripsi	Harga	Aksi
1		Ayam Bakar	Makanan	Cabang 1	Ayam Bakar Satu Porsi	Rp. 20,000	<button>Ubah</button> <button>Hapus</button>
2		Gurame Asam Manis	Makanan	Cabang 1	Gurame Kuah Asam Manis	Rp. 60,000	<button>Ubah</button> <button>Hapus</button>
3		Nasi Bebek Penyet	Makanan	Cabang 1	Nasi Bebek Penyet	Rp. 32,000	<button>Ubah</button> <button>Hapus</button>

Gambar 4.27 Interface Daftar Menu Admin

13. Tambah Daftar Menu

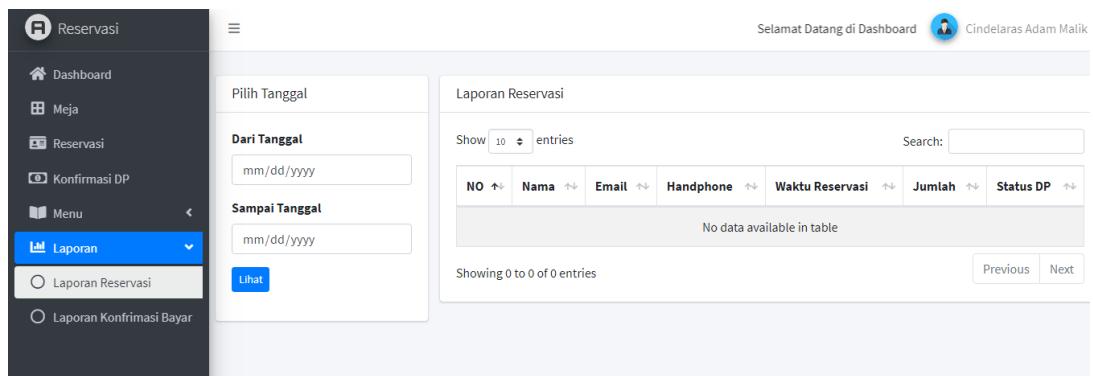
Tampilan tambah daftar menu ialah menampilkan halaman untuk menambahkan menu makanan dan minuman pada cabang cindelaras, ini dilakukan oleh admin.



Gambar 4.28 Interface Tambah Daftar Menu

14. Laporan

Tampilan laporan ialah menampilkan halaman yang digunakan oleh admin cabang untuk melihat data laporan yaitu riwayat reservasi perminggu atau perbulan dari para pelanggan.



Gambar 4.29 Interface Laporan

4.9.2 Blackbox Testing

Blackbox testing adalah pengujian terakhir dengan menjalankan program dan memastikan tidak ada error di dalam program tersebut dan program tersebut telah sesuai harapan.

Tabel 4.10 Blackbox Testing

No.	RancanganProses	Hasil diharapkan	Hasil Pengujian	Kesimpulan
1	Membuka website reservasi	Menampilkan <i>form</i> home	Sesuai	Valid
2	Melihat Menu	Menampilkan <i>form</i> menu	Sesuai	Valid
3	Pilih tombol reservasi	Menampilkan <i>form</i> reservasi	Sesuai	Valid
4	Pilih tombol selanjutnya	Menampilkan <i>form</i> daftar menu	Sesuai	Valid
5	Pilih tombol selanjutnya	Menampilkan <i>form</i> detail reservasi	Sesuai	Valid

No.	Rancangan Proses	Hasil diharapkan	Hasil Pengujian	Kesimpulan
6	Pilih tombol disini	Menampilkan <i>form</i> konfirmasi pembayaran	Sesuai	Valid
7	Pilih tombol selesai	Menampilkan <i>form</i> selesai reservasi	Sesuai	Valid
8	Membuka home admin	Menampilkan <i>form</i> home admin	Sesuai	Valid
9	Masukkan username dan password admin	Menampilkan <i>form</i> dashboard	Sesuai	Valid
10	Pilih menu meja	Menampilkan <i>form</i> data meja	Sesuai	Valid
11	Pilih menu reservasi	Menampilkan <i>form</i> data reservasi	Sesuai	Valid
12	Pilih menu konfirmasi DP	Menampilkan <i>form</i> konfirmasi DP	Sesuai	Valid
13	Pilih daftar menu	Menampilkan <i>form</i> daftar menu	Sesuai	Valid
14	Pilih tambah daftar menu	Menampilkan <i>form</i> tambah daftar menu	Sesuai	Valid
15	Pilih menu cabang store	Menampilkan <i>form</i> laporan	Sesuai	Valid
16	Pilih menu keluar	Kembali ke <i>form</i> awal admin	Sesuai	Valid

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan seluruh penelitian yang di dapatkan untuk membangun aplikasi ini, dapat diambil kesimpulan sebagai berikut:

1. Sistem layanan pemesanan akhirnya dapat diterapkan pada Restoran Ayam Penyet Cindelaras di Kota Medan.
2. Adanya sistem ini dapat memudahkan penulis dan juga pembaca mengenai sistem informasi layanan pemesanan pada Restoran Cindelaras Kota Medan dengan menggunakan *framework* laravel.

