

**Pengembangan Sistem Informasi Pemesanan Daily Catering
Senjani Kitchen berbasis *Progressive Web App* dengan
Metode *Waterfall***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Baariq Fairuuz Azhar

NIM: 175150401111030



**PROGRAM STUDI SISTEM INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2021**



PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI PEMESANAN DAILY CATERING SENJANI
KITCHEN BERBASIS *PROGRESSIVE WEB APP* DENGAN METODE *WATERFALL*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Baariq Fairuuz Azhar

NIM: 175150401111030

Skripsi ini telah diuji dan dinyatakan lulus pada
9 Juli 2021

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Buce Trias Harngara, S.Kom., M.Kom.

NIP. 198904262019031009

Dosen Pembimbing II

Bondan Sapta Prakoso, S.T., M.Kom.

NIP. 197408072003121002

Mengetahui

Ketua Jurusan Sistem Informasi



Issa Arwani, S.Kom., M.Sc.

NIP. 198309222012121003

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsurunsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 9 Juli 2021



Baariq Fairuuz Azhar

NIM: 175150401111030



ABSTRAK

Baariq Fairuuz Azhar, Pengembangan Sistem Informasi Pemesanan Daily Catering Senjani Kitchen berbasis *Progressive Web App* dengan Metode *Waterfall*.

Pembimbing: Buce Trias Hanggara, S.Kom., M.Kom. dan Bondan Sapta Prakoso, S.T., M.Kom.

Senjani Kitchen adalah perusahaan rintisan yang bergerak dibidang pangan di Kota Malang. Daily Catering Senjani Kitchen menggunakan sistem kupon, sehingga pelanggan harus membeli kupon dan menukarkan kupon tersebut dengan makanan sesuai jadwal. Kondisi saat ini pemesanan masih melalui aplikasi *chat* Whatsapp menggunakan *smartphone*, sehingga pelanggan terlalu bergantung kepada balasan admin. Penelitian ini bertujuan untuk mengembangkan sistem informasi yang bertujuan agar pelanggan lebih mudah dalam memesan makanan dan pencatatan yang lebih baik agar meminimalisir kerugian perusahaan. *Progressive web app* adalah sebuah aplikasi yang dibangun dengan melakukan optimasi pada sebuah *website*, *PWA* digunakan agar pelanggan mendapatkan pengalaman yang lebih nyaman. Dalam pengembangan ini menggunakan metode *waterfall* sebagai *system development life cycle* (SDLC). Tahap definisi kebutuhan menghasilkan 2 proses bisnis utama, 2 aktor yaitu pelanggan dan admin, dan 18 *use case* yang digambarkan dalam *use case diagram* dan *use case spesification*. Pada tahap perancangan sistem menghasilkan 18 *activity diagram*, 18 *sequence diagram*, *class diagram*, *entity relationship diagram*, dan 26 desain antarmuka. Pada tahap implementasi menggunakan konsep *Model View Controller* (MVC), React Js sebagai *View* dan CodeIgniter sebagai *Controller* dan *Model*. Pada tahap pengujian, dalam pengujian validasi 31 skenario uji berjalan sesuai harapan dan nilai *usability* 82.75 (*adjective rating excellent*) untuk sisi pelanggan dan nilai 72.5 (*adjective rating good*) untuk sisi admin melalui *system usability scale*.

Kata Kunci: sistem informasi, manajemen pemesanan, *waterfall*, *validation testing*, *system usability scale*, *progressive web app*

ABSTRACT

Baariq Fairuuz Azhar, Development of the Daily Catering Senjani Kitchen Ordering Information System based on the Progressive Web App with the Waterfall Method.

Advisors: Buce Trias Hanggara, S.Kom., M.Kom. dan Bondan Sapta Prakoso, S.T., M.Kom.

Senjani Kitchen is a start-up company engaged in food in the city of Malang. Daily Catering Senjani Kitchen uses a coupon system, so customers must buy coupons and exchange the coupons for food according to a schedule. The current condition is that orders are still through the Whatsapp chat application using a smartphone, so customers are too dependent on admin replies. This study aims to develop an information system that aims to make it easier for customers to order food and better record keeping to minimize company losses. Progressive Web App is an application built by optimizing a website, PWA is used so that customers get a more comfortable experience. This development using the waterfall method as a system development life cycle (SDLC). The requirements definition stage produces 2 main business processes, 2 actors, namely customers and admins, and 18 use cases described in use case diagrams and use case specifications. At the system design stage, 18 activity diagrams, 18 sequence diagrams, class diagrams, entity-relationship diagrams, and 26 interface designs were produced. At the implementation stage using the concept of Model View Controller (MVC), React Js as View, and Codelgniter as Controller and Model. In the testing phase, in the validation test 31 test scenarios run as expected and the usability value is 82.75 (adjective rating excellent) for the customer side and 72.5 (adjective rating good) for the admin side through the usability scale system.

Keywords: *system information, order management, waterfall, validation testing, system usability scale, progressive web app*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xvii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	3
1.4 Manfaat	3
1.5 Batasan Masalah	3
1.6 Sistematika Pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Tinjauan Pustaka	5
2.2 Senjani Kitchen	6
2.3 <i>Progressive Web App</i>	6
2.4 Sistem Informasi	8
2.5 <i>Software Development Life Cycle (SDLC)</i>	8
2.6 <i>Waterfall</i>	8
2.7 <i>Unified Modelling Language (UML)</i>	9
2.7.1 <i>Use case diagram</i>	10
2.7.2 <i>Use Case Specification</i>	11
2.7.3 <i>Activity Diagram</i>	11
2.7.4 <i>Sequence Diagram</i>	12
2.7.5 <i>Class Diagram</i>	13
2.7.6 <i>Entity Relationship Diagram</i>	13
2.7.7 <i>Desain Antarmuka</i>	14

2.8 Business Process Model and Notation (BPMN)	15
2.9 Wireframe	16
2.10 Codeigniter framework	16
2.11 React Js library	16
2.12 Pengujian	17
2.12.1 Validation Testing	17
2.12.2 System Usability Scale (SUS)	17
BAB 3 METODOLOGI PENELITIAN	20
3.1 Tipe Penelitian	20
3.2 Tahapan Pelaksanaan	20
3.3 Identifikasi Masalah	21
3.4 Studi Literatur	21
3.5 Definisi Kebutuhan	21
3.6 Perancangan Sistem	22
3.7 Implementasi Sistem	22
3.8 Pengujian Sistem	22
3.9 Kesimpulan dan Saran	23
BAB 4 REKAYASA PERSYARATAN / KEBUTUHAN	24
4.1 Definisi Kebutuhan	24
4.1.1 Analisis Permasalahan	24
4.1.2 Pemodelan Proses Bisnis	25
4.1.2.1 Pembelian Kupon	25
4.1.2.2 Pemesanan Menu	27
4.1.3 Analisis Kebutuhan	28
4.1.3.1 Identifikasi Pengguna	28
4.1.3.2 Identifikasi Kebutuhan Fungsional	29
4.1.3.3 Identifikasi Kebutuhan Non Fungsional	30
4.1.3.4 Use Case Diagram	31
4.1.3.5 Use Case Specification	32
BAB 5 PERANCANGAN DAN IMPLEMENTASI	44
5.1 Perancangan Sistem	44
5.1.1 Activity Diagram	44
5.1.1.1 Register untuk Pelanggan	44



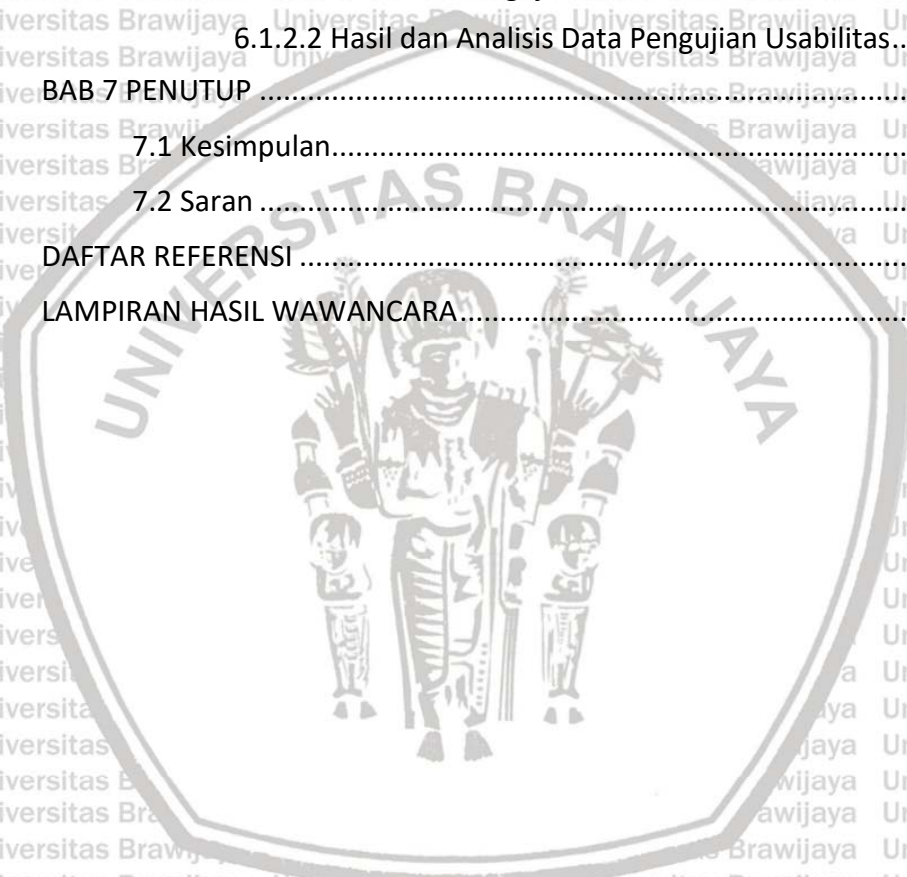
5.1.1.2 Login untuk Pelanggan.....	45
5.1.1.3 Logout untuk Pelanggan.....	45
5.1.1.4 Melihat Jadwal Menu untuk Pelanggan.....	45
5.1.1.5 Membeli Kupon untuk Pelanggan.....	46
5.1.1.6 Menggunakan Kupon untuk Pelanggan.....	47
5.1.1.7 Melihat Riwayat Pesanan untuk Pelanggan.....	48
5.1.1.8 Melihat Riwayat Kupon untuk Pelanggan.....	49
5.1.1.9 Login untuk Admin.....	49
5.1.1.10 Logout untuk Admin.....	50
5.1.1.11 Verifikasi Bukti Pembayaran untuk Admin.....	50
5.1.1.12 Membuat Jadwal Menu Baru untuk Admin.....	51
5.1.1.13 Mengedit Jadwal Menu untuk Admin.....	52
5.1.1.14 Mengedit Paket kupon untuk Admin.....	52
5.1.1.15 Melihat Hasil Pesanan untuk Admin.....	53
5.1.1.16 Mengedit Pesanan untuk Admin.....	53
5.1.1.17 Mengedit Pelanggan untuk Admin.....	54
5.1.1.18 Mengedit Kupon Pelanggan untuk Admin.....	55
5.1.2 <i>Sequence Diagram</i>	55
5.1.2.1 <i>Sequence Diagram</i> Membeli Kupon untuk Pelanggan.....	56
5.1.2.2 <i>Sequence Diagram</i> Menggunakan Kupon untuk Pelanggan.....	57
5.1.2.3 <i>Sequence Diagram</i> Verifikasi Bukti Pembayaran untuk Admin.....	59
5.1.2.4 <i>Sequence Diagram</i> Melihat Hasil Pesanan untuk Admin.....	61
5.1.3 <i>Class Diagram</i>	62
5.1.4 <i>Entity Relationship Diagram</i>	64
5.1.5 Desain Antarmuka.....	65
5.1.5.1 Desain Antarmuka Register untuk Pelanggan.....	65
5.1.5.2 Desain Antarmuka Utama untuk Pelanggan.....	66
5.1.5.3 Desain Antarmuka Login untuk Pelanggan.....	66
5.1.5.4 Desain Antarmuka Jadwal Menu untuk Pelanggan.....	67
5.1.5.5 Desain Antarmuka Pilih Paket Kupon untuk Pelanggan.....	67
5.1.5.6 Desain Antarmuka Metode Pembayaran untuk Pelanggan.....	68



5.1.5.7 Desain Antarmuka Form Bukti Pembayaran bagi Pelanggan	69
5.1.5.8 Desain Antarmuka Pilih Jadwal Menu bagi Pelanggan	69
5.1.5.9 Desain Antarmuka Form Detail Pengantaran bagi Pelanggan	70
5.1.5.10 Desain Antarmuka Riwayat Pesanan bagi Pelanggan	70
5.1.5.11 Desain Antarmuka Riwayat Kupon Pelanggan	71
5.1.5.12 Desain Antarmuka Login bagi Admin	72
5.1.5.13 Desain Antarmuka Utama bagi Admin	72
5.1.5.14 Desain Antarmuka Verifikasi Bukti Pembayaran bagi Admin	73
5.1.5.15 Desain Antarmuka Form Verifikasi Bukti Pembayaran bagi Admin	74
5.1.5.16 Desain Antarmuka Tabel Jadwal Menu bagi Admin	74
5.1.5.17 Desain Antarmuka Form Tambah Tabel Jadwal Menu bagi Admin	75
5.1.5.18 Desain Antarmuka Form Edit Tabel Jadwal Menu bagi Admin	75
5.1.5.19 Desain Antarmuka Tabel Paket Kupon bagi Admin	76
5.1.5.20 Desain Antarmuka Form Edit Tabel Paket Kupon bagi Admin	76
5.1.5.21 Desain Antarmuka Tabel Pesanan	77
5.1.5.22 Desain Antarmuka Form Edit Tabel Pesanan bagi Admin	77
5.1.5.23 Desain Antarmuka Tabel Pelanggan bagi Admin	78
5.1.5.24 Desain Antarmuka Form Edit Tabel Pelanggan bagi Admin	78
5.1.5.25 Desain Antarmuka Tabel Kupon Pelanggan	79
5.1.5.26 Desain Antarmuka Form Edit Tabel Kupon Pelanggan	79
5.2 Implementasi Sistem	80
5.2.1 Spesifikasi Sistem	80
5.2.2 Implementasi Kode	81
5.2.2.1 Implementasi Register untuk Pelanggan	81
5.2.2.2 Implementasi Login untuk Pelanggan	83
5.2.2.3 Implementasi Logout untuk Pelanggan	85
5.2.2.4 Implementasi Melihat Jadwal Menu untuk Pelanggan	86

5.2.2.5 Implementasi Membeli Kupon untuk Pelanggan.....	87
5.2.2.6 Implementasi Menggunakan Kupon untuk Pelanggan.....	91
5.2.2.7 Implementasi Melihat Riwayat Pesanan untuk Pelanggan	94
5.2.2.8 Implementasi Melihat Riwayat Kupon untuk Pelanggan..	98
5.2.2.9 Implementasi Progressive Web App untuk Pelanggan.....	99
5.2.2.10 Implementasi Login untuk Admin.....	101
5.2.2.11 Implementasi Logout untuk Admin.....	103
5.2.2.12 Implementasi Verifikasi Bukti Pembayaran untuk Admin	104
5.2.2.13 Implementasi Membuat Jadwal Menu Baru untuk Admin	107
5.2.2.14 Implementasi Mengedit Jadwal Menu untuk Admin ...	109
5.2.2.15 Implementasi Mengedit Paket Kupon untuk Admin ...	111
5.2.2.16 Implementasi Melihat Hasil Pesanan untuk Admin.....	113
5.2.2.17 Implementasi Mengedit Pesanan untuk Admin.....	114
5.2.2.18 Implementasi Mengedit Pelanggan untuk Admin.....	115
5.2.2.19 Implementasi Mengedit Kupon Pelanggan.....	117
5.2.2.20 Implementasi Progressive Web App untuk Admin.....	119
BAB 6 PENGUJIAN	122
6.1 Pengujian Sistem.....	122
6.1.1 Pengujian Validasi	122
6.1.1.1 Register untuk Pelanggan	122
6.1.1.2 Login untuk Pelanggan.....	123
6.1.1.3 Logout untuk Pelanggan	124
6.1.1.4 Melihat Jadwal Menu untuk Pelanggan.....	124
6.1.1.5 Membeli Kupon untuk Pelanggan.....	125
6.1.1.6 Menggunakan Kupon untuk Pelanggan.....	126
6.1.1.7 Melihat Riwayat Kupon untuk Pelanggan.....	127
6.1.1.8 Melihat Riwayat Pesanan untuk Pelanggan.....	128
6.1.1.9 Progressive Web App untuk Pelanggan.....	128
6.1.1.10 Login untuk Admin.....	129
6.1.1.11 Logout untuk Admin.....	130
6.1.1.12 Verifikasi Bukti Pembayaran untuk Admin	130
6.1.1.13 Membuat Jadwal Menu Baru untuk Admin.....	131

6.1.1.14 Mengedit Jadwal Menu untuk Admin.....	132
6.1.1.15 Mengedit Paket Kupon untuk Admin.....	133
6.1.1.16 Melihat Hasil Pesanan untuk Admin.....	134
6.1.1.17 Mengedit Pesanan untuk Admin.....	135
6.1.1.18 Mengedit Pelanggan untuk Admin.....	136
6.1.1.19 Mengedit Kupon Pelanggan untuk Admin.....	137
6.1.1.20 Progressive Web App untuk Admin.....	138
6.1.2 Pengujian Usabilitas.....	138
6.1.2.1 Prosedur Pengujian Usabilitas.....	138
6.1.2.2 Hasil dan Analisis Data Pengujian Usabilitas.....	140
BAB 7 PENUTUP	142
7.1 Kesimpulan.....	142
7.2 Saran.....	143
DAFTAR REFERENSI	144
LAMPIRAN HASIL WAWANCARA	146