5.2 Saran

1. Aplikasi ini masih berbasis web, diharapkan di masa yang akan datang aplikasi ini bisa berkembang menjadi berbasis *webmobile*.
2. Untuk penelitian selanjutnya sistem ini dapat didukung dengan pembayaran *scanbarcode* melalui *e-money* atau sejenisnya.

DAFTAR PUSTAKA

- Akbar, A. (2015). *Proposal Franchise Ayam Penyet Cindelaras*. SCRIBD. <https://www.scribd.com/doc/260930735/Proposal-Franchise-Ayam-Penyet-CINDELARAS#>
- Al Fatta, H. (2007). *Analisis dan Perancangan Sistem Informasi untuk Keunggulan Bersaing Perusahaan dan Organisasi Modern* (A. Triyuliana Heni (ed.)). ANDI.
- Aminudin. (2015). *Cara Efektif Belajar Framework LARAVEL* (L. Hakim (ed.)). Lokomedia.
- Anggraeni Yunaeti, E., & Irviani, R. (2017). *Pengantar Sistem Informasi* (E. Risanto (ed.)). ANDI.
- Anhar. (2010). *Panduan menguasai PHP & MySQL secara otodidak*. mediakita.
- Azhar, S. (2017). *Sistem Informasi Akutansi: Pemahaman Konsep Secara Terpadu*. Lingga Jaya.
- Djahir, Y., & Pratita, D. (2014). *Bahan Ajar Sistem Informasi Manajemen*. Deepublish.
- Hai, B., & Setiawan, S. H. (2019). *Aplikasi Absensi Dosen dengan Java dan Smartphone sebagai Barcode*. PT. Alex Media Komputindo.
- Harianto, K., Pratiwi, H., & Suhariyadi, Y. (2019). *SISTEM MONITORING LULUSAN PERGURUAN TINGGI DALAM MEMASUKI DUNIA KERJA MENGGUNAKAN TRACER STUDY* (Lutfiah (ed.)). Media Sahabat Cendekia.
- Hutahaean, J. (2014). *Konsep Sistem Informasi*. Deepublish.
- Irawan, D., & Aprillia, S. (2018). Implementasi E-Arsip Pada Program Studi Tekhnik Informatika. *Jurnal Teknologi Informasi*, 2.
- Kemendikbud. (2019). *Kamus Besar Bahasa Indonesia*. 2019.

<https://kbbi.kemendikbud.go.id>

- Kusrini & Koniyo, A. (2007). *Tuntutan Praktis Membangun Sistem Informasi Akuntasi dengan Visual Basic dan Microsoft SQL Server* (S. Suyantoro (ed.)). ANDI.
- Mandar, R. (2017). *Kitab Kumpulan Tips, Latihan, dan Soal Database*. Elex Media Komputindo.
- Maniah. (2017). *Analisis dan Perancangan Sistem Informasi: Pembahasan Secara Praktis dengan Contoh Kasus*. Deepublish.
- Muslihudin, M., & Oktafianto. (2016). *Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML*. ANDI.
- Nogi, H. (2005). *Manajemen Publik* (Y. Hardiwati (ed.)). Grasindo.
- Puji, Rulia, H., & dkk. (2015). Sistem Penjualan Berbasis Web (E-Commerce) Pada Distro Kabupaten Pacitan. *Bianglala Informatika*, 3.
- Puspitasari, A. H. (2011). *Pemograman Web Database dengan PHP & MySQL*. Skripta Media Creative.
- Putih, T. A. (2014). *Panduan Laravel PHP Framework*. Creative Commons.
- Rangkuti, F. (2017). *CUSTOMER CARE EXCELLENCE Meningkatkan Kinerja Perusahaan melalui Pelayanan Prima Plus Analisis Kasus Jasa Raharja* (Mahardika (ed.)). Gramedia.
- Rerung, R. R. (2018). *Pemograman Web Dasar*. Deepublish.
- S. Hut, W. (2010). *Menjadi Master PHP dengan Framework Codeigniter*. Elex Media Komputindo.
- S.ST, N. (2019). *Pembangunan Website dengan Yii2 Framework: Studi Kasus Sistem Perekaman Fenomena (SIPENA)*. LeutikaPrio.
- Samsudin, Putra, Y., & R. (2014). Perancangan Aplikasi text to speech pengenalan kalimat dalam bahasa inggris menggunakan metode linear

predictive coding. *KetIK*.

Siahaan, V., & Sianipar, R. H. (2019). *LANGKAH DEMI LANGKAH PEMROGRAMAN DATABASE ACCESS & SQL SERVER DENGAN VISUAL BASIC.NET*. SPARTA.

Suendri. (2018). Penerapan Konsep Model View Controller Pada Perancangan Sistem Manajemen Software Berbasis Web. *JISTech*, 3.

Sulistyo, N. (2017). *Analisis dan Perancangan Sistem Informasi*. Trans Tekno.

Sutabri, T. (2004). *Analisis Sistem Informasi*. ANDI.

Wahana Komputer. (2010). *Panduan Belajar MySQL Database Server*. mediakita.

Wiwoho, A. (2008). *Pengetahuan Tata Hidang* (A. M. Sofyan & Y. Sumiharti (eds.)). Erlangga.

Yuhefizard. (2008). *Database Management Menggunakan Microsoft Access 2003*. Elex Media Komputindo.

Yurindra. (2017). *Software Engineering*. Deepublish.

Zufria, I. (2013). Pemodelan Berbasis UML (Unified Modeling Language) dengan Strategi Teknik Orientasi Objek User Centered Design (UCD) dalam Sistem Administrasi Pendidikan. *Research Gate*.

LAMPIRAN I

SOURCE CODE

1. Admin

Controller Cabang

```
<?php

namespace
App\Http\Controllers\Admin;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use DB;

class CabangController extends Controller
{
    @return
    public function index()
    {
        $data = DB::table('cabangs')-
>orderBy('id', 'ASC')->get();

        returnview('admin.cabang.index'
, compact('data'));
    }

    @return
    public function create()
    {

    }

    @param
    @return

    public function store(Request
$request)
    {
        DB::table('cabangs')->insert([
            'cabang' => $request-
>cabang,
            'alamat' => $request->alamat
        ]);
    }

    @param
    @return

    public function show($id)
    {
    }

    @param
    @return

    public function edit($id)
    {
    }

    /**
     * Update the specified
     * resource in storage.
     *
     * @param
     \Illuminate\Http\Request
     $request
     * @param int $id
     * @return
     \Illuminate\Http\Response
     */
    public function
update(Request $request, $id)
    {
        DB::table('cabangs')-
>where('id', $id)->update([
            'cabang' => $request-
>cabang,
            'alamat' => $request-
>alamat
        ]);

        return redirect()->back()->with('sukses',
            'Berhasil
menambahkan
Cabang
Store.');
    }
}
```

```

'Berhasil mengubah Cabang
Store.');
}

/**
 * Remove the specified
resource from storage.
*
* @param int $id
* @return
 Illuminate\Http\Response
*/
public function
destroy($id)
{
    DB::table('cabangs')-
>where('id', $id)->delete();

    return redirect()->back()->with('sukses',
'Berhasil dihapus.');
}
}

```

Controller Download Reservasi

```

<?php

namespace
App\Http\Controllers\Admin;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use DB;
use PDF;

class DownloadReservasiController
extends Controller
{

```

```

public function download($token)
{
    $data
    DB::table('reservasi_menus')-
->join('reservasis',
'reservasi_menus.reservasi_id', '=',
'reservasis.id')

->join('menus',
'reservasi_menus.menu_id', '=',
'menus.id')

->where('reservasis.reservasi_token',
$token)

->get();
}

$pdf
PDF::loadView('download.reservasi',
compact('data'));

```

```

return $pdf-
>download('invoice.pdf');

// return
view('download.reservasi',
compact('data'));

}
}

```

Controller Kategori Menu

```

<?php

namespace
App\Http\Controllers\Admin;
```

```

        }

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use DB;

class KategoriMenuController
extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $this->middleware('auth:admin');
        $this->middleware('role:super');
    }

    public function index()
    {
        $data =
DB::table('kategori_menus')->get();

        return view('admin.menu-kategori.index', compact('data'));
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        //
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        DB::table('kategori_menus')-
>insert([
            'nama_kategori' => $request-
>nama
        ]);
    }
}

```

```

    ]);

    // }

    return redirect()->back()->with('sukses', 'Berhasil menambahkan kategori menu.');
}

/**
 * Update the specified resource in storage.
 *
 * @param Illuminate\Http\Request $request
 * @param int $id
 * @return Illuminate\Http\Response
 */
public function update(Request $request, $id)
{
    DB::table('kategori_menus')
        ->where('id', $id)
        ->update([
            'nama_kategori' => $request->nama
        ]);

    return redirect()->back()->with('sukses', 'Berhasil mengubah kategori menu');
}

}

public function show($id)
{
    //
}

/**
 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return Illuminate\Http\Response
 */
public function edit($id)
{
}

```

```

* Remove the specified resource
from storage.

*
* @param int $id
* @return \Illuminate\Http\Response
*/
public function destroy($id)
{
    DB::table('kategori_menus')
        ->where('id', $id)
        ->delete();

    return redirect()->back()
->with('sukses', 'Berhasil menghapus
kategori menu.');
}

Controller Konfirmasi Bayar
<?php

namespace
App\Http\Controllers\Admin;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use DB;

class KonfirmasiBayarController
extends Controller

{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        if (auth('admin')->user()->cabang_id > 0) {
            $data = DB::table('konfirmasi_bayars')
                ->join('reservasis',
'konfirmasi_bayars.reservasi_id', '=',
'reservasis.id')
                ->orderBy('konfirmasi_bayars.id',
'DESC')
                ->where('reservasis.cabang_id', '=',
auth('admin')->user()->cabang_id)
                ->get();
        } else {
            $data = DB::table('konfirmasi_bayars')
                ->join('reservasis',
'konfirmasi_bayars.reservasi_id', '=',
'reservasis.id')
        }
    }
}

```

```

        /*
>orderBy('konfirmasi_bayars.id',
'DESC')
->get();
}

return view('admin.konfirmasi-
bayar.index', compact('data'));

}

/**
 * Show the form for creating a
new resource.
*
* @return \Illuminate\Http\Response
*/
public function create()
{
//
}

/**
 * Store a newly created resource
in storage.
*
* @param \Illuminate\Http\Request $request
* @return \Illuminate\Http\Response
*/
public function store(Request
$request)
{
//
}

}

/**
 * Display the specified resource.
*
* @param int $id
* @return \Illuminate\Http\Response
*/
public function show($id)
{
//
}

/**
 * Show the form for editing the
specified resource.
*
* @param int $id
* @return \Illuminate\Http\Response
*/
public function edit($id)
{
}