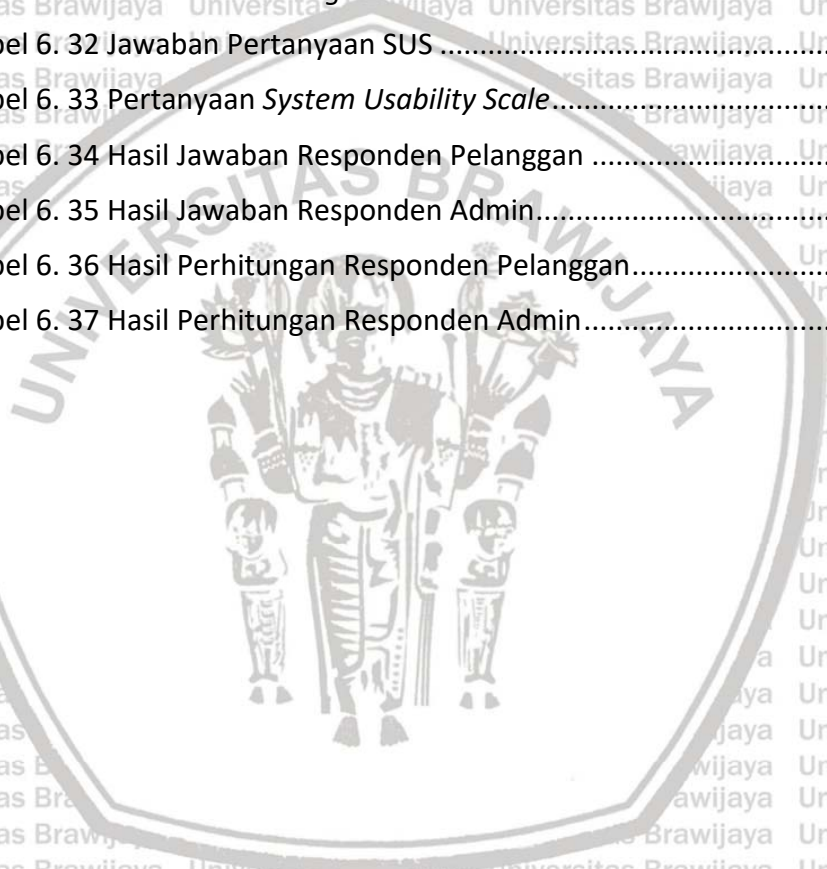
DAFTAR TABEL

Tabel 2. 1 Notasi <i>Use Case Diagram</i>	10
Tabel 2. 2 Notasi <i>Activity Diagram</i>	11
Tabel 2. 3 Notasi <i>Sequence Diagram</i>	12
Tabel 2. 4 Notasi <i>Class Diagram</i>	13
Tabel 2. 5 Notasi <i>Entity Relationship Diagram</i>	14
Tabel 2. 6 Tabel Notasi BPMN	15
Tabel 4. 1 Tabel Identifikasi Pengguna	28
Tabel 4. 2 Tabel Identifikasi Kebutuhan Fungsional	29
Tabel 4. 3 Tabel Identifikasi Non Fungsional	30
Tabel 4. 4 Tabel <i>Use Case Specification Register</i>	32
Tabel 4. 5 Tabel <i>Use Case Specification Login untuk Pelanggan</i>	33
Tabel 4. 6 Tabel <i>Use Case Specification Logout untuk Pelanggan</i>	33
Tabel 4. 7 <i>Use Case Specification</i> Melihat Jadwal Menu	34
Tabel 4. 8 <i>Use Case Spesification</i> Membeli Kupon	34
Tabel 4. 9 <i>Use Case Spesification</i> Menggunakan Kupon	36
Tabel 4. 10 <i>Use Case Spesification</i> Melihat Riwayat Pesanan	36
Tabel 4. 11 <i>Use Case Spesification</i> Melihat Riwayat Kupon	36
Tabel 4. 12 <i>Use Case Spesification Login</i> untuk Admin	37
Tabel 4. 13 <i>Use Case Spesification Logout</i> untuk Admin	37
Tabel 4. 14 <i>Use Case Spesification</i> Verifikasi Bukti Pembayaran	38
Tabel 4. 15 <i>Use Case Spesification</i> Membuat Jadwal Menu Baru	39
Tabel 4. 16 <i>Use Case Spesification</i> Mengedit Jadwal Menu	39
Tabel 4. 17 <i>Use Case Spesification</i> Mengedit Paket kupon	40
Tabel 4. 18 <i>Use Case Spesification</i> Melihat Hasil Pesanan	41
Tabel 4. 19 <i>Use Case Spesification</i> Mengedit Pesanan	41
Tabel 4. 20 <i>Use Case Spesification</i> Mengedit Pelanggan	42
Tabel 4. 21 <i>Use Case Spesification</i> Mengedit Kupon Pelanggan	43

Tabel 5. 1 Perangkat Keras.....	80
Tabel 5. 2 Perangkat Lunak.....	81
Tabel 5. 3 Kode <i>backend register()</i> pada <i>Controller</i> Pelanggan.....	82
Tabel 5. 4 Kode <i>Backend Login</i> pada <i>Controller</i> Pelanggan.....	84
Tabel 5. 5 Kode <i>Frontend Akun</i>	85
Tabel 5. 6 Kode <i>Backend Index</i> pada <i>Controller</i> Menu.....	87
Tabel 5. 7 Kode <i>Backend ShowPaketKuponBerdasarkanJenisPaketKupon</i> pada <i>Controller</i> PaketKupon.....	88
Tabel 5. 8 Kode <i>Backend KuponPelanggan</i> pada <i>Controller</i> Kupon Pelanggan....	89
Tabel 5. 9 Kode <i>Backend UpdateKuponPelanggan</i> pada <i>Controller</i> Kupon Pelanggan.....	90
Tabel 5. 10 Kode <i>Backend showAllMenu</i> pada <i>Controller</i> Menu.....	92
Tabel 5. 11 Kode <i>Backend CreatePesanan</i> pada <i>Controller</i> Pesanan.....	93
Tabel 5. 12 Kode <i>Backend GetPesananKuponPelangganMenu</i> pada <i>Model</i> Menu.....	95
Tabel 5. 13 Kode <i>Backend ShowPesananBerdasarkanIdPelanggan</i> pada <i>Controller</i> Mix.....	95
Tabel 5. 14 Kode <i>Frontend DetailPesanan</i>	96
Tabel 5. 15 Kode <i>Backend GetPaketKuponPelanggan</i> pada <i>Model</i> Mix.....	99
Tabel 5. 16 Kode <i>Backend ShowKuponPelangganBerdasarkanIdPelanggan</i> pada <i>Controller</i> Mix.....	99
Tabel 5. 17 Kode <i>Frontend Manifest</i> untuk Pelanggan.....	100
Tabel 5. 18 Kode <i>Frontend Index</i> untuk Pelanggan.....	101
Tabel 5. 19 Kode <i>Backend login</i> untuk Admin.....	102
Tabel 5. 20 Kode <i>Frontend DrawerComponents</i>	104
Tabel 5. 21 Kode <i>Backend Model Kupon Pelanggan</i>	106
Tabel 5. 22 Kode <i>Backend showBuktiPembayaranStatusMenungguDiverifikasi</i>	106
Tabel 5. 23 Kode <i>Backend adminUpdateKuponPelanggan</i>	107
Tabel 5. 24 Kode <i>Backend showAllMenu</i> pada <i>Controller</i> Menu.....	108
Tabel 5. 25 Kode <i>Backend createMenu</i> pada <i>Controller</i> Menu.....	109
Tabel 5. 26 Kode <i>Backend updateMenu</i> pada <i>Controller</i> Menu.....	110
Tabel 5. 27 Kode <i>Backend showAllPaketKupon</i> pada <i>Controller</i> Paket Kupon ..	112
Tabel 5. 28 Kode <i>Backend updatePaketKupon</i> pada <i>Controller</i> Paket Kupon...	112
Tabel 5. 29 Kode <i>Backend showAllPesanan</i> pada <i>Controller</i> Pesanan.....	113

Tabel 5. 30 Kode <i>Backend</i> adminUpdatePesanan pada <i>Controller</i> Pesanan.....	114
Tabel 5. 31 Kode <i>Backend</i> showAllPelanggan pada <i>Controller</i> Pelanggan.....	116
Tabel 5. 32 Kode <i>Backend</i> updatePelanggan pada <i>Controller</i> Pelanggan	116
Tabel 5. 33 Kode <i>Backend</i> showAllKuponPelanggan pada <i>Controller</i> Mix.....	118
Tabel 5. 34 Kode <i>Backend</i> getAllKuponPelangganPaketKuponPelanggan pada Model Mix.....	118
Tabel 5. 35 Kode <i>Backend</i> adminUpdateKuponPelanggan pada <i>Controller</i> Kupon Pelanggan.....	118
Tabel 5. 36 Kode <i>Frontend Manifest</i> untuk Admin.....	120
Tabel 5. 37 Kode <i>Frontend Index</i> untuk Admin.....	121
Tabel 6. 1 <i>Validation Testing Register Email</i> Belum Terdaftar	122
Tabel 6. 2 <i>Validation Testing Register Email</i> Telah Terdaftar.....	123
Tabel 6. 3 <i>Validation Testing</i> Pelanggan Berhasil.....	123
Tabel 6. 4 <i>Validation Testing</i> Pelanggan Gagal.....	123
Tabel 6. 5 <i>Validation Testing Logout</i> Pelanggan.....	124
Tabel 6. 6 <i>Validation Testing</i> Melihat Jadwal Menu.....	124
Tabel 6. 7 <i>Validation Testing</i> Membeli Kupon Unggah Gambar.....	125
Tabel 6. 8 <i>Validation Testing</i> Membeli Kupon Unggah Bukan Gambar.....	125
Tabel 6. 9 <i>Validation Testing</i> Menggunakan Kupon Berhasil	126
Tabel 6. 10 <i>Validation Testing</i> Menggunakan Kupon Gagal	127
Tabel 6. 11 <i>Validation Testing</i> Melihat Riwayat Kupon.....	127
Tabel 6. 12 <i>Validation Testing</i> Melihat Riwayat Pesanan.....	128
Tabel 6. 13 <i>Validation Testing PWA</i> untuk Pelanggan.....	128
Tabel 6. 14 <i>Validation Testing Login</i> Admin Berhasil.....	129
Tabel 6. 15 <i>Validation Testing Login</i> Admin Gagal	129
Tabel 6. 16 <i>Validation Testing Logout</i> Admin.....	130
Tabel 6. 17 <i>Validation Testing</i> Verifikasi Bukti Pembayaran Valid	130
Tabel 6. 18 <i>Validation Testing</i> Verifikasi Bukti Pembayaran Tidak Valid.....	131
Tabel 6. 19 <i>Validation Testing</i> Membuat Jadwal Menu Baru.....	131
Tabel 6. 20 <i>Validation Testing</i> Mengedit Jadwal Menu Berhasil.....	132
Tabel 6. 21 <i>Validation Testing</i> Mengedit Jadwal Menu Gagal.....	133
Tabel 6. 22 <i>Validation Testing</i> Mengedit Paket Kupon Berhasil.....	133

Tabel 6. 23 <i>Validation Testing</i> Mengedit Paket Kupon Gagal.....	134
Tabel 6. 24 <i>Validation Testing</i> Melihat Hasil Pesanan	134
Tabel 6. 25 <i>Validation Testing</i> Mengedit Pesanan Berhasil.....	135
Tabel 6. 26 <i>Validation Testing</i> Mengedit Pesanan.....	135
Tabel 6. 27 <i>Validation Testing</i> Mengedit Pelanggan Berhasil	136
Tabel 6. 28 <i>Validation Testing</i> Mengedit Pelanggan Gagal	136
Tabel 6. 29 <i>Validation Testing</i> Mengedit Kupon Pelanggan	137
Tabel 6. 30 <i>Validation Testing</i> Mengedit Kupon Pelanggan	137
Tabel 6. 31 <i>Validation Testing</i> PWA untuk Admin	138
Tabel 6. 32 Jawaban Pertanyaan SUS	139
Tabel 6. 33 Pertanyaan <i>System Usability Scale</i>	139
Tabel 6. 34 Hasil Jawaban Responden Pelanggan	140
Tabel 6. 35 Hasil Jawaban Responden Admin.....	140
Tabel 6. 36 Hasil Perhitungan Responden Pelanggan.....	140
Tabel 6. 37 Hasil Perhitungan Responden Admin.....	141



DAFTAR GAMBAR

Gambar 2. 1 <i>Capabilities & Reach PWA</i>	7
Gambar 2. 2 Tahapan Metode <i>Waterfall</i>	9
Gambar 2. 3 <i>System Usability Scale Answer</i>	18
Gambar 2. 4 <i>System Usability Scale Question</i>	18
Gambar 2. 5 <i>System Usability Scale Rating</i>	19
Gambar 3. 1 Tahapan Penelitian.....	20
Gambar 4. 1 BPMN <i>As-Is</i> Pembelian Kupon.....	26
Gambar 4. 2 BPMN <i>To-Be</i> Pembelian Kupon.....	26
Gambar 4. 3 BPMN <i>As-Is</i> Pemesanan Menu.....	27
Gambar 4. 4 BPMN <i>To-Be</i> Pemesanan Menu.....	28
Gambar 4. 5 <i>Use Case Diagram</i>	31
Gambar 5. 1 <i>Activity Diagram Register</i>	44
Gambar 5. 2 <i>Activity Diagram Login</i> untuk Pelanggan.....	45
Gambar 5. 3 <i>Activity Diagram Logout</i> untuk Pelanggan.....	45
Gambar 5. 4 <i>Activity Diagram</i> Melihat Jadwal Menu.....	46
Gambar 5. 5 <i>Activity Diagram</i> Membeli Kupon.....	47
Gambar 5. 6 <i>Activity Diagram</i> Menggunakan Kupon.....	48
Gambar 5. 7 <i>Activity Diagram</i> Melihat Riwayat Pesanan.....	49
Gambar 5. 8 <i>Activity Diagram</i> Melihat Riwayat Kupon.....	49
Gambar 5. 9 <i>Activity Diagram Login</i> untuk Admin.....	50
Gambar 5. 10 <i>Activity Diagram Logout</i> untuk Admin.....	50
Gambar 5. 11 <i>Activity Diagram</i> Verifikasi Bukti Pembayaran.....	51
Gambar 5. 12 <i>Activity Diagram</i> Membuat Jadwal Menu Baru.....	52
Gambar 5. 13 <i>Activity Diagram</i> Mengedit Jadwal Menu.....	52
Gambar 5. 14 <i>Activity Diagram</i> Mengedit Paket Kupon.....	53
Gambar 5. 15 <i>Activity Diagram</i> Melihat Hasil Pesanan.....	53
Gambar 5. 16 <i>Activity Diagram</i> Mengedit Pesanan.....	54
Gambar 5. 17 <i>Activity Diagram</i> Mengedit Pelanggan.....	54



Gambar 5. 18 <i>Activity Diagram</i> Mengedit Kupon Pelanggan	55
Gambar 5. 19 <i>Sequence Diagram</i> Membeli Kupon untuk Pelanggan.....	56
Gambar 5. 20 <i>Sequence Diagram</i> Menggunakan Kupon untuk Pelanggan	58
Gambar 5. 21 <i>Sequence Diagram</i> Verifikasi Bukti Pembayaran untuk Admin	60
Gambar 5. 22 <i>Sequence Diagram</i> Melihat Hasil Pesanan untuk Admin	62
Gambar 5. 23 <i>Class Diagram Controller</i>	63
Gambar 5. 24 <i>Class Diagram Models</i>	63
Gambar 5. 25 <i>Entity Relationship Diagram</i>	64
Gambar 5. 26 Desain Antarmuka <i>Register</i> untuk Pelanggan.....	65
Gambar 5. 27 Desain Antarmuka Utama untuk Pelanggan	66
Gambar 5. 28 Desain Antarmuka <i>Login</i> untuk Pelanggan	66
Gambar 5. 29 Desain Antarmuka Jadwal Menu untuk Pelanggan	67
Gambar 5. 30 Desain Antarmuka Pilih Paket Kupon untuk Pelanggan.....	68
Gambar 5. 31 Desain Antarmuka Metode Pembayaran untuk Pelanggan.....	68
Gambar 5. 32 Desain Antarmuka Form Bukti Pembayaran bagi Pelanggan	69
Gambar 5. 33 Desain Antarmuka Pilih Jadwal Menu bagi Pelanggan	70
Gambar 5. 34 Desain Antarmuka Form Detail Pengantaran bagi Pelanggan	70
Gambar 5. 35 Desain Antarmuka Riwayat Pesanan bagi Pelanggan	71
Gambar 5. 36 Desain Antarmuka Riwayat Kupon Pelanggan	72
Gambar 5. 37 Desain Antarmuka <i>Login</i> bagi Admin	72
Gambar 5. 38 Desain Antarmuka Utama bagi Admin.....	73
Gambar 5. 39 Desain Antarmuka Verifikasi Bukti Pembayaran bagi Admin	73
Gambar 5. 40 Desain Antarmuka Form Verifikasi Bukti Pembayaran bagi Admin	74
Gambar 5. 41 Desain Antarmuka Tabel Jadwal Menu bagi Admin	74
Gambar 5. 42 Desain Antarmuka Form Tambah Tabel Jadwal Menu bagi Admin	75
Gambar 5. 43 Desain Antarmuka Form Edit Tabel Jadwal Menu bagi Admin	75
Gambar 5. 44 Desain Antarmuka Tabel Paket Kupon bagi Admin	76
Gambar 5. 45 Desain Antarmuka Form Edit Tabel Paket Kupon bagi Admin	77
Gambar 5. 46 Desain Antarmuka Tabel Pesanan.....	77
Gambar 5. 47 Desain Antarmuka Form Edit Tabel Pesanan bagi Admin.....	78
Gambar 5. 48 Desain Antarmuka Tabel Pelanggan bagi Admin	78
Gambar 5. 49 Desain Antarmuka Form Edit Tabel Pelanggan bagi Admin.....	79

Gambar 5. 50 Desain Antarmuka Tabel Kupon Pelanggan	79
Gambar 5. 51 Desain Antarmuka Form Edit Tabel Kupon Pelanggan	80
Gambar 5. 52 Hasil Implementasi <i>Login</i> untuk Pelanggan	82
Gambar 5. 53 Hasil Implementasi <i>Login</i> untuk Pelanggan	83
Gambar 5. 54 Hasil Implementasi Halaman Utama untuk Pelanggan	84
Gambar 5. 55 Hasil Implementasi <i>Logout</i> untuk Pelanggan	85
Gambar 5. 56 Hasil Implementasi Melihat Jadwal Menu untuk Pelanggan	87
Gambar 5. 57 Hasil Implementasi Pilih Paket Kupon untuk Pelanggan	88
Gambar 5. 58 Hasil Implementasi Metode Pembayaran untuk Pelanggan	89
Gambar 5. 59 Hasil Implementasi Unggah Bukti Pembayaran untuk Pelanggan	90
Gambar 5. 60 Hasil Implementasi Pilih Menu untuk Pelanggan	92
Gambar 5. 61 Hasil Implementasi Form Pengantaran untuk Pelanggan	93
Gambar 5. 62 Hasil Implementasi Riwayat Pesanan untuk Pelanggan	95
Gambar 5. 63 Hasil Implementasi Detail Pesanan untuk Pelanggan	96
Gambar 5. 64 Hasil Implementasi Riwayat Kupon untuk Pelanggan	98
Gambar 5. 65 Bukti Implementasi <i>PWA</i> untuk Pelanggan	100
Gambar 5. 66 Proses Pemasangan <i>PWA</i> untuk Pelanggan	100
Gambar 5. 67 Hasil Implementasi <i>Login</i> untuk Admin	102
Gambar 5. 68 Hasil Implementasi Halaman Utama untuk Admin	102
Gambar 5. 69 Hasil Implementasi <i>Logout</i> untuk Admin	103
Gambar 5. 70 Hasil Implementasi Tampilan Tabel Bukti Pembayaran untuk Admin	105
Gambar 5. 71 Hasil Implementasi Tampilan <i>Form</i> Verifikasi untuk Admin	106
Gambar 5. 72 Hasil Implementasi Tampilan Tabel Menu untuk Admin	108
Gambar 5. 73 Hasil Implementasi Tampilan <i>Form</i> Tambah Menu untuk Admin	108
Gambar 5. 74 Hasil Implementasi <i>Form</i> Edit untuk Admin	110
Gambar 5. 75 Hasil Implementasi Tabel Paket Kupon untuk Admin	111
Gambar 5. 76 Hasil Implementasi <i>Form</i> Edit Menu	112
Gambar 5. 77 Hasil Implementasi Tabel Pesanan untuk Admin	113
Gambar 5. 78 Hasil Implementasi <i>Form</i> Edit untuk Admin	114
Gambar 5. 79 Hasil Implementasi Tabel Pelanggan untuk Admin	115
Gambar 5. 80 Hasil Implementasi <i>Form</i> Edit Pelanggan untuk Admin	116
Gambar 5. 81 Hasil Implementasi Tampilan Tabel Kupon Pelanggan	117

Gambar 5. 82 Hasil Implementasi Tampilan *Form* Edit Kupon Pelanggan 118
Gambar 5. 83 Bukti Implementasi *PWA* untuk Admin 120
Gambar 5. 84 Proses Pemasangan *PWA* untuk Admin 120



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Sistem informasi merupakan himpunan komponen-komponen yang saling terkait satu sama lain untuk melakukan pengumpulan, pemrosesan, penyimpanan, dan pendistribusian informasi. Salah satu faktor penting kelancaran kinerja perusahaan adalah informasi. Pengelolaan informasi yang baik akan berdampak ke perusahaan tersebut, perusahaan akan berkembang dengan cepat karena adanya perbaikan pada informasi yang dikelola oleh perusahaan tersebut (Aknovia, 2018). Jika perusahaan tidak mau berkembang dan menyesuaikan dengan teknologi sesuai zamannya, perusahaan akan sulit berkompetisi dengan pesaing perusahaan tersebut. Namun, bila perusahaan tersebut dapat mengelola sistem informasi mereka dengan baik maka mereka akan bisa berkembang dan justru perusahaan dapat mengalahkan pesaingnya.