```

```

//



}

return redirect()->back()->with('sukses', 'Berhasil dikonfirmasi.');
```

}

```

/**



 * Remove the specified resource from storage.

 *

 * @param int $id

 * @return Illuminate\Http\Response

 */

public function destroy($id)

{

    $data = DB::table('konfirmasi_bayars')->where('reservasi_id', $id)->first();

    $unlink = unlink(public_path('bukti-transfer/'.$data->bukti_transfer));

    if ($unlink) {

        DB::table('konfirmasi_bayars')->where('reservasi_id', $id)->delete();

    }

}

```

```

* Update the specified resource in storage.

 *

 * @param Illuminate\Http\Request $request

 * @param int $id

 * @return Illuminate\Http\Response

 */

public function update(Request $request, $id)

{

    $status_transfer = DB::table('konfirmasi_bayars')->where('reservasi_id', $id)->update(['status_transfer' => 'Terkonfirmasi']);

    if ($status_transfer) {

        DB::table('reservasis')->where('id', $id)->update(['status' => 'Terkonfirmasi']);

    }

}

```

```

$from = $req->from;
$to = $req->to;
if (auth('admin')->user()->cabang_id > 0) {
    $data = DB::table('konfirmasi_bayars as kb')
        ->join('reservasis as r',
            'kb.reservasi_id', '=', 'r.id' )
        ->whereBetween('kb.created_at',
            [$from, $to])
        ->where('r.cabang_id',
            '=', auth('admin')->user()->cabang_id)
        ->get();
} else {
    $data = DB::table('konfirmasi_bayars as kb')
        ->join('reservasis as r',
            'kb.reservasi_id', '=', 'r.id' )
        ->whereBetween('kb.created_at',
            [$from, $to])
        ->get();
}
return redirect()->back()->with(['data' => $data, 'from' => $from, 'to' => $to]);
}

```

```

public function download($from,
$to)
{
    if (auth('admin')->user()->cabang_id > 0) {
        $data = DB::table('konfirmasi_bayars as kb')
            ->join('reservasis as r',
'kb.reservasi_id', '=', 'r.id' )
            -
>whereBetween('kb.created_at',
[$from, $to])

        -
>where('kb.cabang_id', '=', auth('admin')->user()->cabang_id)
        ->get();
    }else{
        $data = DB::table('konfirmasi_bayars as kb')
            ->join('reservasis as r',
'kb.reservasi_id', '=', 'r.id' )
            -
>whereBetween('kb.created_at',
[$from, $to])
        ->get();
    }

    $pdf = PDF::loadView('download.laporan-konfirmasi-bayar', compact('data',
'from', 'to'));
}

return $pdf->download('Laporan_Reservasi.pdf');
}

}

Controller Laporan Reservasi
<?php

namespace App\Http\Controllers\Admin;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Reservasi;
use PDF;

class LaporanReservasiController extends Controller
{
    public function index()
    {
        return view('admin.laporan.reservasi');
    }

    public function result(Request $req)
    {
        $from = $req->from;

```

```

$to = $req->to;                                ->get();

        if      (auth('admin')->user()->cabang_id) {
            $data      = Reservasi::whereBetween('created_at', [$from, $to])->get();

            ->where('cabang_id', '=', auth('admin')->user()->cabang_id)
                ->get();
        } else {
            $data      = Reservasi::whereBetween('created_at', [$from, $to])->get();
        }

        return redirect()->back()->with(['data' => $data, 'from' => $from, 'to' => $to]);
    }

    public function download($from, $to)
    {
        if      (auth('admin')->user()->cabang_id) {
            $data      = Reservasi::whereBetween('created_at', [$from, $to])

            ->where('cabang_id', '=', auth('admin')->user()->cabang_id)
                ->get();
        } else {
            $data      = Reservasi::whereBetween('created_at', [$from, $to])->get();
        }

        $pdf      = PDF::loadView('download.laporan-reservasi', compact('data', 'from', 'to'));
        $pdf->download('Laporan_Reservasi.pdf');
    }
}

Controller Meja
<?php

namespace App\Http\Controllers\Admin;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use DB;

class MejaController extends Controller
{
    /**

```

```

* Display a listing of the resource.
*/
*
* @return \Illuminate\Http\Response
*/
public function index()
{
    if (auth('admin')->user()->cabang_id > 0) {
        $data = DB::table('mejas')
            ->where('cabang_id', '=', auth('admin')->user()->cabang_id)
            ->orderBy('id', 'ASC')
            ->get();
    } else {
        $data = DB::table('mejas')
            ->orderBy('id', 'ASC')->get();
    }

    return view('admin.meja.index', compact('data'));
}

/**
 * Show the form for creating a new resource.
 */
* @return \Illuminate\Http\Response
*/
public function create()
{
    /**
}
*/
/***
 * Store a newly created resource in storage.
*/
* @param \Illuminate\Http\Request $request
* @return \Illuminate\Http\Response
*/
public function store(Request $request)
{
    DB::table('mejas')->insert([
        'nama_meja' => $request->nama_meja,
        'cabang_id' => $request->cabang_id,
        'status' => 0
    ]);
}

return redirect()->back()->with('sukses', 'Data meja berhasil ditambah.');
}