Senjani Kitchen dengan nama resmi CV Senjani Tekno Boga, perusahaan rintisan yang bergerak dibidang pangan. Dari hasil wawancara, Senjani Kitchen telah melayani kawasan Kota Malang, Kota Batu, dan Kabupaten Malang dengan sekitar 100 pelanggan aktif dalam layanan Daily Catering. Proses bisnis Daily Catering saat ini, pertama-tama pelanggan melakukan *chat* ke admin Senjani Kitchen via Whatsapp untuk berlanggan Daily Catering. Lalu, pelanggan memilih salah satu jenis paket kupon yaitu Basic, Reusable, Deluxe, atau Family. Selanjutnya, admin akan memberikan *Form Online Link* sesuai dengan jenis paket yang dipilih. Berikutnya, pelanggan mengisi *Form Online* tersebut sesuai dengan kebutuhan seperti jumlah kupon yang dapat ditukarkan dengan makanan, jenis nasi, harga, alamat pengiriman, alergi makanan bila ada, dan waktu kapan saja makanan dikirimkan. Kemudian, pelanggan melakukan pembayaran dan mengirimkan bukti pembayaran ke admin Senjani Kitchen lewat Whatsapp. Setelah data pesanan dan bukti pembayaran lengkap, admin akan mengirim tabel data pesanan ke staf dapur dan staf pengiriman untuk diproduksi dan diantarkan sesuai dengan waktu pemesanan.

Dari hasil wawancara dengan pihak Senjani Kitchen didapatkan permasalahan di sistem pemesanan Daily Catering. Selama ini pemesanan Daily Catering oleh pelanggan melalui aplikasi *chat* Whatsapp menggunakan *smartphone* mereka, sehingga pelanggan terlalu bergantung kepada balasan admin. Karena keterbatasan waktu admin tidak bisa menjawab dengan cepat. Pelanggan seharusnya dapat mendapatkan informasi seperti jadwal makanan, paket kupon, harga, dan memesan waktu pemesanan lebih fleksibel tanpa harus menunggu balasan dari admin. Dikarenakan pencatatan pemesanan masih manual sering terjadi kesalahan pencatatan. Sehingga mengakibatkan kerugian mulai yang tidak fatal seperti pakai sambal atau tidak, hingga yang fatal seperti tercatat minta dikirim namun ternyata tidak sehingga Senjani Kitchen harus menanggung kerugian atas makanan tersebut.

Dari masalah tersebut, penulis ingin mengembangkan sebuah sistem informasi pemesanan dalam layanan Daily Catering. Sehingga dari sisi pelanggan lebih bebas dalam mendapatkan informasi dan memesan makanan tanpa ketergantungan dari admin. Diharapkan sistem informasi tersebut menyelesaikan 2 proses bisnis utama yaitu pembelian kupon dan pemesanan menu. Dari sisi pelanggan fitur yang diharapkan yaitu pembelian kupon, menggunakan kupon atau memesan menu dengan jadwal lebih fleksibel, melihat riwayat pesanan, melihat riwayat kupon, dan nyaman digunakan di *smartphone*. Sedangkan dari sisi admin fitur yang diharapkan yaitu menverifikasi bukti pembayaran, menambahkan dan mengedit jadwal makanan, hingga melihat hasil pemesanan.

Salah satu media sistem informasi yaitu aplikasi *web*. Contoh aplikasi *web* yang menggunakan teknologi modern adalah *Progressive Web App*. *Progressive Web App* adalah teknologi yang digunakan untuk melakukan optimasi pada *website*, sehingga *website* menjadi lebih cepat dan mampu untuk dijalankan layaknya aplikasi *mobile native* pada umumnya (Nurwanto, 2019). Dikarenakan *PWA* merupakan *web* sehingga dapat diakses berbagai macam ukuran *device* dari *smartphone*, *tablet*, hingga *PC/laptop* sehingga penggunaan *PWA* sesuai dengan kebutuhan. *PWA* dapat menyelesaikan masalah tanpa mengorbankan kenyamanan pelanggan yang telah terbiasa memesan Daily Catering melalui pesan singkat Whatsapp menggunakan *smartphone* mereka, karena *PWA* memiliki kemampuan yaitu nyaman digunakan di perangkat *mobile*. Oleh karena itu, *PWA* dipilih agar pelanggan mendapatkan pengalaman yang lebih nyaman tanpa mengorbankan aksesibilitas perangkat.

Dalam pengembangan ini siklus hidup pengembangan sistem menggunakan metode *waterfall* dikarenakan telah memiliki tujuan yang jelas. Metode *waterfall* memiliki ciri tahapan yang terstruktur sehingga cocok bila digunakan pada kebutuhan pengguna yang spesifik dan tidak banyak perubahan dalam pengembangan sistem informasi nantinya (Sommerville, 2011). *Waterfall* mempunyai beberapa tahapan yaitu, penggalan kebutuhan, perancangan, implementasi, pengujian, hingga pemeliharaan. Setiap tahapan yang ada menghasilkan dokumen sendiri dan tidak dapat melompati ke tahapan selanjutnya bila tahapan saat ini belum selesai. Oleh karena itu, penelitian ini menggunakan metode *waterfall* dikarenakan permasalahan dan definisi yang jelas.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang dirumuskan di atas, kemudian kami merumuskan masalah sebagai berikut:

1. Bagaimana hasil analisis kebutuhan sistem pemesanan Daily Catering Senjani Kitchen berbasis *Progressive Web App*?
2. Bagaimana hasil perancangan sistem pemesanan Daily Catering Senjani Kitchen berbasis *Progressive Web App*?

3. Bagaimana hasil implementasi sistem pemesanan Daily Catering Senjani Kitchen berbasis *Progressive Web App* berdasarkan hasil perancangan?
4. Bagaimana hasil pengujian sistem pemesanan Daily Catering Senjani Kitchen berbasis *Progressive Web App* berdasarkan hasil implementasi?

1.3 Tujuan

Tujuan dari pengembangan sistem informasi *Progressive Web App* pemesanan Daily Catering Senjani Kitchen adalah :

1. Mengetahui hasil analisis kebutuhan sistem pemesanan Daily Catering Senjani Kitchen berbasis *Progressive Web App*
2. Mengetahui hasil perancangan sistem pemesanan Daily Catering Senjani Kitchen berbasis *Progressive Web App*
3. Mengetahui hasil implementasi sistem pemesanan Daily Catering Senjani Kitchen berbasis *Progressive Web App*
4. Mengetahui hasil pengujian sistem pemesanan Daily Catering Senjani Kitchen berbasis *Progressive Web App*

1.4 Manfaat

Manfaat yang bisa didapatkan dalam penelitian ini yaitu membuat sistem informasi untuk memudahkan pelanggan dan admin dalam proses pemesanan Daily Catering Senjani Kitchen.

1.5 Batasan Masalah

Adapun kami membatasi masalah pada pengembangan sistem informasi pemesanan Daily Catering Senjani Kitchen berbasis *Progressive Web App* pada:

1. Pengembangan hanya berfokus pada halaman pemesanan pelanggan dan admin
2. Sistem informasi dikembangkan dengan berbasis *mobile web oriented*
3. Menggunakan *framework* PHP CodeIgniter untuk *backend*
4. Menggunakan React Js *library* untuk *frontend*
5. Dikembangkan dengan metode *waterfall*
6. Pengujian menggunakan *Validation Testing* dan *System Usability Scale*

1.6 Sistematika Pembahasan

Sistematika pembahasan akan digunakan sebagai gambaran secara umum tahapan yang akan dilakukan dalam melakukan penelitian. Tahapan tersebut meliputi :

BAB I Pendahuluan

Pada bab pendahuluan penelitian ini berisi tentang latar belakang, tujuan, manfaat, batasan masalah dan sistematika penulisan laporan.

BAB II Landasan Kepustakaan

Pada bab ini berisikan kajian pustaka dan referensi penelitian yang telah dilakukan sebelumnya yang digunakan sebagai salah satu dasar penelitian.

BAB III Metodologi Penelitian

Pada bab ini berisikan langkah – langkah yang akan dilakukan ketika melakukan penelitian dengan metode *waterfall*.

BAB IV Analisis dan Perancangan

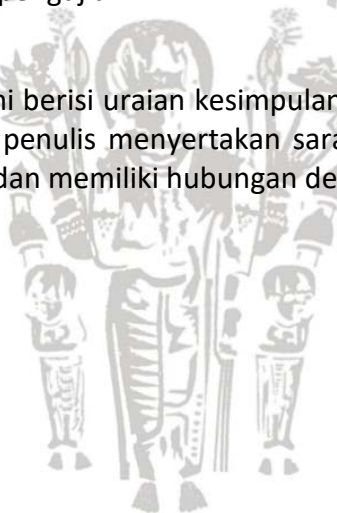
Pada bab ini berisi hasil analisis kebutuhan dari aplikasi yang akan dikembangkan oleh penulis, baik itu kebutuhan fungsional maupun non fungsional.

BAB V Hasil dan Pembahasan

Pada bab ini berisi hasil dari penelitian diantaranya, hasil perancangan, implementasi dan pengujian.

BAB VI Penutup

Pada bab ini berisi uraian kesimpulan yang didapatkan dari penelitian ini. Pada bab ini juga penulis menyertakan saran-saran untuk penelitian-penelitian yang akan datang dan memiliki hubungan dengan topik pada penelitian ini.



BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini akan membahas mengenai teori-teori yang akan digunakan sebagai landasan dari penelitian ini serta berhubungan dengan pengembangan sistem informasi pemesanan Daily Catering.

2.1 Tinjauan Pustaka

Tinjauan pustaka yang digunakan sebagai panduan dan sumber referensi oleh peneliti untuk melakukan penelitian ini didapatkan dari jurnal dan penelitian yang dilakukan oleh peneliti sebelumnya. Penelitian yang pertama bersumber dari Laar, konjaang, dan tankia (2015) dengan judul "*Design and Development of a Sales Management System for SMEs in Northern Ghana*". Dalam penelitian tersebut menjelaskan manfaat-manfaat jika perusahaan kecil menggunakan sistem manajemen berbasis komputer, yaitu pembuatan laporan lebih mudah apabila sistem dikelola dengan baik, sistem dengan ramah antarmuka akan mudah digunakan, dan sistem yang menyelesaikan permasalahan perusahaan akan berdampak baik kepada perusahaan itu sendiri.

Penelitian yang kedua bersumber dari Sadi, Dyah Rachmawati, dan M Shodiq Abdul (2019) dengan judul "*Designing Mobile Alumni Tracer Study System Using Waterfall Method*". Dalam Penelitian tersebut menggunakan metode pengembangan *Waterfall* untuk membangun aplikasi *mobile android* dengan nama *Alumni Tracer Study*. Metode *waterfall* dipilih dikarenakan memiliki sifat sistematis dan berurutan. Penelitian ini memiliki tingkat tes *usability* sebesar 78% yang telah melebihi standar *usability* sebesar 68%. Penelitian ini yang keluar pada tahun 2019 yang membuktikan metode *Waterfall* masih digunakan dan masih relevan untuk digunakan saat ini.

Penelitian yang ketiga dilakukan oleh Try Ratnasari, Awalludiyah Ambarwati, Mohammad Noor Al Azam (2017) dengan judul "Rancang Bangun Sistem Informasi Manajemen Proyek untuk Pengembang Perangkat Lunak pada Pt. Quantum Leap". Penelitian tersebut menyelesaikan masalah pencatatan yang masih dilakukan manual dan menggunakan aplikasi *spreadsheet*. Penelitian ini membuat perangkat lunak *website* untuk memenuhi kebutuhan diantaranya alokasi sumber daya, penjadwalan proyek, pelaporan status proyek, dan dokumentasi proyek. Pengembangan perangkat lunak ini menggunakan *Software Development Life Cycle (SDLC)* berupa model *waterfall*.

Penelitian yang keempat bersumber dari Irena Vuksanovic dan Sudarevic (2011) dengan judul "*Use of Web Application Framework in the Development of Small Applications*". Penelitian berisi tentang perbandingan kegiatan pengembangan aplikasi yang menggunakan *framework* dan aplikasi lainnya tidak menggunakan *framework*. Hasil dari penelitian yang dilakukan, didapatkan bahwa durasi pengerjaan aplikasi yang tidak menggunakan *framework* membutuhkan durasi sekitar 19 jam. Sedangkan pengembangan aplikasi dengan menggunakan *framework* membutuhkan durasi sekitar 11 jam. Menghasilkan kesimpulan bahwa

aplikasi yang dikerjakan menggunakan *framework* dapat mengurangi waktu pengerjaan pengembangan aplikasi.

Penelitian yang kelima bersumber dari Aminudin, Basri Basren, dan Ilyas Nuryasin (2019) dengan judul “Perancangan Sistem Repositori Tugas Akhir Menggunakan *Progressive Web App (PWA)*”. Penelitian ini berisi tentang pengembangan sistem repositori tugas akhir untuk ruang baca di fakultas teknik UMM menggunakan teknologi optimasi web *progressive web app*. Penelitian ini berhasil mengimplementasikan *service worker* bagian dari *PWA* sehingga web tidak terlalu bergantung pada konektivitas jaringan. Hasil pengujian penelitian ini berhasil memenuhi 14 dari 15 syarat *baseline PWA checklist* yang ada sehingga dapat disimpulkan bahwa *PWA* berhasil diimplementasikan.

2.2 Senjani Kitchen

Senjani Kitchen dengan nama resmi CV Senjani Tekno Boga merupakan perusahaan rintisan yang bergerak dibidang pangan. Senjani Kitchen melayani kebutuhan makan harian, catering acara hingga frozen food. Beroperasi sejak 16 April 2018, berangkat dari lomba dan mendapatkan apresiasi baik dengan mendapatkan banyak dorongan untuk mentransformasikan yang hanya berupa ide menjadi kenyataan. Senjani didirikan oleh lima orang dengan disiplin ilmu yang berbeda-beda. Lima orang tersebut di antaranya yaitu: Abdul Latif (*CEO*), Sigit Prasetyo Haq, S.T. (*CMO*), Amira Ulvi Annisa (*COO*), Keffa Shidraty (*CFO*), dan Yafi Alif Ainuddin, A.Md.Kom. (*CTO*).

Pada awal berdiri tidak langsung sebagai platform namun memproduksi dan mengantarkan sendiri mengingat keterbatasan modal finansial maupun non-finansial dari para pendiri. Awal produksi dilakukan di rumah kos salah satu pendiri dengan peralatan seadanya dan melayani pelanggan dengan jumlah dan kawasan yang sangat terbatas hingga kini telah melayani kawasan Kota Malang, Kota Batu, dan Kabupaten Malang. Untuk saat ini layanan utama dari Senjani Kitchen adalah Daily Catering dan Event Catering. Event Catering menyediakan makanan pada acara tertentu dan porsi dalam jumlah banyak. Sedangkan Daily Catering menyediakan dan mengantarkan makanan tanpa ongkos kirim menggunakan Mealbox setiap Pagi, Siang dan Sore setiap hari ke rumah pelanggan.

2.3 Progressive Web App

Progressive Web App adalah teknologi yang digunakan untuk melakukan optimasi pada *website*, sehingga *website* menjadi lebih cepat dan mampu untuk dijalankan layaknya aplikasi *mobile native* pada umumnya (Nurwanto, 2019). *PWA* dapat berjalan layaknya aplikasi *native* karena pengguna dapat memasang pada perangkat mereka. Dikarenakan aplikasi berbasis web maka aplikasi bisa digunakan oleh semua platform baik mobile maupun desktop selama browser pada platform tersebut telah memenuhi syarat.



Gambar 2. 1 Capabilities & Reach PWA

(Sumber: Richard, 2020)

Aplikasi khusus platform atau *native app* memiliki kemampuan untuk mengakses fitur perangkat lebih leluasa seperti sistem *local file*, *port USB*, kamera, hingga *bluetooth*, namun *native app* memiliki kelemahan yaitu jangkauan *platform* yang sangat rendah. Oleh karena itu, *PWA* dipilih karena memiliki jangkauan *platform* yang lebih luas namun memiliki fitur mendekati *native app*. *PWA* memberikan peluang unik untuk menghadirkan pengalaman web yang akan disukai oleh pengguna, karena telah dirancang memiliki 3 pilar yaitu *capable*, *reliable*, dan *installable* (Richard, 2020).

1. *Capable*

Dulu kemampuan untuk mengakses fitur perangkat seperti akses sistem *file*, kontrol media, hingga akses notifikasi seperti *app badging & full clipboard* hanya terbatas untuk aplikasi *native* sesuai dengan *platform*. Namun saat ini dengan kemampuan perkembangan teknologi web yang lebih maju fitur-fitur tersebut sekarang telah dapat dinikmati dan kemampuan itu akan terus berkembang.

2. *Reliable*

Kecepatan sangatlah penting untuk meningkatkan pengalaman pengguna. Faktanya, peningkatan waktu pemuatan halaman dari 1 detik menjadi 10 detik akan meningkatkan *user bouncing* hingga 123%. Pengguna mengharapkan aplikasi tetap bisa digunakan walaupun dengan koneksi jaringan yang lambat, pengguna menyukai aplikasi yang dapat merespon interaksi dalam sekejap. Kecepatan *PWA* bisa terjadi karena perangkat pengguna akan menyimpan *file statis* pada kunjungan pertama sehingga akan meringankan *traffic request response* saat kunjungan setelahnya.

3. *Installable*

PWA dapat dipasang di perangkat dan berjalan dengan jedela mandiri daripada menggunakan *tab browser* seperti web pada umumnya dan juga dapat dijalankan melalui ikon pada layar beranda dan *drawer* aplikasi.

Sehingga pengguna mendapatkan pengalaman lebih baik daripada aplikasi web pada umumnya.

2.4 Sistem Informasi

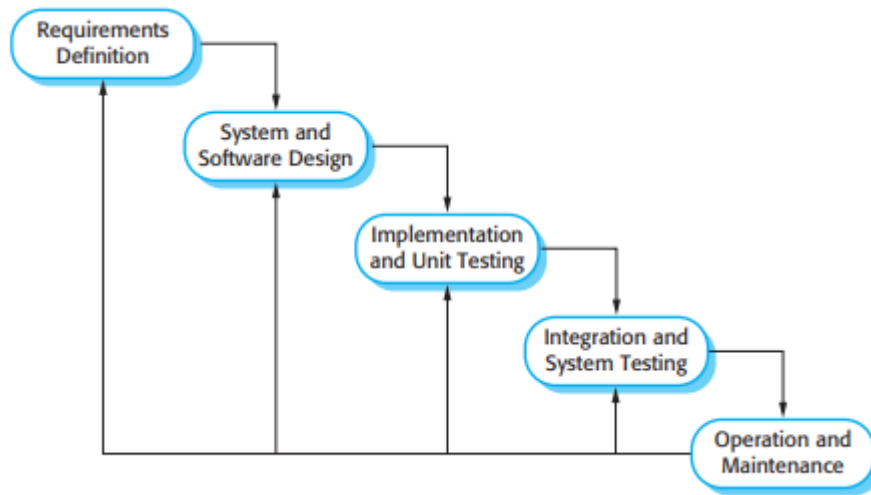
Sistem informasi mencakup manusia, perangkat lunak maupun keras, jaringan dan komunikasi, pengumpulan data, aturan dan prosedur yang dibuat, diterima, diubah, dan disebarluaskan berupa informasi yang diterapkan dalam organisasi (O'Brien & Marakas, 2010). Sistem informasi merupakan himpunan komponen-komponen yang saling terkait satu sama lain untuk melakukan pengumpulan, pemrosesan, penyimpanan, dan pendistribusian informasi. Komponen yang menjadi pembentuk dari sistem informasi adalah input, proses dan output.

2.5 Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC) adalah siklus yang digunakan dalam pembuatan sistem informasi, bertujuan menyelesaikan permasalahan secara efektif. SDLC mempunyai tahapan-tahapan yang bertujuan untuk menghasilkan sistem yang berkualitas tinggi, sesuai dengan keinginan pelanggan, membantu admin dan menyelesaikan permasalahan yang ada. SDLC digunakan agar selama proses pengembangan perangkat lunak memiliki kerangka kerja yang baik dan tepat agar sesuai dengan kebutuhan pengguna (Sommerville, 2011).

2.6 Waterfall

Waterfall merupakan salah satu contoh *Software Development Life Cycle (SDLC)*, metode pengembangan ini memiliki karakteristik pengerjaan harus dilakukan secara berurutan yang dimulai dari tahap penggalian kebutuhan, perancangan, implementasi, pengujian hingga pemeliharaan. Metode *waterfall* menitik beratkan pada tahap awal yaitu analisis kebutuhan dan melakukan pengecekan sebelum melangkah ke tahap selanjutnya (Laudon, 2014). *Waterfall* memiliki tahapan-tahapan, setiap tahapan menghasilkan dokumen sendiri dan tidak dapat melompati tahapan selanjutnya, bila tahapan saat ini belum selesai. Oleh karena itu, metode *waterfall* cocok untuk pengembangan sistem informasi yang telah memiliki permasalahan dan definisi yang jelas.



Gambar 2. 2 Tahapan Metode Waterfall

(Sumber: Sommerville, 2011)

Berdasarkan gambar 2.1 diatas dapat dilihat bahwa metode *Waterfall* merupakan metode yang terstruktur. Tahapan yang sedang dilaksanakan harus terlebih dahulu terselesaikan apabila ingin melanjutkan ke tahapan selanjutnya. *Waterfall* memiliki beberapa tahapan yaitu, *requirements definition*, *system and software design*, *implementation and unit testing*, *integration and system testing*, dan *operation and maintenance*. Dalam penelitian ini proses hanya sampai pada tahap *integration and system testing*.

Menurut Sommerville (2011), *requirements definition* adalah proses pencarian layanan, kendala, dan tujuan sistem melalui wawancara dengan pengguna sistem, kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem. Hasil dari wawancara tersebut didokumentasi dan dianalisis untuk menentukan kebutuhan pengguna, lalu kebutuhan pengguna tersebut dibuat menjadi kebutuhan fungsional dan digambarkan dalam *use case diagram*. *System and and software design* merupakan proses mengalokasikan persyaratan perangkat lunak dan perangkat keras dengan membuat arsitektur sistem secara menyeluruh serta hubungan antar elemen di dalamnya. Pada tahap ini nantinya akan menghasilkan *activity diagram*, *sequence diagram*, *class diagram*, *entity relationship diagram*, dan desain antarmuka. *Implementation and unit testing* merupakan tahap mengubah rancangan dalam tahap sebelumnya menjadi suatu sistem serta memastikan sistem sudah memenuhi spesifikasinya. Tahap ini mengubah bagan rancangan menjadi sebuah produk yang dapat memenuhi kebutuhan pengguna. *Integration and system testing* merupakan proses menghubungkan setiap unit program kemudian diuji sebagai sistem yang lengkap untuk memastikan bahwa persyaratan sistem telah terpenuhi.

2.7 Unified Modelling Language (UML)


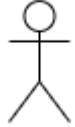


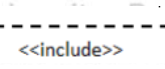
Unified Modelling language (UML) adalah metode perancangan visual untuk merancang sistem berorientasi objek. UML telah menjadi standar dalam

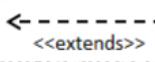
pembuatan *blue print* dari perangkat lunak. UML berfungsi untuk menggambarkan, mendefinisikan perincian, melakukan konstruksi, dan menghasilkan dokumentasi untuk sistem yang akan dibuat (Pressman, 2010). UML memiliki beberapa diagram diantaranya ialah *Use Case Diagram*, *Use Case Specification*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*, *Entity Relationship Diagram*, dan Desain Antarmuka.

2.7.1 Use case diagram

Use Case Diagram adalah diagram yang memodelkan perilaku sistem tampak dari luar. *Use Case Diagram* bertujuan untuk mendeskripsikan kebutuhan dan fungsionalitas dari sistem informasi yang akan dibuat (Grechanik, 2007). Diagram ini memiliki 2 elemen utama, yaitu *actor* dan *use-case*. *Actor* dapat berupa user atau sistem lain sedangkan *use-case* merupakan perilaku sistem yang dilengkapi dengan skenario. Langkah awal dalam membuat diagram ini diawali dengan mengidentifikasi *actor* kemudian dilanjutkan dengan mengidentifikasi *use-case* setiap aktornya. Berikut tabel notasi yang digunakan dalam *Use Case Diagram*.

Tabel 2. 1 Notasi Use Case Diagram

No	Nama Notasi	Deskripsi	Gambar Notasi
1	<i>Use Case</i>	Notasi berbentuk oval, berfungsi menggambarkan fungsionalitas dan sistem. Suatu <i>Use-Case</i> dapat dijalankan oleh beberapa <i>actor</i> .	
2	<i>Actor</i>	Menggambarkan orang atau sistem lain yang digunakan untuk menjadi pemicu aktifnya suatu fungsi. Satu aktor dapat memiliki beberapa <i>use-case</i> . Notasi dapat berupa orang atau sistem luar. Notasi digunakan untuk menjadi pemicu dari aktifnya suatu fungsi. Suatu <i>actor</i> dapat memiliki beberapa <i>use-case</i> .	
3	<i>Asociation</i>	Notasi berbentuk garis tanpa anak panah. Notasi berfungsi menghubungkan <i>Actor</i> dengan <i>Use-Case</i> mengartikan bahwa <i>Actor</i> dapat melakukan suatu fungsi tertentu.	
4	<i>Passive Association</i>	Notasi berbentuk garis yang memiliki tanda panah. Notasi berfungsi menghubungkan <i>Actor</i> dengan <i>Use-Case</i> mengartikan interaksi secara pasif.	
5	<i>Include</i>	Notasi berbentuk garis putus-putus tanpa anak panah. Notasi berfungsi	

		menghubungkan antara <i>Use-Case</i> yang bersifat ketergantungan (<i>required</i>).	
6	<i>Extend</i>	Notasi berbentuk garis putus-putus yang memiliki anak panah. Notasi menghubungkan antara <i>Use-Case</i> dengan <i>Use-Case</i> induknya.	

Sumber: Dennis, et al. (2012)


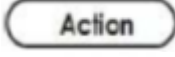

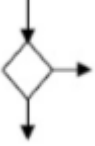
2.7.2 Use Case Specification

Use Case Specification adalah sekumpulan skenario yang dilakukan oleh *Actor* saat berinteraksi dengan sistem. *Use Case Specification* terdapat beberapa bagian, yaitu aktor primer, tujuan, kondisi awal (*pre condition*), alur (*flow*), dan kondisi akhir (*post condition*) (Kurniawan, 2018). Aktor primer adalah aktor yang memulai berjalannya suatu fungsi untuk mencapai tujuan tertentu. Kondisi awal (*pre condition*) adalah kondisi yang harus terpenuhi sebelum skenario dimulai. Alur (*flow*) adalah jalur satu arah dari kondisi awal hingga kondisi akhir, alur memiliki 2 jenis yaitu alur utama dan alur alternatif. Alur utama adalah alur yang tanpa melewati kondisi menyimpang. Sedangkan, alur alternatif adalah alur yang menjadi pilihan jika kondisi tidak memenuhi syarat yang ada. Kondisi akhir (*post condition*) adalah kondisi yang diharapkan ketika fungsi telah selesai dijalankan.

2.7.3 Activity Diagram

Activity Diagram adalah diagram digunakan untuk memodelkan alur kerja dari proses bisnis, langkah-langkah *use-case*, ataupun logika dari perilaku objek yang berupa function (Whitten, 2007). *Activity Diagram* digambarkan dengan tabel berdasarkan aktor yang berperan dalam suatu fungsi. Dalam tabel tersebut dapat digambarkan aktivitas yang dilakukan oleh aktor. Adapun beberapa notasi yang digunakan dalam *Activity Diagram*.

Tabel 2. 2 Notasi *Activity Diagram*

No	Nama Notasi	Deskripsi	Simbol
1	<i>Initial node</i>	Notasi berbentuk lingkaran hitam. Notasi berfungsi untuk penanda titik awal dimana suatu proses dimulai/	
2	<i>Actions</i>	Notasi berbentuk oval. Notasi berfungsi untuk langkah yang dilakukan oleh aktor.	
3	<i>Flow</i>	Notasi panah dengan anak panah. Notasi menunjukkan arah alur dari proses.	
4	<i>Decision</i>	Notasi berbentuk belah ketupat. Notasi menunjukkan proses seleksi kondisi sesuai dengan kondisi yang ada.	

5	<i>Merge</i>	Notasi berbentuk belah ketupat. Notasi menunjukkan pertemuan / penggabungan dari beberapa flow menjadi satu.	
6	<i>Forge</i>	Notasi berbentuk garis datar. Notasi menunjukkan aksi dilakukan secara paralel.	
7	<i>Join</i>	Hasil dari forge yang merupakan aksi secara paralel kemudian disatukan kembali dengan notasi ini. Notasi berbentuk garis datar. Notasi menunjukkan penggabungan aksi menjadi satu dari notasi <i>forge</i> .	
8	<i>Activity Final</i>	Notasi berbentuk lingkaran dengan lingkaran hitam di dalamnya. Notasi berfungsi untuk penanda berakhirnya <i>activity diagram</i> .	

2.7.4 Sequence Diagram

Sequence Diagram adalah diagram memodelkan komunikasi antar objek secara dinamis ketika eksekusi sebuah tugas yang sedang berjalan (Pressman, 2010). *Sequence Diagram* menjelaskan bagaimana interaksi antar aktor, *entity*, *boundary*, dan *control* dalam suatu sistem. Penggunaan diagram bertujuan untuk menjelaskan interaksi antara *actor* dengan objek-objek tersebut dalam sistem. Berikut notasi yang ada dalam *sequence diagram*.

Tabel 2. 3 Notasi *Sequence Diagram*

No.	Nama Notasi	Deskripsi	Simbol
1	<i>Entity</i>	Notasi berbentuk lingkaran dan garis datar. Notasi menggambarkan entitas yang dapat menyimpan data.	
2	<i>Boundary</i>	Notasi menggambarkan halaman yang menghubungkan antara aktor dan sistem	
3	<i>Control</i>	Notasi menggambarkan kontrol aktivitas yang dijalankan oleh sistem.	
4	<i>Message</i>	Notasi menggambarkan arah pengiriman pesan dari suatu objek ke objek lainnya.	
5	<i>Return Values</i>	Notasi menggambarkan arah kembalian setelah pengiriman pesan sebelumnya	

		telah diterima oleh objek dan menghasilkan balasan.	
6	<i>Lifelines</i>	Notasi menggambarkan kehidupan dari suatu objek.	
7	<i>Bar</i>	Notasi menggambarkan masa aktif dari suatu objek (<i>entity, boundary, atau control</i>).	

Sumber : Booch, et.al (2005)

2.7.5 Class Diagram

Class Diagram adalah digram yang memodelkan aspek statis suatu sistem tentang kelas-kelas objek dari sistem, hubungan antar kelas, hingga batasan dari hubungan tersebut (Kim & David, 1999). Diagram ini berfungsi untuk memberikan pandangan mengenai suatu hubungan antara kelas-kelas yang ada di dalam sistem, namun diagram ini tidak menjelaskan kejadian saat kelas berhubungan. Di dalam kelas diagram terdapat relasi *assosiations*, relasi *generalization*, relasi *aggregation*, atribut, *method/behavior* dan *visibility*.

Tabel 2. 4 Notasi *Class Diagram*

No.	Nama Notasi	Deskripsi	Simbol
1	<i>Class</i>	Notasi berisi atribut, <i>method</i> , dan <i>intance</i> dari objek.	
2	<i>Association</i>	Notasi berguna untuk menunjukkan hubungan antar kelas.	
3	<i>Generalization</i>	Notasi berguna untuk menunjukkan relasi pewarisan sifat dari kelas utama ke kelas anak.	
4	<i>Dependency</i>	Notasi menjelaskan ketergantungan kelas tersebut kepada kelas lainnya.	
5	<i>Agregation</i>	Notasi menunjukkan bahwa suatu objek dari suatu kelas terdiri dari objek dari kelas lain	

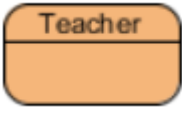
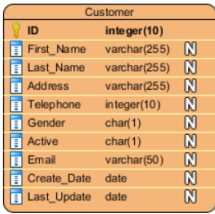


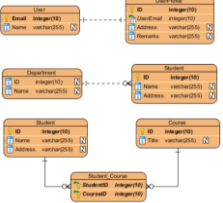
Sumber: Dennis, et al. (2012)

2.7.6 Entity Relationship Diagram

Entity Relationship Diagram (ERD) adalah diagram struktural yang digunakan untuk memodelkan rancangan *database* (Visual Paradigm, 2021). ERD

berisi simbol dan konektor yang memvisualisasikan 2 informasi penting, yaitu entitas utama dalam cakupan sistem dan antar hubungan entitas yang ada. ERD memiliki beberapa notasi dalam penggambarannya, yaitu *entity*, *entity attributes*, *primary key*, *foreign key*, dan *relation*.

Tabel 2. 5 Notasi Entity Relationship Diagram

No.	Nama Notasi	Deskripsi	Simbol
1	<i>Entity</i>	<i>Entity</i> , hal atau konsep yang data didefinisikan dalam suatu sistem. Contoh: Orang (Mahasiswa), Objek (<i>invoice</i>), konsep (profil), dll.	
2	<i>Entity attributes</i>	<i>Entity Attributes</i> disebut juga sebagai kolom dari <i>entity</i> . Hal ini memiliki nama yang mendeskripsikan properti dan tipe yang mendeskripsikan jenis atributnya.	
3	<i>Primary key</i>	<i>Primary Key</i> adalah atribut entitas unik yang mendefinisikan data dalam tabel database dan tidak ada 2 PK yang memiliki nilai yang sama.	
4	<i>Foreign Key</i>	<i>Foreign Key</i> adalah referensi kunci utama dari tabel lain digunakan mengidentifikasi hubungan antar tabel.	
5	<i>Relation</i>	<i>Relation</i> adalah hubungan antara 2 entitas. Memiliki beberapa jenis, yaitu <i>one to one</i> , <i>one to many</i> , dan <i>many to many</i> .	

Sumber: Visual Paradigm (2021)

2.7.7 Desain Antarmuka

Desain antarmuka merupakan rancangan bagi mesin, peralatan, computer, perangkat *mobile*, *software*, dan situs web yang berfokus pada pengalaman pengguna. Desain antarmuka memiliki tujuan agar interaksi pengguna sederhana dan seefisien mungkin sehingga mencapai tujuan pengguna. Untuk meningkatkan peluang keberhasilan dan angka kepuasan *user*, antarmuka pengguna harus memiliki prinsip desain antarmuka pengguna, yaitu: perspektif dari user, inovasi, produk yang jelas, mengandung nilai-nilai estetika desain, desain yang konsisten dan familiar, dan elemen visual yang kuat (Oliver, 2020). Pada penelitian ini, proses desain antarmuka menggunakan metode *wireframe*.

2.8 Business Process Model and Notation (BPMN)

BPMN adalah standar dalam perancangan sebuah proses bisnis, rancangan ini menggunakan notasi-notasi yang telah ditentukan sehingga dapat dimengerti oleh semua pengguna bisnis (Object Management Group, 2013). Dengan merancang proses bisnis perusahaan, pengguna bisnis dapat memahami bagaimana proses yang terjadi pada perusahaan tersebut.