```

```

        *

    /**
     * Display the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        //
    }

    /**
     * Show the form for editing the
     * specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function edit($id)
    {
        //
    }

    /**
     * Update the specified resource in
     * storage.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request
    $request, $id)
    {
        DB::table('mejas')->where('id',
        $id)->update([
            'nama_meja' => $request-
            >nama_meja,
            'cabang_id' => $request-
            >cabang_id,
            'status' => 0
        ]);

        return redirect()->back()->with('sukses', 'Data berhasil di
        ubah.');
    }
}

/**
 * Remove the specified resource
 * from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */

```

```

        */

    public function destroy($id)
    {
        DB::table('mejas')->where('id', $id)->delete();

        return redirect()->back()->with('sukses', 'Berhasil di hapus.');
    }
}

Controller Menu

<?php

namespace
App\Http\Controllers\Admin;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use DB;

class MenuController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        if (auth('admin')->user()->cabang_id > 0) {
            $data = DB::table('menus')
                ->join('cabangs',
                    'menus.cabang_id', '=', 'cabangs.id')
                ->where('menus.cabang_id', '=', auth('admin')->user()->cabang_id)
                ->select('menus.*',
                    'menus.id', 'cabangs.cabang')
                ->get();
        } else {
            $data = DB::table('menus')
                ->join('cabangs',
                    'menus.cabang_id', '=', 'cabangs.id')
                ->select('menus.*',
                    'menus.id', 'cabangs.cabang')
                ->get();
        }
        return view('admin.menu.index', compact('data'));
    }

    /**
     * Show the form for creating a new resource.
     *

```

```

    *
    @return
    \Illuminate\Http\Response
    */

    public function create()
    {
        $cabang = DB::table('cabangs')-
>get();

        return
        view('admin.menu.tambah',
        compact('cabang'));
    }

    /**
     * Store a newly created resource
     * in storage.
     *
     * @param
     \Illuminate\Http\Request $request
     *
     * @return
     \Illuminate\Http\Response
    */

    public function store(Request
$request)
    {
        request()->validate([
            'gambar'          =>
'required|image|mimes:jpeg,png,jpg|
max:2048',
        ]);

        $imageName = uniqid().'.'.
request()->gambar-
>getClientOriginalExtension();

        $move = request()->gambar-
>move(public_path('img-menu'),
$imageName);

        if ($move) {
            DB::table('menus')->insert([
                'cabang_id' => $request-
>cabang_id,
                'kategori'  => $request-
>kategori,
                'nama'      => $request->nama,
                'deskripsi' => $request-
>deskripsi,
                'harga'     => $request->harga,
                'gambar'    => $imageName
            ]);
        }

        return redirect()->back()-
>with('sukses', 'Berhasil menambah
menu.');
    }

    /**
     * Display the specified resource.
     *
     * @param int $id

```

```

    *
    @return      /**
    \Illuminate\Http\Response

    */

    public function show($id)
    {
        //
    }

    /**
     * Show the form for editing the
     * specified resource.
     *
     * @param int $id
     *
     * @return \Illuminate\Http\Response
     */
    public function edit($id)
    {
        $data = DB::table('menus')
            ->where('id', $id)
            ->first();

        $cabang = DB::table('cabangs')
            ->get();

        return view('admin.menu.ubah',
            compact('data', 'cabang'));
    }

    /**
     * Update the specified resource in
     * storage.
     *
     * @param \Illuminate\Http\Request $request
     * @param int $id
     *
     * @return \Illuminate\Http\Response
     */
    public function update(Request
    $request, $id)
    {
        $data = DB::table('menus')
            ->where('id', $id)
            ->first();

        request()->validate([
            'gambar'          =>
            'required|image|mimes:jpeg,png,jpg|
            max:2048',
        ]);

        $imageName          =
            uniqid().'.'request()->gambar-
            >getClientOriginalExtension();

        $move   =  request()->gambar-
            >move(public_path('img-menu'),
            $imageName);

        if ($move) {
    }
}

```

```

        unlink(public_path('img-
menu/'.$data->gambar));
        *
        @return
        \Illuminate\Http\Response
        */

        public function destroy($id)
        {
            $data = DB::table('menus')-
>where('id', $id)->first();
            $unlink =
            unlink(public_path('img-
menu/'.$data->gambar));
            if ($unlink) {
                DB::table('menus')
                ->where('id', $id)
                ->delete();
            }
            return redirect()->back()-
>with('sukses', 'Berhasil menghapus
data menu.');
        }
    }

    /**
     * Remove the specified resource
     * from storage.
     *
     * @param int $id
     */

```

```

namespace
App\Http\Controllers\Admin;

use Illuminate\Http\Request;

```

```

use App\Http\Controllers\Controller;
use DB;

class PembayaranController extends Controller
{
    public function index($id)
    {
        $data = DB::table('reservasi_menus')
            ->join('reservasasi',
            'reservasi_menus.reservasi_id', '=', 'reservasasi.id')
            ->join('menus',
            'reservasi_menus.menu_id', '=', 'menus.id')
            ->where('reservasasi.meja_id', $id)
            // -
            ->where('reservasasi.status', 'Terkonfirmasi')
            ->get();

        return view('admin.pembayaran.index',
            compact('data'));
    }

    public function bayar($id)
    {
        $reservasi = DB::table('reservasasi')
            ->where([
                'meja_id' => $id
                // 'status' =>
                'Terkonfirmasi'
            ])
            ->update([
                'status' => 'Selesai'
            ]);

        if ($reservasi) {
            DB::table('mejas')
            ->where('id', $id)->update([
                'status' => 0
            ]);
        }
    }
}

return redirect()
->route('admin.home')->with('sukses',
    'Sudah dibayar.');

}

Controller Reservasi
<?php

namespace
App\Http\Controllers\Admin;

use Illuminate\Http\Request;

```

```

use App\Http\Controllers\Controller;
use DB;

class ReservasiController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    //fungsi untuk menampilkan data
    //reservasi

    public function index()
    {
        if (auth('admin')->user()->cabang_id > 0) {
            $data =
                DB::table('reservas')
                    ->where('cabang_id', '=', auth('admin')->user()->cabang_id)
                    ->where('status', 0)
                    ->orderBy('nama_meja', 'ASC')
                    ->get();
        } else {
            $data =
                DB::table('reservas')
                    ->orderBy('tgl', 'ASC')
                    ->orderBy('waktu', 'ASC')
                    ->get();
        }

        $meja = DB::table('mejas')
            ->where('status', 0)
            ->orderBy('nama_meja', 'ASC')
            ->get();
    }

    return view('admin.reservasi.index',
        compact('data', 'meja'));
}

/**
 * Show the form for creating a new resource.
 */

```

```

    *
    *
    * @return \Illuminate\Http\Response
    */
public function create()
{
    //
}

/**
 * Store a newly created resource
 * in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
*/
public function store(Request
$request)
{
    //
}

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
*/
public function show($id)
{
    $data
    =
DB::table('reservasi_menus')
    ->join('reservasis',
'reservasi_menus.reservasi_id', '=',
'reservasis.id')
    ->join('menus',
'reservasi_menus.menu_id', '=',
'menus.id')
    ->where('reservasis.id', $id)
    ->get();

    return
view('admin.reservasi.detail',
compact('data'));
}

/**
 * Show the form for editing the
 * specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
*/

```

```

/*
                                     // 'status' => $request-
>status
public function edit($id)
{
    //
}

/*
 * Update the specified resource in
storage.
*
* @param
\Illuminate\Http\Request $request
* @param int $id
*
* @return
\Illuminate\Http\Response
*/
//fungsi pilih meja untuk reservasi
//dan mengubah status reservasi

public function update(Request
$request, $id)
{
    //pilih meja untuk reservasi
    $pilih_meja =
DB::table('reservas')
    ->where('id', $id)
    ->update([
        'meja_id' => $request-
>meja_id,
                                     // 'status' => $request-
>status
    ]);
}

/*
 * Mengubah status meja menjadi
berisi
if ($pilih_meja) {
    // DB::table('mejas')-
>where('status', 1)
    // ->update([
    // 'status' => 0
    // ]);
}

DB::table('mejas')-
>where('id', $request->meja_id)
->update([
    'status' => 1
]);
}

return redirect()->back()->with('sukses', 'Reservasi
Terkonfirmasi.');
}

/*
 * Remove the specified resource
from storage.
*
* @param int $id

```

```

*
@return
\Illuminate\Http\Response
*/
//fungsi untuk menghapus data
reservasi

public function destroy($id)
{
    //menghapus data makanan dan
    minuman sesuai reservasi

    $del_resev = DB::table('reservasi_menus')
    ->where('reservasi_id', $id)
    ->delete();

    //menghapus data reservasi jika
    makanan & minuman berhasil di
    hapus

    if ($del_resev) {
        DB::table('reservasis')
        ->where('reservasis.id', $id)
        ->delete();
    }

    return redirect()->back()
    ->with('sukses', 'Reservasi berhasil di
    hapus.');
}
}

```

2. User Controller Home

```

<?php

namespace
App\Http\Controllers\User;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use DB;

class HomeController extends Controller
{
    public function index()
    {
        $makanan = DB::table('menus')
        ->where('kategori',
        'Makanan')
        ->orderBy('id', 'desc')
        ->limit(6)

        $minuman = DB::table('menus')
        ->get();
    }
}
```

```

use App\Model\KonfirmasiBayar;
->where('kategori',
'Minuman')

->orderBy('id', 'desc')

->limit(6)

->get();

$cabang2 =
DB::table('cabangs')->orderBy('id',
'desc')->limit(1)->first();

$cabang1 =
DB::table('cabangs')->orderBy('id',
'asc')->limit(1)->first();

return
view('user.home.index',
compact('cabang1', 'cabang2',
'makanan', 'minuman'));

}

Controller          Konfirmasi
Pembayaran

<?php

namespace
App\Http\Controllers\User;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use DB;

use App\Model\KonfirmasiBayar;
class KonfirmasiBayarController
extends Controller
{
public function index($token)
{
    $data =
DB::table('reservasi_menus')
->join('reservasis','reservasi_menus.reservasi_id', '=', 'reservasis.id')
->join('menus','reservasi_menus.menu_id', '=', 'menus.id')
->where('reservasis.reservasi_token',$token)
->get();

if ($data->count() < 1) {
    return redirect()
->route('user.home');
} else {
    return
view('user.konfirmasi-bayar.index',
compact('data'));
}
}