Tabel 2. 6 Tabel Notasi BPMN

No	Nama Notasi	Deskripsi	Simbol
1	Start	Sebagai titik awal proses bisnis dimulai	
2	End	Sebagai titik akhir proses bisnis diakhiri	
3	Send Task	Menjelaskan aktivitas pengiriman pesan ke aktor lainnya	
4	Receive Task	Menjelaskan aktivitas penerimaan pesan dari aktor lainnya	
5	Manual Task	Menjelaskan aktivitas yang dilakukan secara manual atau tanpa aplikasi	
6	User Task	Menjelaskan aktivitas yang dilakukan menggunakan aplikasi / sistem	
7	Sequence flow	Menandakan urutan aktivitas dalam proses bisnis	
8	Message flow	Menandakan aliran pesan antara 2 aktor yang berbeda	
9	Association	Menandakan keterkaitan artefak dan informasi dengan elemen grafis BPMN	
10	Data Input	Menjelaskan bahwa data menjadi syarat untuk melaksanakan aktivitas	
11	Exclusive gateway	Menjelaskan percabangan aktivitas selanjutnya tergantung pada kondisi yang dipilih	
12	Pool	Menjelaskan tempat aktivitas-aktivitas yang dilakukan oleh aktor	

Sumber: Object Management Group (2013)

2.9 Wireframe

Wireframe merupakan desain aplikasi dan *website* pada level struktural dengan kata lain sebagai kerangka gambar. Mendesain sebuah *wireframe* bermaksud untuk merencanakan kerangka utama sebuah *website* atau aplikasi yang ingin dibuat. Dengan *Wireframe*, perancang dapat merancang struktur halaman yang baik, *layout*, arsitektur informasi, *user flow*, fungsionalitas, dan perilaku yang diinginkan dari pengguna. *Wireframe* dapat dilakukan secara sederhana menggunakan gambar manual menggunakan tangan maupun dengan *software* khusus seperti Figma, Adobe XD, dan Framer. Melakukan tahap *wireframe* memiliki keuntungan, yaitu memastikan konsep fokus pada *user*, memperjelas fitur, dan cepat dan murah (Rahmalia, 2021).

2.10 Codeigniter framework

Codeigniter adalah kerangka kerja (*framework*) PHP yang digunakan dalam pengembangan *web*. Codeigniter memiliki ukuran yang sangat kecil, dibuat untuk pengembangan yang membutuhkan perangkat sederhana dan elegan untuk membuat aplikasi *web* berfitur lengkap (Codeigniter, 2021). *Framework* ini memiliki kelebihan, yaitu ukuran yang kecil, performa yang stabil, menggunakan konsep MVC, memiliki sistem keamanan dengan perlindungan CSRF dan XSS, dokumentasi yang lengkap, dan struktur file yang rapi. Kerangka kerja ini memanfaatkan pola MVC (*model, view, controller*). *Model* adalah kumpulan file berfungsi untuk untuk memanipulasi database. *View* adalah bagian yang ditampilkan kepada user. *Controller* adalah kumpulan file yang berfungsi sebagai logika *website* dan penghubung antara *model* dan *view*.

2.11 React Js library

React Js merupakan *library* bahasa pemrograman JavaScript dibuat oleh Facebook sejak 2013. React Js memiliki kemampuan *composable user interface* berarti *developer* dapat membagi *user interface* menjadi komponen-komponen, sehingga komponen tersebut bisa digunakan kembali pada *user interface* lainnya (React, 2021). React Js bersifat open source sehingga dapat digunakan dan disesuaikan oleh siapapun. Fitur-fitur yang dimiliki oleh React Js, yaitu:

1. Declarative

React Js memiliki kemampuan untuk membuat *user interface* menjadi lebih interaktif karena memiliki konsep *state*, sehingga lebih mudah untuk dibuat dari pada HTML, CSS, dan JS *native*. *Declarative* juga dapat membuat kode lebih mudah untuk dibaca, diprediksi, dan lebih mudah untuk di *debug*.

2. Component – Based

React Js memiliki kemampuan untuk membangun sebuah *user interface* berawal dari kumpulan komponen yang dibuat secara terpisah. Sehingga

developer dapat menggunakan komponen tersebut pada *user interface* lainnya.

3. *Learn Once, Write Anywhere*

Salah satu cabang fitur yang sedang dikembangkan adalah React Native, berfungsi bila *developer* ingin mengembangkan aplikasi *mobile* secara *native* dengan menggunakan *syntax* yang sama dalam pengembangan React Js.

2.12 Pengujian

Pengujian adalah tahap akhir dalam pengembangan suatu perangkat lunak. Pengujian dilakukan bertujuan untuk mengukur kualitas dari perangkat lunak yang dibuat sebelum masuk ke dalam proses *deployment*. Pengujian dilakukan untuk memastikan semua yang telah dibuat dapat berjalan semestinya dan mencapai tujuan dengan performa yang baik. Terdapat beberapa jenis testing perangkat lunak yang dapat dilakukan, untuk penelitian ini penulis menggunakan dua pengujian, yaitu *validation testing* dan *system usability scale*.

2.12.1 *Validation Testing*

Validation testing adalah pengujian untuk mencari tahu apakah sistem berjalan sesuai harapan atau tidak dengan melakukan beberapa kasus uji (Sommerville, 2011). Pengujian dilakukan dengan cara memasukkan *input* pada sistem lalu menguji apakah *output* yang dikeluarkan oleh sistem sesuai dengan harapan atau tidak. Apabila *output* yang dihasilkan oleh sistem tidak sesuai dengan harapan, maka sistem tersebut masih memiliki kekurangan yang harus diperbaiki.

2.12.2 *System Usability Scale (SUS)*

SUS adalah sebuah cara yang digunakan peneliti untuk melakukan evaluasi produk atau layanan perangkat keras, perangkat lunak, perangkat seluler, situs web, serta aplikasi. *System Usability Scale (SUS)* adalah alat yang handal untuk mengukur tingkat *usability* (Usability.gov, 2021). Keuntungan dalam menggunakan pengujian ini yaitu sebagai berikut:

- a. Mudah dalam pengimplementasiannya
- b. Jumlah sampel yang dibutuhkan kecil dan hasilnya masih dapat diandalkan
- c. Pengujian SUS yang valid dapat digunakan untuk menilai sebuah sistem layak digunakan atau tidak layak untuk digunakan.

Strongly disagree					Strongly agree
1	2	3	4	5	

Gambar 2. 3 System Usability Scale Answer

Pada pengujian SUS, pengguna akan diberikan 10 pernyataan dalam Gambar 2. 4 dan diminta untuk menilai dari 1 hingga 5. Nilai 1 menyatakan pengguna sangat tidak setuju dengan pernyataan tersebut dan nilai 5 menyatakan pengguna sangat setuju dengan pernyataan tersebut. Berikut merupakan 10 pernyataan patokan dari *system usability scale* yang akan disampaikan dalam pengujian:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

Gambar 2. 4 System Usability Scale Question

Setelah pengguna memberi nilai terhadap 10 pernyataan di atas, maka dapat dilakukan penghitungan *SUS Score* dengan ketentuan sebagai berikut:

- a. Untuk pertanyaan bernomor ganjil, nilai didapat dari mengurangi angka dengan 1 atau $X-1$.
- b. Untuk pertanyaan bernomor genap, nilai didapat dari mengurangi 5 dengan angka atau $5-X$.
- c. Selanjutnya, cari total nilai dengan menjumlahkan seluruh hasil dari pertanyaan bernomor ganjil dan genap. Kemudian kalikan hasil penjumlahan tersebut dengan 2,5.

Setelah dilakukan penghitungan, skor SUS di atas 68 akan dianggap di atas rata-rata dan di bawah 68 dianggap di bawah rata-rata (Usability.gov, 2021). Akan tetapi terdapat cara lebih baik dalam melakukan penilaian terhadap hasil penghitungan tersebut, yaitu dengan melihat pada tabel di bawah ini:

SUS Score	Grade	Adjective Rating
> 80.3	A	Excellent
68 – 80.3	B	Good
68	C	Okay
51 – 68	D	Poor
< 51	F	Awful

Gambar 2. 5 System Usability Scale Rating



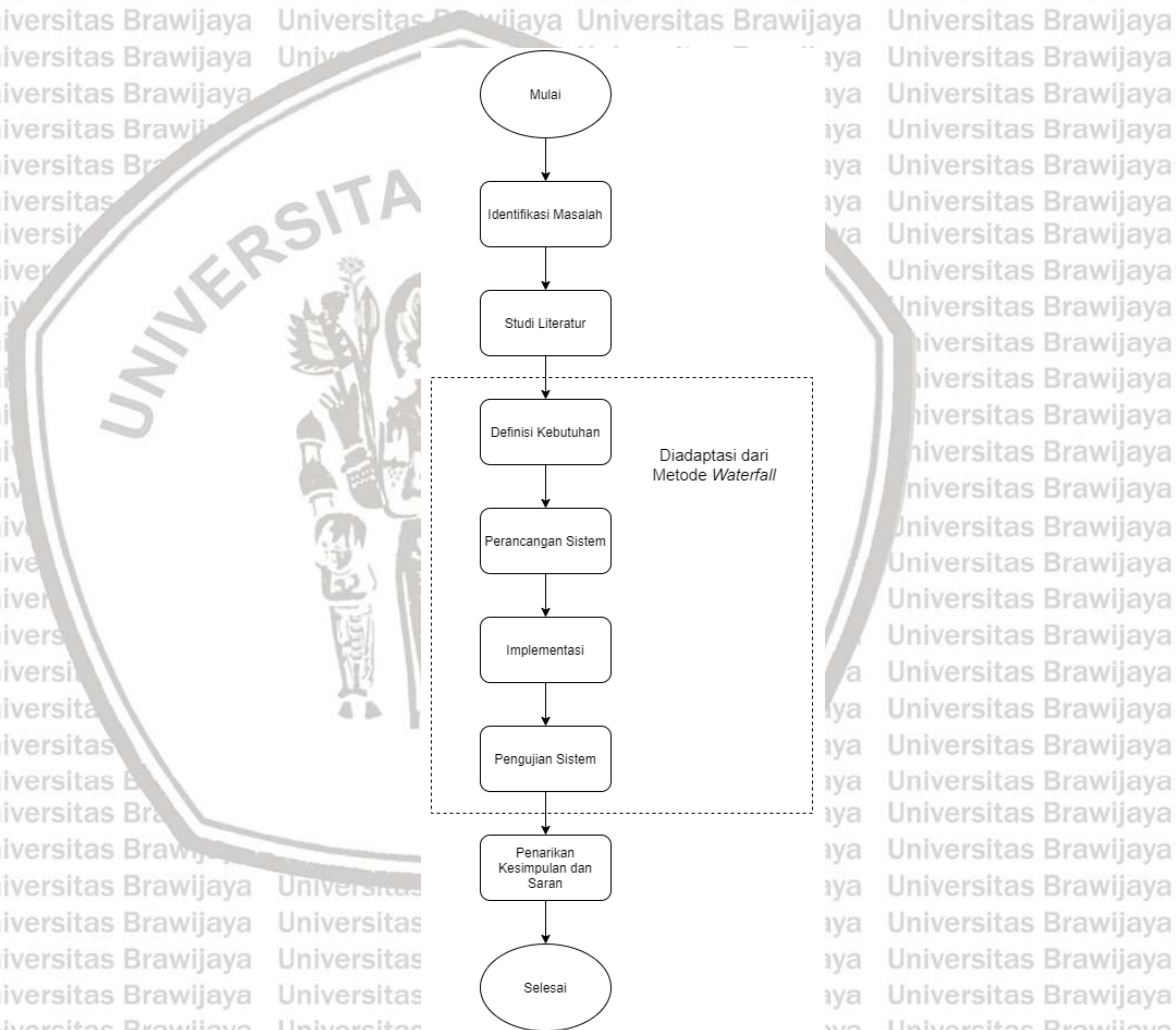
BAB 3 METODOLOGI PENELITIAN

3.1 Tipe Penelitian

Penelitian ini merupakan penelitian implementatif pengembangan (*development*) sistem informasi.

3.2 Tahapan Pelaksanaan

Tahapan pada penelitian ini menggunakan tahap-tahap dalam *system development life cycle* (SDLC) mengadaptasi dari metode *waterfall*. Tahap-tahap penelitian bisa dilihat pada gambar 3.1.



Gambar 3. 1 Tahapan Penelitian

Penelitian ini menggunakan tahapan-tahapan *Waterfall* yang dapat dilihat dalam Gambar 2. 2. Namun karena batasan masalah, maka tahapan *Waterfall* yang penulis gunakan hanya definisi kebutuhan, perancangan sistem, implementasi, dan pengujian sistem.

3.3 Identifikasi Masalah

Langkah pertama dalam penelitian ini adalah identifikasi masalah. Identifikasi masalah dilakukan untuk mengetahui permasalahan ada pada Senjani Kitchen dengan cara melakukan wawancara dengan CEO Sejani Kitchen yaitu Abdul Latif. Setelah melakukan wawancara dengan CEO Sejani Kitchen, diperoleh masalah sebagai berikut.

1. Sistem pemesanan Daily Catering yang sekarang masih kurang efektif dikarenakan masih menggunakan aplikasi *chat* seperti Whatsapp, pelanggan harus menunggu balasan dari admin.
2. Pencatatan pemesanan Daily Catering saat ini yang dilakukan oleh admin masih manual, sehingga sering terjadi kerugian baik fatal maupun tidak fatal bagi perusahaan.

3.4 Studi Literatur

Tahap studi literatur tahap dimana penulis mencari referensi yang berfungsi sebagai dasar teori untuk penulisan penelitian ini. Referensi berasal dari buku, jurnal, laporan penelitian sejenis, artikel dan *website*. Teori-teori yang penulis gunakan adalah:

1. Tinjauan Pustaka
2. Senjani Kitchen
3. Sistem Informasi
4. Metode *Waterfall*
5. *Unified Modelling Language*
6. *Business Process Model and Notation*
7. *Codeigniter framework*
8. *Progressive Web App*
9. *React Js library*
10. *System Usability Scale*

3.5 Definisi Kebutuhan

Definisi kebutuhan merupakan tahap pertama dalam metode *waterfall*. Dalam menentukan kebutuhan pengguna dibutuhkan pengumpulan data. Penelitian ini menggunakan metode wawancara sebagai cara pengumpulan data, wawancara dilakukan dengan CEO Senjani Kitchen Abdul Latif, lalu melakukan analisis proses bisnis yang terjadi di Senjani Kitchen saat ini dan digambarkan dalam *Business Proces Model and Notation (BPMN)*. Selanjutnya, dilakukan analisis kebutuhan untuk mengetahui siapa saja pengguna yang terlibat dengan

sistem. Setelah itu, kebutuhan sistem yang akan dijadikan fitur pada sistem informasi yang baru dimodelkan dalam *use case diagram* dan *use case specification*.

3.6 Perancangan Sistem

Perancangan sistem dilakukan berdasarkan *use case* yang telah dibuat ditahap sebelumnya. *Use case* tersebut dianalisis untuk menentukan fitur yang dapat mempermudah pemesanan. Fitur-fitur tersebut dimodelkan dalam *activity diagram*, *sequence diagram*, *class diagram*, *entity relationship diagram*, dan desain antarmuka. Dokumen-dokumen tersebut menjadi dasar dalam implementasi ditahap selanjutnya.

3.7 Implementasi Sistem

Pada tahap implementasi, rancangan sistem yang sudah dibuat diterjemahkan ke dalam bentuk kode menggunakan alat pengembangan aplikasi web yang telah ditentukan. Pengembangan *frontend* menggunakan React Js *library*, implementasi *library* ini dibangun menggunakan bahasa JSX, lalu menghasilkan *build* berisi HTML, CSS, Javascript yang akan digunakan *browser* pada sisi client. Sedangkan, untuk pengembangan *backend* menggunakan CodeIgniter *framework*, implementasi *framework* ini dibangun menggunakan bahasa PHP yang akan digunakan pada sisi server. Pengembangan ini menggunakan pola *Model View Controller* (MVC), CodeIgniter *framework* berperan sebagai *Model* dan *Controller* dan React Js *library* bertugas sebagai *View*. *Model* bertugas untuk mengatur, menyiapkan, memanipulasi dan mengorganisasikan data dari *database*, sedangkan *Controller* bertugas untuk mengatur apa yang harus dilakukan oleh *Model* dan mengelola data yang diperlukan oleh *View*. *View* bertugas untuk menyajikan informasi dan berinteraksi dengan pengguna. Pengembangan ini juga menggunakan optimasi web yaitu *Progressive Web App* yang memiliki kemampuan untuk menyimpan data *statis* ke *cache* perangkat sehingga kecepatan akses meningkat dan dapat dipasang di perangkat.

3.8 Pengujian Sistem

Pada tahap ini, sistem yang sudah selesai dibuat diuji menggunakan 2 metode pengujian, yaitu:

1. *Validation Testing* atau pengujian validasi: pengujian ini dilakukan oleh penulis sendiri sebagai penguji menggunakan skenario uji yang sebelumnya telah dibuat. Pengujian ini dilakukan untuk mencari tahu apakah sistem yang telah dibuat mengeluarkan hasil yang diharapkan atau tidak.
2. *System Usability Scale (SUS)*: pengujian ini dilakukan oleh calon pengguna untuk mengetahui tingkat *usability*. Bertujuan untuk melakukan evaluasi terhadap hasil implementasi yang telah dibuat.

3.9 Kesimpulan dan Saran

Penarikan kesimpulan dan saran merupakan proses terakhir dalam penelitian. Kesimpulan dibuat berdasarkan pembahasan yang dilakukan dalam penelitian ini dan menjawab semua pertanyaan dalam rumusan masalah. Penulisan saran bertujuan untuk penelitian selanjutnya agar memperbaiki kekurangan yang ada dan untuk menyempurnakan pengembangan sistem informasi selanjutnya.



BAB 4 REKAYASA PERSYARATAN / KEBUTUHAN

4.1 Definisi Kebutuhan

Definisi kebutuhan adalah proses untuk memahami kebutuhan pengguna. Hasil dokumen dari proses ini adalah *Business Proces Model and Notation (BPMN)*, *use case diagram*, dan *use case spesification*.

4.1.1 Analisis Permasalahan

Senjani Kitchen adalah perusahaan rintisan yang bergerak dibidang pangan. Dari hasil wawancara, Senjani Kitchen telah melayani kawasan Kota Malang, Kota Batu, dan Kabupaten Malang dengan sekitar 100 pelanggan aktif dalam layanan Daily Catering. Proses bisnis Daily Catering saat ini, yaitu pelanggan melakukan chat ke admin Senjani Kitchen via Whatsapp untuk berlanggan Daily Catering. Lalu, pelanggan memilih salah satu jenis paket kupon yaitu Basic, Reusable, Deluxe, atau Family. Selanjutnya, admin akan memberikan Form Online Link sesuai dengan jenis paket yang dipilih. Berikutnya, pelanggan mengisi Form Online tersebut sesuai dengan kebutuhan seperti jumlah kupon yang dapat ditukarkan dengan makanan, jenis nasi, harga, alamat pengiriman, alergi makanan bila ada, dan waktu kapan saja makanan dikirimkan. Kemudian, pelanggan melakukan pembayaran dan mengirimkan bukti pembayaran ke admin Senjani Kitchen lewat Whatsapp. Setelah data pesanan dan bukti pembayaran lengkap, admin akan mengirim tabel data pesanan ke staf dapur dan staf pengiriman untuk diproduksi dan diantarkan sesuai dengan waktu pemesanan.

Selama ini pemesanan Daily Catering oleh pelanggan melalui aplikasi chat Whatsapp menggunakan *smartphone* mereka, sehingga pelanggan terlalu bergantung kepada balasan admin. Karena keterbatasan waktu admin tidak bisa menjawab dengan cepat. Pelanggan seharusnya dapat mendapatkan informasi seperti jadwal makanan, paket kupon, harga, dan memesan waktu pemesanan lebih fleksibel tanpa harus menunggu balasan dari admin. Dikarenakan pencatatan pemesanan masih manual sering terjadi kesalahan pencatatan. Sehingga mengakibatkan kerugian mulai yang tidak fatal seperti pakai sambal atau tidak, hingga yang fatal seperti tercatat minta dikirim namun ternyata tidak sehingga Senjani Kitchen harus menanggung kerugian atas makanan tersebut.

Adapun dalam proses pencarian solusi media sistem informasi yang akan dikembangkan, penulis mempertimbangkan yaitu kondisi perusahaan dan kondisi pelanggan yang telah memakai layanan. Dari sisi perusahaan, Senjani kitchen merupakan perusahaan rintisan yang masih dalam tahap awal dimana perusahaan masih memprioritaskan kegunaan finansial pada *core business* tidak untuk pengembangan IT sehingga sangat kecil kemungkinan untuk bisa melakukan peluncuran aplikasi diberbagai *platform*. Sedangkan dari sisi pelanggan, pelanggan telah terbiasa untuk melakukan proses pembelian menggunakan *smartphone* mereka.

Untuk menjawab permasalahan dan kondisi yang ada saat ini, penulis mencari alternatif solusi media sistem informasi. Terdapat beberapa jenis media sistem informasi, seperti media sosial, aplikasi *mobile*, aplikasi *website*, aplikasi *desktop*, dan lain-lain. Salah satu kelebihan dari aplikasi *website* adalah mudah diakses disemua perangkat selama perangkat tersebut memiliki browser (aplikasi *multi-platform*) dikarenakan pengguna tidak perlu melakukan instalasi sehingga dapat menghemat storage perangkat pengguna. Sedangkan perusahaan hanya perlu mengeluarkan biaya pada aplikasi web saja sehingga dapat menghemat pengeluaran perusahaan (Arviana, 2021). Saat ini perusahaan teknologi yang mengandalkan internet sebagai lahan bisnis utamanya meluncurkan semua aplikasi di berbagai platform agar mendapatkan pengguna sebanyak-banyaknya (Akbar, 2019). Penggunaan aplikasi web sebagai media sistem informasi untuk Senjani Kitchen merupakan pilihan yang paling cocok dengan keterbatasan sumber daya yang dimiliki, Senjani kitchen cukup mengembangkan aplikasi web untuk menghemat pengeluaran perusahaan.

Salah satu teknologi pengembangan lebih lanjut dari aplikasi web adalah *Progressive Web App*. *Progressive Web App* adalah teknologi yang digunakan untuk melakukan optimasi pada *website*, sehingga *website* menjadi lebih cepat dan mampu untuk dijalankan layaknya aplikasi *mobile native* pada umumnya (Kurniawan, 2019). Keuntungan lainnya dari *Progressive Web App*, yaitu contoh aplikasi Twitter mencatat konsumsi data *PWA* hanya 0,6 MB dibandingkan dengan aplikasi android 23 MB dan ios 100MB dan memiliki *service worker* yang bertugas menyimpan file web sehingga saat pengguna kembali lagi tidak perlu melakukan pengunduhan kembali (Akbar, 2019). Menurut wawancara, pelanggan menggunakan *smartphone* mereka untuk melakukan aktivitas Daily Catering, seperti pemesanan, melihat jadwal makanan, dan lainnya. Oleh karena itu, *PWA* dipilih dalam penelitian ini dan menambahkan opsi pelanggan untuk bisa meng-*install* aplikasi web pemesanan Daily Catering layaknya aplikasi *native*.

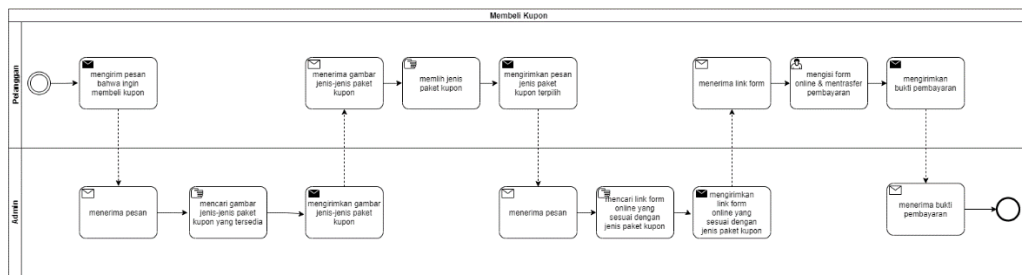
4.1.2 Pemodelan Proses Bisnis

Untuk mempermudah memahami proses bisnis pada penelitian ini, proses bisnis digambarkan dengan pemodelan BPMN. *Business Process Model and Notation (BPMN)* adalah standar dalam perancangan sebuah proses bisnis, rancangan ini menggunakan notasi-notasi yang telah ditentukan sehingga dapat dimengerti oleh semua pengguna bisnis (Object Management Group, 2013). Dalam sistem pemesanan Daily Catering Senjani Kitchen terdapat 2 proses bisnis utama, yaitu pembelian kupon dan pemesanan kupon.

4.1.2.1 Pembelian Kupon

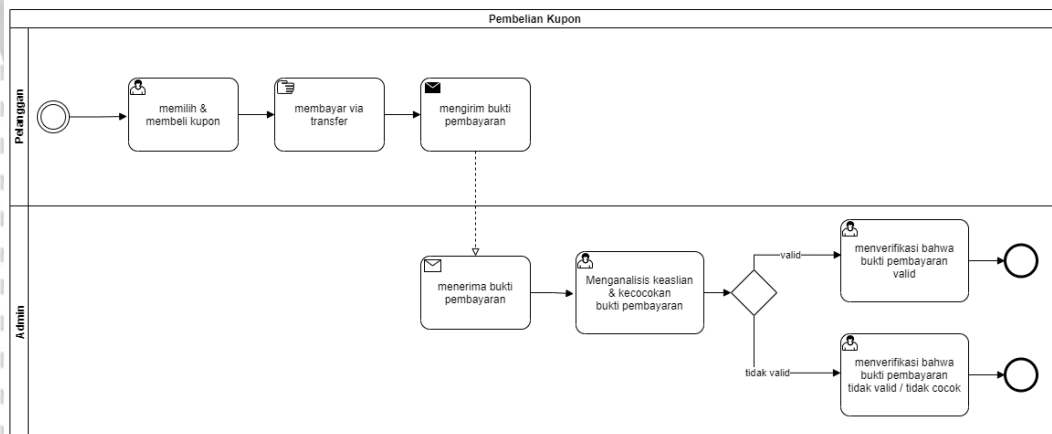
Pembelian kupon hanya bisa dilakukan oleh pelanggan saja. Membeli kupon berfungsi agar kupon pelanggan dapat ditukarkan dengan menu sesuai dengan jadwalnya. Pembayaran di Senjani Kitchen dapat dilakukan dengan

berbagai metode, yaitu Gopay, Dana, Jenius (BTPN), atau BNI. Pengecekan bukti pembayaran dilakukan oleh admin.



Gambar 4. 1 BPMN As-Is Pembelian Kupon

Dalam Gambar 4. 1 menjelaskan proses bisnis dari pembelian kupon *as-is*, proses bisnis ini bertujuan agar pelanggan dapat membeli kupon. Pelanggan harus menghubungi admin melalui pesan singkat *Whatsapp*, memberi tahu bahwa ingin membeli kupon. Admin menerima pesan tersebut dan memberikan informasi jenis-jenis paket kupon yang tersedia. Setelah itu, pelanggan memilih salah satu jenis paket kupon yang tersedia dengan cara mengirimkan pesan ke admin. Selanjutnya, admin mengirimkan link *online form* sesuai dengan pilihan pelanggan. Pelanggan mengisi *online form* tersebut yang berisi biodata pelanggan dan memilih kupon beserta harganya, kemudian pelanggan melakukan pembayaran sesuai dengan metode pembayaran. Lalu, pelanggan mengirimkan bukti pembayaran tersebut ke admin dan akhirnya kupon pembelian pelanggan tersebut dicatat oleh admin.



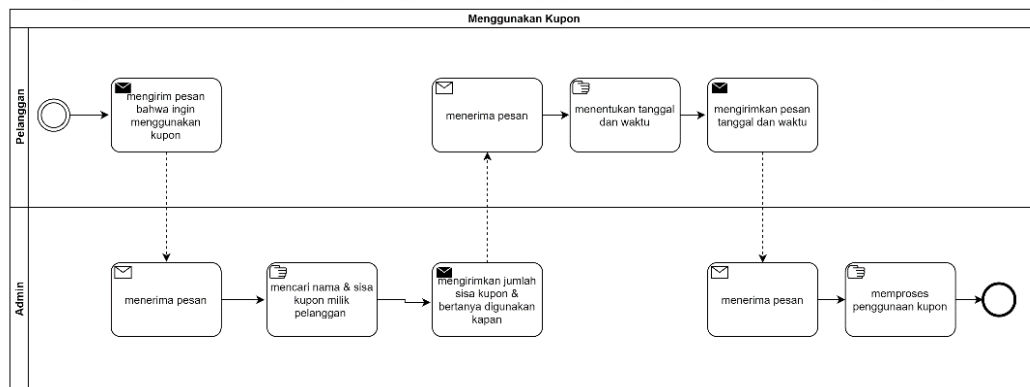
Gambar 4. 2 BPMN To-Be Pembelian Kupon

Dalam Gambar 4. 2 menjelaskan proses bisnis pembelian kupon *to-be*, proses ini bertujuan agar pelanggan mendapatkan kupon yang nanti bisa ditukar dengan menu. Syarat untuk bisa melakukan proses ini, pelanggan harus mempunyai akun dan telah masuk ke dalam sistem. Awalnya pelanggan memilih kupon yang ingin dibeli, lalu melakukan pembayaran seharga paket kupon yang dibeli dengan metode pembayaran yang disediakan. Selanjutnya pelanggan mengunggah bukti pembayaran melalui aplikasi web dan diterima oleh admin.

Setelah admin mendapatkan bukti pembayaran, admin menganalisis apakah bukti pembayaran (harga dan metode pembayaran) tersebut asli dan cocok dengan keterangan pembelian. Jika cocok dan asli maka admin akan menyatakan valid sehingga kupon dapat digunakan, sedangkan jika tidak cocok atau palsu admin akan menyatakan tidak valid sehingga kupon tidak dapat digunakan.

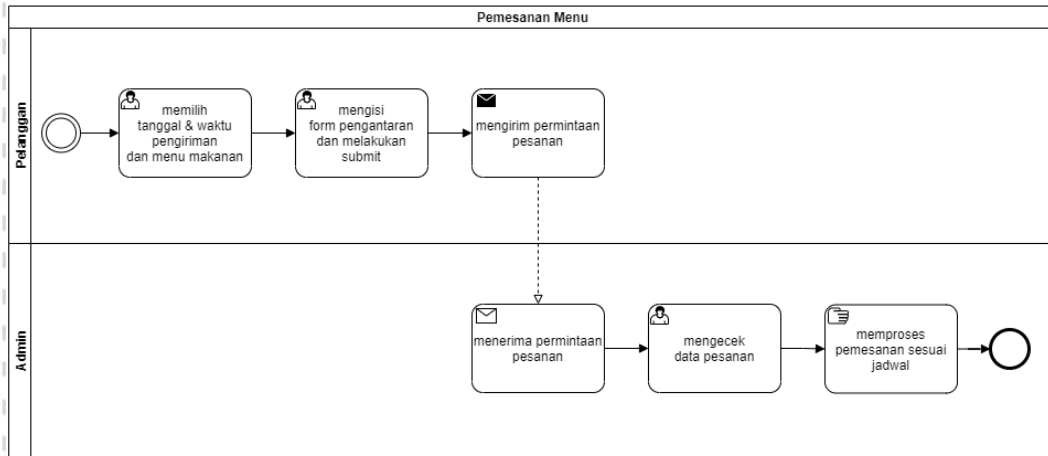
4.1.2.2 Pemesanan Menu

Pemesanan menu atau penggunaan kupon hanya bisa dilakukan oleh pelanggan saja. Menggunakan kupon berfungsi agar pelanggan dapat menukarkan kupon yang telah dibeli sebelumnya dengan menu sesuai dengan jadwalnya.



Gambar 4. 3 BPMN As-Is Pemesanan Menu

Dalam Gambar 4. 3 menjelaskan proses bisnis *as-is* dari pemesanan menu atau penggunaan kupon, tujuan dari proses ini agar pelanggan dapat menggunakan kupon dengan menu yang telah disediakan. Awalnya, pelanggan harus mengirim pesan singkat *Whatsapp* ke admin Senjani Kitchen bahwa ingin memesan menu dengan kupon yang dimilikinya. Setelah admin mendapatkan pesan tersebut, admin mencari secara manual jumlah kupon yang dimiliki pelanggan dan mengirimkan informasi tersebut ke pelanggan. Setelah pelanggan mengetahui jumlah kupon yang dimilikinya, pelanggan menentukan tanggal dan waktu pengiriman. Akhirnya, admin mencatat pemesanan tersebut.



Gambar 4. 4 BPMN To-Be Pemesanan Menu

Dalam Gambar 4. 4 menjelaskan proses bisnis *to-be* dari pemesanan menu atau penggunaan kupon, proses ini memiliki tujuan agar pelanggan dapat menukarkan kupon yang telah dia beli dengan menu. Syarat agar pelanggan dapat melakukan proses ini, yaitu telah memiliki akun, telah login, dan memiliki kupon yang cukup untuk memesan. Pelanggan yang telah masuk sistem memilih fitur gunakan kupon. Lalu, pelanggan memilih menu-menu yang berisi informasi nama makanan dan tanggal & waktu pengiriman, selanjutnya pelanggan mengisi *form* pengantaran agar pengiriman lancar. Setelah pelanggan melakukan pemesanan dan admin mendapatkan informasi pemesanan, admin melakukan pengecekan dan memproses pemesanan sesuai dengan tanggal dan waktu pemesanan.

4.1.3 Analisis Kebutuhan

Analisis kebutuhan adalah proses untuk aktor yang terlibat dan kemampuan yang harus dimiliki sistem untuk memenuhi kebutuhan pengguna. Analisis kebutuhan terdiri dari identifikasi pengguna, identifikasi kebutuhan fungsional, identifikasi kebutuhan non fungsional, *use case diagram*, dan *use case specification*.

4.1.3.1 Identifikasi Pengguna

Identifikasi pengguna adalah proses untuk mencari tahu siapa saja aktor yang terlibat dengan sistem dan kemampuan secara umum yang bisa dilakukan aktor terhadap sistem. Hasil identifikasi pengguna digunakan untuk pembuatan *use case diagram*. Hasil identifikasi dapat dilihat pada Tabel 4. 1.

Tabel 4. 1 Tabel Identifikasi Pengguna

Kode Aktor	Nama Pengguna	Keterangan
A-1	Pelanggan	Pelanggan merupakan aktor melakukan pemesanan Daily Catering

A-2	Admin	Admin merupakan aktor mendapatkan informasi hasil pemesanan Daily Catering
-----	-------	--

4.1.3.2 Identifikasi Kebutuhan Fungsional

Identifikasi kebutuhan fungsional merupakan kebutuhan yang harus diimplementasikan pada sistem. Kebutuhan ini didasarkan dari hasil identifikasi pengguna yang diperlukan oleh sistem.

Tabel 4. 2 Tabel Identifikasi Kebutuhan Fungsional

Kode Aktor	Kode Kebutuhan Fungsional	Nama Kebutuhan Fungsional	Keterangan
A-1	KF-01	Pendaftaran	Sistem dapat menambahkan pelanggan baru
	KF-02	Masuk	Sistem dapat membolehkan pelanggan masuk ke dalam sistem
	KF-03	Keluar	Sistem dapat mengeluarkan pelanggan dari sistem
	KF-04	Melihat jadwal menu	Sistem dapat menampilkan jadwal menu
	KF-05	Membeli Kupon	Sistem dapat memproses pembelian kupon berdasarkan paket kupon yang dipilih
	KF-06	Menggunakan kupon	Sistem dapat memproses penukaran kupon dengan menu dengan kupon yang digunakan dan menu yang dipilih
	KF-07	Melihat riwayat pesanan	Sistem dapat menampilkan semua pesanan yang telah dilakukan dan akan dilakukan
	KF-08	Melihat riwayat kupon	Sistem dapat menampilkan semua kupon yang telah digunakan dan belum digunakan
A-2	KF-09	Masuk	Sistem dapat memasukkan admin ke dalam sistem

KF-10	Keluar	Sistem dapat mengeluarkan admin dari sistem
KF-11	Verifikasi bukti pembayaran	Sistem dapat memproses admin untuk melakukan verifikasi/ memeriksa bukti pembayaran yang telah diunggah oleh pelanggan
KF-12	Membuat jadwal menu baru	Sistem dapat memproses pembuatan data baru pada jadwal menu
KF-13	Mengedit jadwal menu	Sistem dapat memproses penyuntingan pada jadwal menu
KF-14	Mengedit paket kupon	Sistem dapat memproses penyuntingan pada paket kupon
KF-15	Melihat hasil pesanan	Sistem dapat menampilkan kepada admin hasil pesanan yang telah dipesan oleh pelanggan
KF-16	Mengedit pesanan	Sistem dapat memproses penyuntingan pada pesanan
KF-17	Melihat pelanggan	Sistem dapat menampilkan informasi pelanggan kepada admin
KF-18	Mengedit kupon pelanggan	Sistem dapat memproses penyuntingan pada kupon yang dimiliki oleh pelanggan

4.1.3.3 Identifikasi Kebutuhan Non Fungsional

Pada tahap ini akan didefinisikan kebutuhan non-fungsional yang akan diimplementasikan dalam sistem. Kebutuhan ini didasarkan dari identifikasi pengguna.