```

```

public function store(Request $request, $id)
{
    $request->validate([
        'bukti_transfer' => 'required|image|mimes:jpeg,png,jpg|max:2048',
    ]);

    $imageName = uniqid().'.'.$request->bukti_transfer->getClientOriginalExtension();

    $move = request()->bukti_transfer->move(public_path('bukti-transfer'), $imageName);

    if ($move) {
        KonfirmasiBayar::create([
            'reservasi_id' => $id,
            'nama_pengirim' => $request->nama_pengirim,
            'norek_pengirim' => $request->norek_pengirim,
            'jumlah_transfer' => $request->jumlah_transfer,
            'bukti_transfer' => $imageName,
            'status_transfer' => 'Pending'
        ]);
    }
}

return redirect()->back()->with('sukses', 'Konfirmasi pembayaran Anda sudah kami terima. Silahkan datang ke Restaurant kami sesuai jadwal reservasi Anda.');
}

}

Controller Menu
<?php

namespace App\Http\Controllers\User;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use DB;

class MenuController extends Controller
{
    public function index()
    {
        $makanan = DB::table('menus')
            ->where('kategori', 'Makanan')
    }
}

```

```

->orderBy('id', 'desc')
->limit(6)
->get();

$minuman = DB::table('menus')
->where('kategori',
'Minuman')
->orderBy('id', 'desc')
->limit(6)
->get();

return view('user.menu.cabang',
compact('makancabang',
'minumcabang', 'id'));
}

public function menuCabang($id)
{
    $makancabang = DB::table('menus')
->where('kategori',
'Makanan')
->where('cabang_id',
$id)
->orderBy('id', 'desc')
->limit(6)
->get();

$minumcabang = DB::table('menus')
->where('kategori',
'Minuman')
->orderBy('id', 'desc')
->limit(6)
->get();

return view('user.menu.index',
compact('makanan', 'minuman'));
}

Controller Reservasi
<?php

namespace
App\Http\Controllers\User;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Response;
use App\Reservasi;
use DB;

class ReservasiController extends Controller
{
    /**

```

```

* Display a listing of the resource.
*
* @return
\Illuminate\Http\Response
*/
public function index(Request
$request)
{
    $cabang_id = $request->ca;
    $cabang = DB::table('cabangs')-
>get();
    return
        view('user.reservasi.index',
compact('cabang', 'cabang_id'));
}

/**
* Show the form for creating a
new resource.
*
* @return
\Illuminate\Http\Response
*/
public function create()
{
    //
}

/**
* Store a newly created resource
in storage.
*
* @param
\Illuminate\Http\Request $request
* @return
\Illuminate\Http\Response
*/
public function store(Request
$request)
{
    $addReservasi = Reservasi::create([
        'meja_id' => 0,
        'nama_lengkap' => $request-
>nama_lengkap,
        'email' =>
$request->email,
        'hp' =>
$request->hp,
        'tgl' =>
$request->tgl,
        'waktu' =>
$request->waktu,
        'person' =>
$request->person,
        'status' =>
'Pending',
        'receved' => 0,
        'cabang_id' =>
$request->cabang,
    ]);
}

```

```

        // 
'reservasi_token' => str_random('10')
    }
    ]);

    /**
     * Show the form for editing the
     * specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
public function edit($id)
{
    // 
}

return redirect()->route('reservasi.menu.index',
$reservasi_token);
}

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function show($id)
{

```

```

        //
        $menu_id      =  $request-
        >menu_id;
        $harga = $request->harga;
        $jlh = $request->jlh;
        $total_harga = $harga * $jlh;

        $data = [
            'reservasi_id'      =>
            $reservasi_id,
            'menu_id' => $menu_id,
            'jumlah' => $jlh,
            'total_harga' => $total_harga
        ];

        $add = DB::table('reservasi_menus')-
        >insert($data);

        //      return      response()->json(['sukses' => 'Menu berhasil di tambah.']);
    }

    if ($add) {
        $menu = DB::table('menus')-
        >where('id',
        $menu_id)
        ->first();
    }
}

public function reservasiMenu(Request $request)
{
    $reservasi = DB::table('reservasis')-
    >where('reservasi_token',  $request-
    >reservasi_token)
    ->where('status',
    'Pending')
    ->first();
    $reservasi_id = $reservasi->id;
}

```

```

        //      return      redirect()->route('reservasi.detail', $reservasi->reservasi_token);

        return      redirect()->back()->with('sukses', $menu->nama);

    }

    public      function reservasiMenuItem($token)
    {
        $reservasi      = DB::table('reservasisis')
        ->where('reservasi_token', $token)
        ->where('status', 'Pending')
        ->get();
    }

    if      ($reservasi->first()->cabang_id > 0) {
        $makanan      = DB::table('menus')
        ->where('kategori', 'Makanan')
        ->where('cabang_id', $reservasi->first()->cabang_id)
        ->orderBy('id', 'desc')
        ->get();
    }

    $minuman      = DB::table('menus')
    ->where('kategori', 'Minuman')
    ->where('cabang_id', $reservasi->first()->cabang_id)
    ->orderBy('id', 'desc')
    ->get();

    if ($reservasi->count() < 1) {
        return      redirect()->route('user.home');
    }