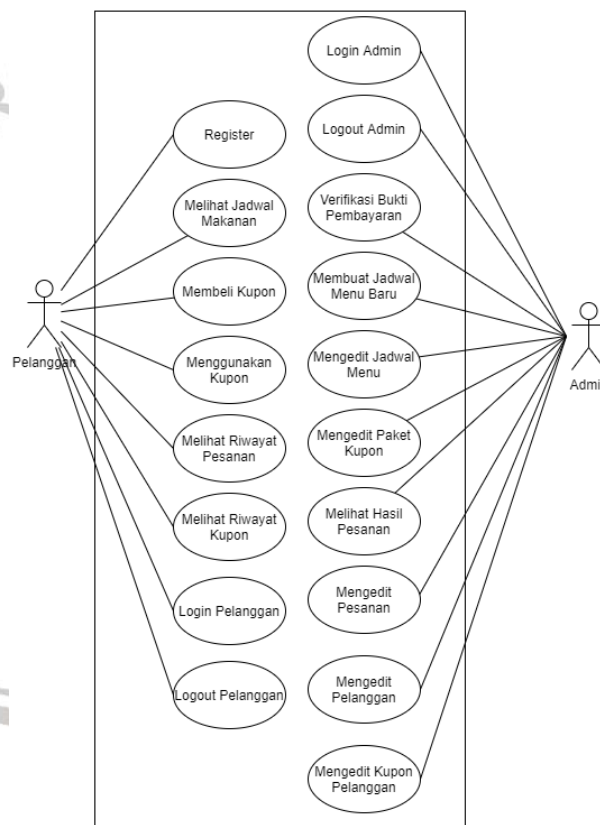
Tabel 4. 3 Tabel Identifikasi Non Fungsional

Kode Aktor	Kode Kebutuhan Non Fungsional	Nama Kebutuhan Non Fungsional	Keterangan
A-1	KNF-01	Usability	Sistem mudah untuk digunakan dan memiliki nilai

A-2	KNF-02	Usability	usability lebih dari sama dengan 68
			Sistem mudah untuk digunakan dan memiliki nilai usability lebih dari sama dengan 68

4.1.3.4 Use Case Diagram

Hasil dari proses identifikasi kebutuhan pada tahapan sebelumnya menjadi dasar dari *use case diagram* yang dibuat. *Use case diagram* dibuat agar mempermudah untuk memodelkan kemampuan sistem. Terdapat 2 aktor dalam *use case diagram* yaitu pelanggan dan admin yang memiliki kemampuan yang berbeda.



Gambar 4. 5 Use Case Diagram

Dalam Gambar 4. 5 menjelaskan *use case diagram* yang memiliki 2 aktor yaitu pelanggan dan admin. Sistem dapat memfasilitasi pelanggan dalam beberapa aktivitas, yaitu *register*, *login*, *logout*, melihat jadwal menu, membeli kupon, menggunakan kupon, melihat riwayat kupon, dan melihat riwayat pesanan. Sistem juga dapat memfasilitasi admin dalam beberapa aktivitas, yaitu *login*, *logout*, verifikasi bukti pembayaran, membuat jadwal menu baru, mengedit

jadwal menu, mengedit paket kupon, melihat hasil pesanan, mengedit pesanan, mengedit pelanggan, dan mengedit kupon pelanggan.

4.1.3.5 Use Case Specification

Use case specification dibuat untuk menjelaskan lebih rinci dari *use case diagram* yang telah dibuat sebelumnya.

A. Register untuk Pelanggan

Untuk melihat setiap aktivitas yang terjadi pada *use case register*, penjelasan terdapat pada Tabel 4. 4.

Tabel 4. 4 Tabel Use Case Specification Register

<i>Use Case Name</i>	<i>Register</i>
<i>Description</i>	Sistem dapat melakukan menambahkan akun pelanggan baru berdasarkan <i>email</i> yang <i>unique</i>
<i>Actor</i>	Pelanggan
<i>Pre-Condition</i>	Aktor belum terdaftar dalam sistem sebelumnya
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengakses sistem 2. Sistem menampilkan halaman <i>register</i> 3. Aktor memasukkan <i>email</i>, <i>password</i>, nama lengkap, dan biodata lainnya 4. Sistem akan menverifikasi akun, menyimpan pada database, dan menampilkan notifikasi sukses mendaftar
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. <i>Email</i> telah terdaftar 2. Sistem menampilkan notifikasi bahwa <i>email</i> telah terdaftar
<i>Post Condition</i>	Aktor masuk ke halaman utama

B. Login untuk Pelanggan

Untuk melihat setiap aktivitas yang terjadi pada *use case login*, penjelasan terdapat pada Tabel 4. 5.

Tabel 4. 5 Tabel Use Case Specification Login untuk Pelanggan

<i>Use Case Name</i>	<i>Register</i>
<i>Description</i>	Sistem dapat melakukan autentifikasi pengguna agar sesuai dengan hak akses pengguna terhadap sistem
<i>Actor</i>	Pelanggan
<i>Pre-Condition</i>	Actor telah terdaftar sebagai pelanggan maupun admin
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengakses sistem 2. Sistem menampilkan halaman <i>login</i> 3. Aktor memasukkan <i>email</i> dan <i>password</i> 4. Sistem melakukan pengecekan aotentifikasi dan otorisasi
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. <i>Email</i> dan <i>password</i> tidak cocok 2. Sistem akan menampilkan notifikasi <i>email</i> dan/atau <i>password</i> tidak cocok
<i>Post Condition</i>	Aktor masuk ke dalam halaman utama

C. *Logout* untuk Pelanggan

Untuk melihat setiap aktivitas yang terjadi pada *use case logout*, penjelasan terdapat pada Tabel 4. 6.

Tabel 4. 6 Tabel Use Case Specification Logout untuk Pelanggan

<i>Use Case Name</i>	<i>Logout</i>
<i>Description</i>	Sistem dapat mengeluarkan pengguna agar dapat mengakhiri kebutuhan aktor dalam sistem
<i>Actor</i>	Pelanggan
<i>Pre-Condition</i>	Membuka halaman utama
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih tombol <i>logout</i> 2. Sistem akan menghapus hasil otentifikasi aktor sebagai aktor yang sedang aktif 3. Sistem menampilkan halaman <i>login</i>
<i>Alternative Flow</i>	-

Post Condition Aktor masuk ke dalam halaman login

D. Melihat Jadwal Menu untuk Pelanggan

Untuk melihat setiap aktivitas yang terjadi pada *use case* jadwal menu, penjelasan terdapat pada Tabel 4. 7.

Tabel 4. 7 Use Case Specification Melihat Jadwal Menu

<i>Use Case Name</i>	Melihat Jadwal Menu
<i>Description</i>	Aktor dapat melihat jadwal makanan berdasarkan urutan tanggal
<i>Actor</i>	Pelanggan
<i>Pre-Condition</i>	Aktor telah masuk dan berada di halaman utama
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih tombol lihat jadwal menu 2. Sistem menampilkan jenis paket kupon 3. Aktor memilih berdasarkan jenis paket kupon 4. Sistem menampilkan jadwal menu berdasarkan paket kupon yang dipilih oleh aktor
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Aktor melihat jadwal menu

E. Membeli Kupon untuk Pelanggan

Untuk melihat setiap aktivitas yang terjadi pada *use case* membeli kupon, penjelasan terdapat pada Tabel 4. 8.

Tabel 4. 8 Use Case Spesification Membeli Kupon

<i>Use Case Name</i>	Membeli Kupon
<i>Description</i>	Sistem dapat melayani proses aktor dalam melihat paket kupon, memilih pembayaran, hingga pembayaran paket kupon
<i>Actor</i>	Pelanggan
<i>Pre-Condition</i>	Aktor telah masuk ke halaman utama
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih salah satu jenis paket kupon yang disediakan

	<ol style="list-style-type: none"> 2. Sistem menampilkan daftar paket kupon dengan spesifikasinya yaitu harga, jumlah kupon, dan jenis nasi. 3. Aktor memilih salah satu paket kupon 4. Sistem menampilkan pilihan metode pembayaran 5. Aktor memilih salah satu dari metode pembayaran 6. Sistem menampilkan tenggat waktu pembayaran dan <i>form</i> unggah bukti pembayaran 7. Aktor mengirim bukti pembayaran 8. Sistem menampilkan notifikasi bukti pembayaran berhasil diupload dan pembelian paket kupon sedang diproses
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Aktor salah mengunggah file yang bukan gambar 2. Sistem akan menampilkan informasi bahwa file bukti pembayaran harus bertipe gambar
<i>Post Condition</i>	Aktor akan melihat notifikasi pembelian kupon sedang diproses

F. Menggunakan Kupon untuk Pelanggan
 Untuk melihat setiap aktivitas yang terjadi pada *use case* menggunakan kupon, penjelasan terdapat pada Tabel 4. 9.

<i>Use Case Name</i>	Menggunakan kupon
<i>Description</i>	Sistem dapat melayani proses aktor untuk menggunakan kupon yang telah dibeli, menukarkan dengan menu berdasarkan jadwal menu yang tersedia
<i>Actor</i>	Pelanggan
<i>Pre-Condition</i>	Aktor sudah <i>login</i> dan berada pada halaman riwayat kupon
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih tombol gunakan kupon 2. Sistem menampilkan jadwal menu berdasarkan kupon 3. Aktor memilih menu yang disediakan 4. Sistem menampilkan <i>form</i> detail pengantaran 5. Aktor mengisi <i>form</i> pengantaran 6. Sistem menampilkan pesanan sukses dan sedang diproses
<i>Alternative Flow</i>	1. Aktor tidak mengisi <i>form</i> pengantaran

	2. Sistem menampilkan informasi <i>form</i> pengantaran harus diisi
<i>Post Condition</i>	Aktor mendapatkan informasi pesanan sedang diproses

Tabel 4. 9 Use Case Spesification Menggunakan Kupon

G. Melihat Riwayat Pesanan untuk Pelanggan

Untuk melihat setiap aktivitas yang terjadi pada *use case* riwayat pesanan, penjelasan terdapat pada Tabel 4. 10.

Tabel 4. 10 Use Case Spesification Melihat Riwayat Pesanan

<i>Use Case Name</i>	Melihat riwayat pesanan
<i>Description</i>	Sistem dapat melayani aktor untuk melihat riwayat pesanan baik masih diproses ataupun sudah selesai
<i>Actor</i>	Pelanggan
<i>Pre-Condition</i>	Aktor sudah <i>login</i> dan berada di halaman utama
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih tombol riwayat pesanan 2. Sistem menampilkan pesanan yang akan datang dan pesanan yang telah selesai
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Aktor mendapatkan informasi pesanan

H. Melihat Riwayat Kupon untuk Pelanggan

Untuk melihat setiap aktivitas yang terjadi pada *use case* riwayat kupon, penjelasan terdapat pada Tabel 4. 11.

Tabel 4. 11 Use Case Spesification Melihat Riwayat Kupon

<i>Use Case Name</i>	Melihat riwayat kupon
<i>Description</i>	Sistem dapat melayani aktor untuk melihat riwayat kupon baik masih bisa digunakan maupun sudah tidak dapat digunakan
<i>Actor</i>	Pelanggan
<i>Pre-Condition</i>	Aktor sudah <i>login</i> dan berada di halaman utama
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih tombol riwayat kupon

2. Sistem menampilkan kupon yang dapat digunakan dan sudah tidak dapat digunakan

<i>Alternative Flow</i>	-
<i>Post Condition</i>	Aktor mendapatkan informasi kupon yang dimiliki

I. *Login* untuk Admin

Untuk melihat setiap aktivitas yang terjadi pada *use case login*, penjelasan terdapat pada Tabel 4. 12.

Tabel 4. 12 Use Case Spesification Login untuk Admin

<i>Use Case Name</i>	<i>Register</i>
<i>Description</i>	Sistem dapat melakukan autentifikasi pengguna agar sesuai dengan hak akses pengguna terhadap sistem
<i>Actor</i>	Admin
<i>Pre-Condition</i>	Actor telah terdaftar sebagai pelanggan maupun admin
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengakses sistem 2. Sistem menampilkan halaman <i>login</i> 3. Aktor memasukkan <i>email</i> dan <i>password</i> 4. Sistem melakukan pengecekan aautentifikasi dan otorisasi
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. <i>Email</i> dan <i>password</i> tidak cocok 2. Sistem akan menampilkan notifikasi <i>email</i> dan/atau <i>password</i> tidak cocok
<i>Post Condition</i>	Aktor masuk ke dalam halaman utama

J. *Logout* untuk Admin

Untuk melihat setiap aktivitas yang terjadi pada *use case logout*, penjelasan terdapat pada Tabel 4. 13.

Tabel 4. 13 Use Case Spesification Logout untuk Admin

<i>Use Case Name</i>	<i>Logout</i>
----------------------	---------------

<i>Description</i>	Sistem dapat mengeluarkan pengguna agar dapat mengakhiri kebutuhan aktor dalam sistem
<i>Actor</i>	Admin
<i>Pre-Condition</i>	Membuka halaman utama
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih tombol <i>logout</i> 2. Sistem akan menghapus hasil otentifikasi aktor sebagai aktor yang sedang aktif 3. Sistem menampilkan halaman <i>login</i>
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Aktor masuk ke dalam halaman login

K. Verifikasi Bukti Pembayaran untuk Admin

Untuk melihat setiap aktivitas yang terjadi pada *use case* riwayat kupon, penjelasan terdapat pada Tabel 4. 14.

Tabel 4. 14 Use Case Spesification Verifikasi Bukti Pembayaran

<i>Use Case Name</i>	Verifikasi bukti pembayaran
<i>Description</i>	Sistem dapat melayani aktor untuk menverifikasi apakah bukti pembayaran asli atau tidak
<i>Actor</i>	Admin
<i>Pre-Condition</i>	Aktor sudah <i>login</i> berada di halaman utama
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih tombol bukti pembayaran 2. Sistem menampilkan data bukti pembayaran yang telah diunggah oleh pelanggan 3. Aktor memilih salah satu bukti pembayaran milik pelanggan 4. Sistem menampilkan gambar bukti pembayaran 5. Aktor menverifikasi bukti pembayaran 6. Sistem menampilkan notifikasi bukti pembayaran berhasil diverifikasi
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Aktor tidak menverifikasi bukti pembayaran

2. Sistem menampilkan notifikasi bukti pembayaran bahwa bukti pembayaran tidak sesuai

<i>Post Condition</i>	Aktor mendapatkan informasi verifikasi pembayaran berhasil
-----------------------	--

L. Membuat Jadwal Menu Baru untuk Admin

Untuk melihat setiap aktivitas yang terjadi pada *use case* jadwal menu baru, penjelasan terdapat pada Tabel 4. 15.

Tabel 4. 15 Use Case Spesification Membuat Jadwal Menu Baru

<i>Use Case Name</i>	Membuat jadwal menu baru
<i>Description</i>	Sistem dapat melayani proses aktor dalam menambahkan jadwal menu baru
<i>Actor</i>	Admin
<i>Pre-Condition</i>	Aktor sudah <i>login</i> berada di halaman jadwal menu
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih tombol tambah menu 2. Sistem menampilkan <i>form</i> menu 3. Aktor mengisi <i>form</i> menu 4. Sistem menyimpan menu baru dan menampilkan notifikasi menu baru berhasil ditambahkan
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Aktor mendapatkan informasi bahwa menu baru telah ditambahkan

M. Mengedit Jadwal Menu untuk Admin

Untuk melihat setiap aktivitas yang terjadi pada *use case* jadwal menu baru, penjelasan terdapat pada Tabel 4. 16.

Tabel 4. 16 Use Case Spesification Mengedit Jadwal Menu

<i>Use Case Name</i>	Mengedit Jadwal Menu
<i>Description</i>	Sistem dapat melayani aktor dalam mengedit data jadwal menu yang telah dibuat sebelumnya
<i>Actor</i>	Admin
<i>Pre-Condition</i>	Aktor sudah <i>login</i> berada di halaman utama

<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih tombol jadwal menu 2. Sistem menampilkan tabel data jadwal menu 3. Aktor memilih baris yang ingin diedit 4. Sistem menampilkan <i>form</i> edit 5. Aktor mengubah data <i>form</i> 6. Aktor menekan tombol simpan 7. Sistem menyimpan perubahan dan menampilkan notifikasi data berhasil diubah
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol batal 2. Sistem membatalkan penyimpanan
<i>Post Condition</i>	Aktor mendapatkan informasi bahwa data berhasil diubah

N. Mengedit Paket kupon untuk Admin

Untuk melihat setiap aktivitas yang terjadi pada *use case* paket kupon, penjelasan terdapat pada Tabel 4. 17.

Tabel 4. 17 Use Case Spesification Mengedit Paket kupon

<i>Use Case Name</i>	Mengedit paket kupon
<i>Description</i>	Sistem dapat melayani aktor dalam mengedit data paket kupon yang telah ada sebelumnya
<i>Actor</i>	Admin
<i>Pre-Condition</i>	Aktor sudah <i>login</i> berada di halaman utama
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih tombol paket kupon 2. Sistem menampilkan tabel data paket kupon 3. Aktor memilih baris yang ingin diedit 4. Sistem menampilkan <i>form</i> edit 5. Aktor mengubah data <i>form</i> 6. Aktor menekan tombol simpan 7. Sistem menyimpan perubahan dan menampilkan notifikasi data berhasil diubah
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol batal 2. Sistem membatalkan penyimpanan

Post Condition Aktor mendapatkan informasi bahwa data berhasil diubah

O. Melihat Hasil Pesanan untuk Admin

Untuk melihat setiap aktivitas yang terjadi pada *use case* hasil pesanan, penjelasan terdapat pada Tabel 4. 18.

Tabel 4. 18 Use Case Spesification Melihat Hasil Pesanan

<i>Use Case Name</i>	Melihat hasil pesanan
<i>Description</i>	Sistem dapat melayani aktor untuk melihat hasil pesanan yang telah dipesan oleh pelanggan
<i>Actor</i>	Admin
<i>Pre-Condition</i>	Aktor sudah <i>login</i> berada di halaman utama
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih tombol pesanan 2. Sistem menampilkan tabel data pesanan
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Aktor mendapatkan informasi data pesanan

P. Mengedit Pesanan untuk Admin

Untuk melihat setiap aktivitas yang terjadi pada *use case* mengedit pesanan, penjelasan terdapat pada Tabel 4. 19.

Tabel 4. 19 Use Case Spesification Mengedit Pesanan

<i>Use Case Name</i>	Mengedit pesanan
<i>Description</i>	Sistem dapat melayani aktor dalam mengedit data pesanan yang telah dibuat oleh pelanggan
<i>Actor</i>	Admin
<i>Pre-Condition</i>	Aktor sudah <i>login</i> berada di halaman utama
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih tombol pesanan 2. Sistem menampilkan tabel data pesanan 3. Aktor memilih baris yang ingin diedit 4. Sistem menampilkan <i>form</i> edit 5. Aktor mengubah data <i>form</i> 6. Aktor menekan tombol simpan

	7. Sistem menyimpan perubahan dan menampilkan notifikasi data berhasil diubah
<i>Alternative Flow</i>	1. Aktor menekan tombol batal 2. Sistem membatalkan penyimpanan
<i>Post Condition</i>	Aktor mendapatkan informasi bahwa data berhasil diubah

Q. Mengedit Pelanggan untuk Admin

Untuk melihat setiap aktivitas yang terjadi pada *use case* pelanggan, penjelasan terdapat pada Tabel 4. 20.