```

```

}else{
    return
    view('user.menu.index',
    compact('makanan',      'minuman',
    'token'));
}

public function reservasiDetail($token)
{
    $reservasi =
    DB::table('reservasis')
    ->where('reservasi_token', $token)
    ->where('status',
    'Pending')
    ->get();

    $data =
    DB::table('reservasi_menus')
    ->join('reservasis',
    'reservasi_menus.reservasi_id',    '=',
    'reservasis.id')
    ->join('menus',
    'reservasi_menus.menu_id',         '=',
    'menus.id')

    ->where('reservasis.reservasi_token',
    $token)
    ->get();
}

if ($reservasi->count() < 1) {
    return
    redirect()->route('user.home');
}else{
    return
    view('user.reservasi.detail',
    compact('data'));
}

}

public function thank($token)
{
    $reservasi =
    DB::table('reservasis')
    ->where('reservasi_token', $token)
    ->where('status',
    'Pending')
    ->get();

    if ($reservasi->count() < 1) {
        return
        redirect()->route('user.home');
    }else{
        return
        view('user.thank.index');
    }
}
}

```

LAMPIRAN II

DAFTAR PERTANYAAN WAWANCARA

Daftar pertanyaan wawancara ini berfungsi untuk menjawab rumusan masalah pada penelitian yang berjudul “**Sistem Informasi Layanan Pemesanan Pada Restoran Cindelaras Kota Medan Dengan Menggunakan Laravel**”. Berikut daftar pertanyaan wawancara untuk menjawab rumusan masalah implementasi sistem informasi layanan reservasi pada restoran cindelaras.

Daftar Pertanyaan:

1. Bagaimana sistem booking di restoran cindelaras cabang adam malik / cabang marelhan sebelumnya?
2. Apa-apa saja menu makanan dan minuman yang ada di restoran cindelaras cabang adam malik / cabang marelhan, apakah ada perbedaan menu antar cabang cindelaras ataukah sama?
3. Bagaimana sistem manajemen atau struktur restoran cindelaras cabang adam malik / cabang marelhan?
4. Apakah tersedia diskon kepada konsumen yang sering melakukan reservasi?
5. Bagaimana stok persediaan makanan terutama dalam masa-masa pada reservasi?
6. Apakah sudah tersedia pembayaran melalui rekening pada restoran cindelaras cabang adam malik / cabang marelhan?
7. Berapa lama waktu tenggang pembayaran reservasi restoran cindelaras cabang adam malik / cabang marelhan?

Jawaban:

1. Di kedua cabang cindelaras sistem booking masih bersifat manual, yaitu dengan mendatangi restoran cindelaras, menu juga harus dipesan bersamaan dengan meja/tempat.
2. Menu yang tersedia beragam dan menu utama ialah nasi ayam penyet cindelaras, terdapat perbedaan menu antara cabang adm malik dan cabang marelhan contohnya di cindelaras marelhan terdapat menu *steak*, sedangkan cabang adam malik tidak tersedia menu tersebut.

3. Di setiap cabang terdapat penanggung jawab cabang, dan anggotanya sesuai dengan posisinya masing-masing.
4. Diskon telah diterapkan di kedua cabang, jika pelanggan setia melakukan reservasi di cindelaras.
5. Di kedua cabang stok makanan selalu tersedia bagi para pelanggan reservasi.
6. Di kedua cabang cindelaras, belum terdapat adanya rekening restoran, sehingga masih melakukan transaksi tunai.
7. Dikarenakan transaksi manual, maka tidak ada nya waktu tenggan pada kedua cabang cindelaras.

LAMPIRAN III

LAPORAN DATA PENELITIAN CINDELARAS











SPESIAL PAKET CINDELARAS

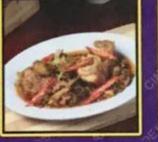
			
Nasi Ayam Penyet CL Rp. 24.000	Nasi Nila Penyet CL Rp. 30.000	Nasi Bebek Penyet CL Rp. 31.000	Nasi Lele Penyet CL Rp. 20.000
AYAM PENSET Rp. 18.000	NILA PENSET Rp. 25.000	BEBEK PENSET Rp. 28.000	LELE PENSET Rp. 15.000

MARELAN

BEBAKARAN

				
Ayam Bakar Rp. 18.000	Nila Bakar Rp. 27.000	Lele Bakar Rp. 17.000	Gurami Bakar / Goreng Rp. 60.000	Bebek Bakar Rp. 29.000

STEAK

				
CHICKEN STEAK Rp. 18.000	TEMPE STEAK Rp. 13.000	CUMI STEAK Rp. 25.000	OSENG - OSENG AYAM Rp. 20.000	OSENG - OSENG BEBEK Rp. 30.000

OSENG - OSENG

NASI

	
Nasi Putih Rp. 6.000	Nasi Bakul Rp. 9.000

PAKET MERAH

	
Nasi Lele Penyet + Manis Dingin Rp. 20.000	Nasi Lele Penyet + Jus Jeruk Rp. 25.000

MENU BARU

Belut Tauco Rp. 30.000
Belut Sambal Rp. 30.000
Belut Goreng Rp. 30.000
Udang Sambal Rp. 28.000
Udang Tauco Rp. 28.000
Udang Goreng Rp. 28.000