Tabel 4. 20 Use Case Spesification Mengedit Pelanggan

<i>Use Case Name</i>	Mengedit pelanggan
<i>Description</i>	Sistem dapat melayani aktor dalam mengedit data pelanggan yang diisi oleh pelanggan saat pendaftaran
<i>Actor</i>	Admin
<i>Pre-Condition</i>	Aktor sudah <i>login</i> berada di halaman utama
<i>Basic Flow</i>	1. Aktor memilih tombol pelanggan 2. Sistem menampilkan tabel data pelanggan 3. Aktor memilih baris yang ingin diedit 4. Sistem menampilkan <i>form</i> edit 5. Aktor mengubah data <i>form</i> 6. Aktor menekan tombol simpan 7. Sistem menyimpan perubahan dan menampilkan notifikasi data berhasil diubah
<i>Alternative Flow</i>	1. Aktor menekan tombol batal 2. Sistem membatalkan penyimpanan
<i>Post Condition</i>	Aktor mendapatkan informasi bahwa data berhasil diubah

R. Mengedit Kupon Pelanggan untuk Admin

Untuk melihat setiap aktivitas yang terjadi pada *use case* kupon pelanggan, penjelasan terdapat pada Tabel 4. 21.

Tabel 4. 21 Use Case Spesification Mengedit Kupon Pelanggan

<i>Use Case Name</i>	Mengedit kupon pelanggan
<i>Description</i>	Sistem dapat melayani aktor dalam mengedit data kupon pelanggan yang telah dibeli oleh pelanggan
<i>Actor</i>	Admin
<i>Pre-Condition</i>	Aktor sudah <i>login</i> berada di halaman utama
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih tombol pelanggan 2. Sistem menampilkan tabel data pelanggan 3. Aktor memilih data pelanggan yang ingin diubah kuponnya 4. Sistem menampilkan kupon yang dimiliki oleh pelanggan tersebut 5. Aktor memilih kupon yang ingin diedit 6. Sistem menampilkan <i>form</i> edit 7. Aktor mengubah data <i>form</i> 8. Aktor menekan tombol simpan 9. Sistem menyimpan perubahan dan menampilkan notifikasi data berhasil diubah
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol batal 2. Sistem membatalkan penyimpanan
<i>Post Condition</i>	Aktor mendapatkan informasi bahwa data berhasil diubah

BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan Sistem

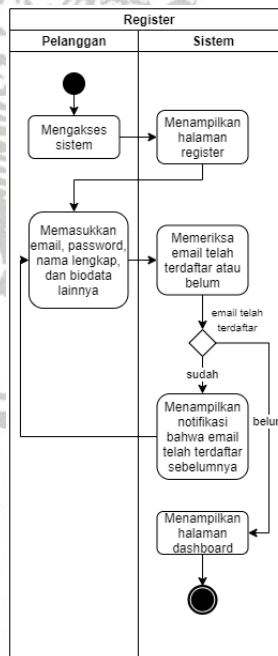
Perancangan sistem dilakukan berdasarkan 18 *use case* yang telah dibuat ditahap sebelumnya. 18 *Use case* tersebut dianalisis untuk menentukan fitur yang dapat mempermudah pemesanan. Fitur-fitur tersebut dimodelkan dalam *activity diagram*, *sequence diagram*, *class diagram*, *entity relationship diagram*, dan desain antarmuka. Dokumen-dokumen tersebut menjadi dasar dalam implementasi ditahap selanjutnya.

5.1.1 Activity Diagram

Activity Diagram adalah diagram digunakan untuk memodelkan alur kerja dari proses bisnis, langkah-langkah *use-case*, ataupun logika dari perilaku objek yang berupa function (Whitten, 2007). *Activity Diagram* digambarkan dengan tabel berdasarkan aktor yang berperan dalam suatu fungsi. Dalam tabel tersebut dapat digambarkan aktivitas yang dilakukan oleh aktor.

5.1.1.1 Register untuk Pelanggan

Gambar 5. 1 menjelaskan alur aktivitas *register* hanya bisa dilakukan oleh pelanggan saja. *Register* berfungsi untuk menambahkan data akun pelanggan baru. Pendaftaran akun pelanggan baru hanya bisa sukses bila *email* yang digunakan belum pernah terdaftar sebelumnya.



Gambar 5. 1 Activity Diagram Register

5.1.1.2 Login untuk Pelanggan

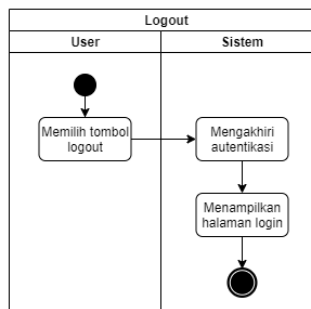
Gambar 5. 2 menjelaskan alur aktivitas *login* dapat dilakukan oleh pelanggan. *Login* berfungsi agar aktor dapat masuk kedalam sistem menggunakan *email* dan *password* yang telah terdaftar sebelumnya. Ada 2 alur, alur pertama dilakukan dengan *email* dan *password* yang telah terdaftar dan benar, sedangkan alur kedua menggunakan *email* dan *password* yang salah.



Gambar 5. 2 Activity Diagram Login untuk Pelanggan

5.1.1.3 Logout untuk Pelanggan

Gambar 5. 3 menjelaskan alur aktivitas *user* pelanggan dalam melakukan *logout* agar bisa menghentikan autentikasi dan keluar dari sistem.

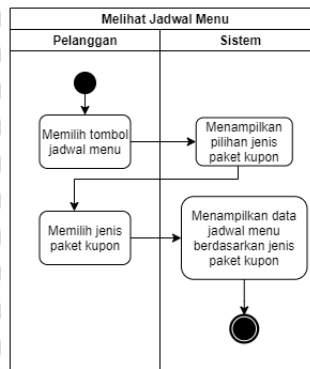


Gambar 5. 3 Activity Diagram Logout untuk Pelanggan

5.1.1.4 Melihat Jadwal Menu untuk Pelanggan

Gambar 5. 4 menjelaskan aktivitas melihat jadwal menu hanya dapat dilakukan oleh pelanggan saja. Melihat jadwal menu berfungsi agar

pelanggan mendapatkan informasi spesifikasi menu, yaitu jadwal dan waktu menu dikirimkan, nama menu, dan keterangan menu.

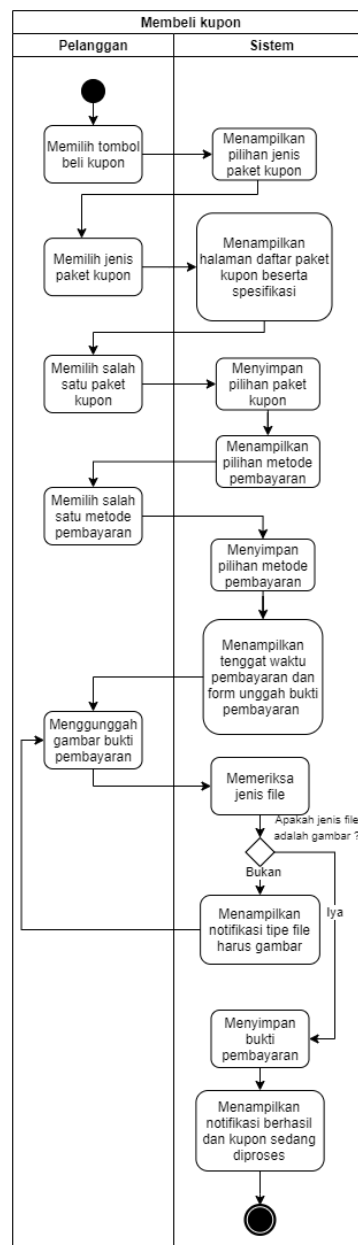
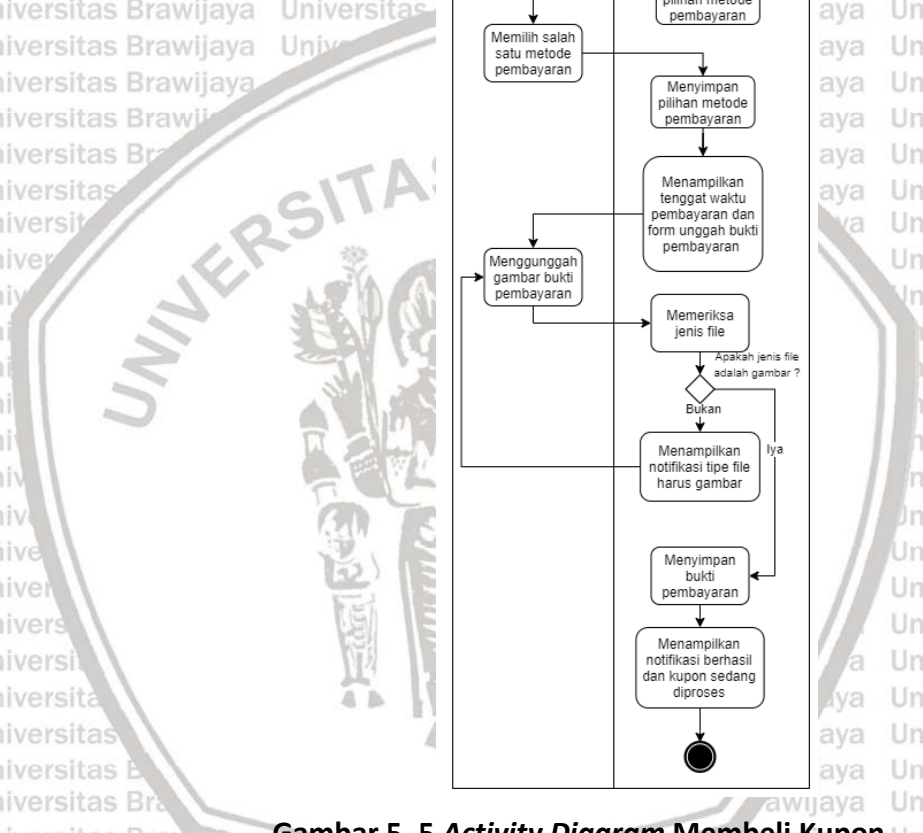


Gambar 5. 4 Activity Diagram Melihat Jadwal Menu

5.1.1.5 Membeli Kupon untuk Pelanggan

Gambar 5. 5 menjelaskan aktivitas membeli kupon hanya bisa dilakukan oleh pelanggan saja. Membeli kupon berfungsi agar pelanggan dapat membeli kupon yang nantinya akan ditukarkan dengan menu sesuai dengan jadwalnya. Ada 2 alur, alur pertama pelanggan menggugah file yang diinginkan yaitu file gambar sedangkan alur kedua pelanggan menggugah file bukan berbentuk gambar.

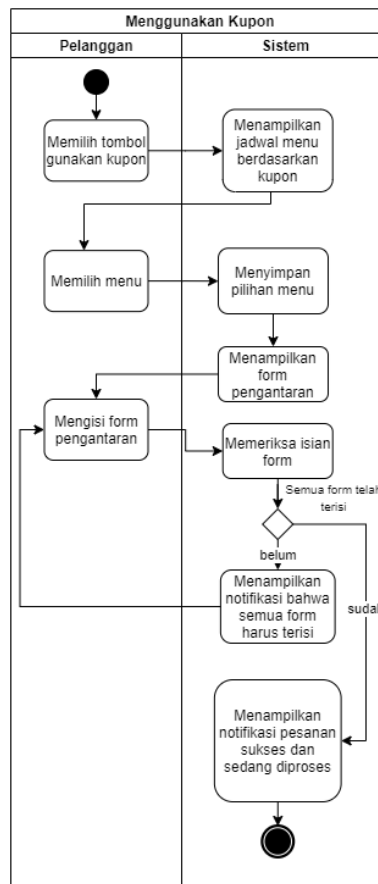




Gambar 5. 5 Activity Diagram Membeli Kupon

5.1.1.6 Menggunakan Kupon untuk Pelanggan

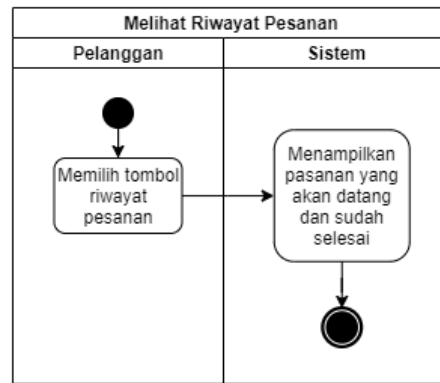
Gambar 5. 6 menjelaskan alur aktivitas menggunakan kupon hanya bisa dilakukan oleh pelanggan saja. Menggunakan kupon berfungsi agar pelanggan dapat menukarkan kupon yang telah dibeli sebelumnya dengan menu makanan. Ada 2 alur, alur pertama dilakukan pelanggan setelah pelanggan mengisi semua *form* yang ada, sedangkan alur kedua pelanggan tidak mengisi semua *form*.



Gambar 5. 6 Activity Diagram Menggunakan Kupon

5.1.1.7 Melihat Riwayat Pesanan untuk Pelanggan

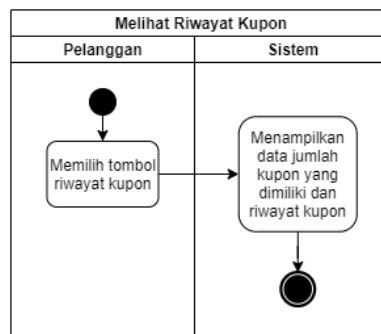
Gambar 5. 7 menjelaskan alur aktivitas melihat riwayat pesanan hanya dapat dilakukan oleh pelanggan saja. Melihat riwayat pesanan milik pelanggan itu sendiri berfungsi agar pelanggan mendapatkan informasi pesanan yang akan terjadi ataupun yang telah terjadi informasi yang didapatkan pelanggan, yaitu tanggal & waktu pengiriman, nama menu, keterangan menu, dan status pesanan.



Gambar 5. 7 Activity Diagram Melihat Riwayat Pesanan

5.1.1.8 Melihat Riwayat Kupon untuk Pelanggan

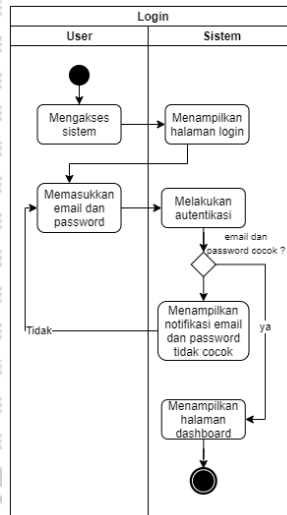
Gambar 5. 8 menjelaskan aktivitas melihat riwayat kupon hanya dapat dilakukan oleh pelanggan saja. Melihat riwayat kupon milik pelanggan itu sendiri berfungsi agar pelanggan mendapatkan informasi kupon yang dimilikinya beserta spesifikasinya (jumlah kupon tersisa, jenis paket kupon, tanggal pembelian, tanggal kedaluwarsa, dan status kupon).



Gambar 5. 8 Activity Diagram Melihat Riwayat Kupon

5.1.1.9 Login untuk Admin

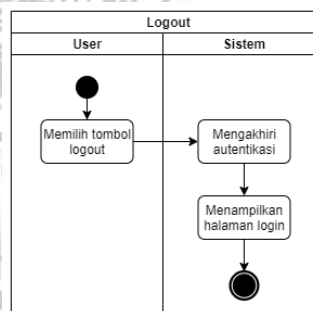
Gambar 5. 9 menjelaskan alur aktivitas *login* dapat dilakukan oleh Admin. *Login* berfungsi aktor dapat masuk kedalam sistem menggunakan *email* dan *password* yang telah terdaftar sebelumnya. Ada 2 alur, alur pertama dilakukan dengan *email* dan *password* yang telah terdaftar dan benar, sedangkan alur kedua menggunakan *email* dan *password* yang salah.



Gambar 5. 9 Activity Diagram Login untuk Admin

5.1.1.10 Logout untuk Admin

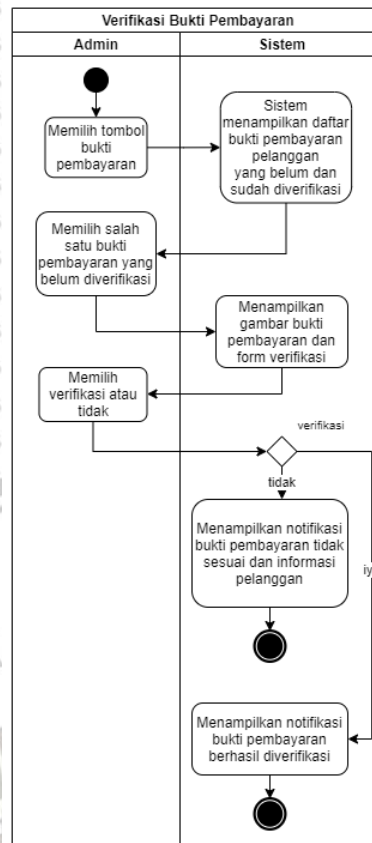
Gambar 5. 10 menjelaskan alur aktivitas user admin dalam melakukan *logout* agar bisa menghentikan autentikasi dan keluar dari sistem.



Gambar 5. 10 Activity Diagram Logout untuk Admin

5.1.1.11 Verifikasi Bukti Pembayaran untuk Admin

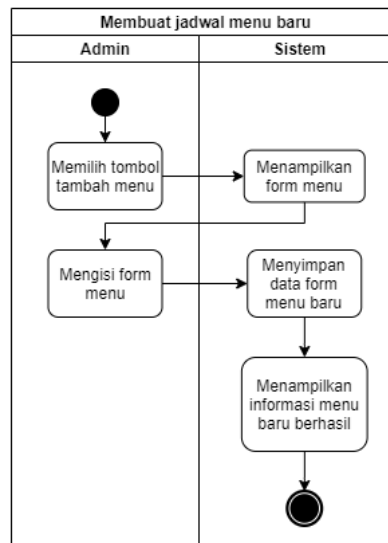
Gambar 5. 11 menjelaskan aktivitas verifikasi bukti pembayaran hanya bisa dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin bisa menganalisis keaslian dari foto bukti pembayaran yang diunggah oleh pelanggan. pengujian dilakukan saat admin telah selesai menganalisis foto bukti pembayaran. Alur pertama admin menentukan bahwa foto bukti pembayaran tersebut asli, sedangkan alur kedua admin menentukan bahwa foto bukti pembayaran tersebut tidak asli atau tidak cocok.



Gambar 5. 11 Activity Diagram Verifikasi Bukti Pembayaran

5.1.1.12 Membuat Jadwal Menu Baru untuk Admin

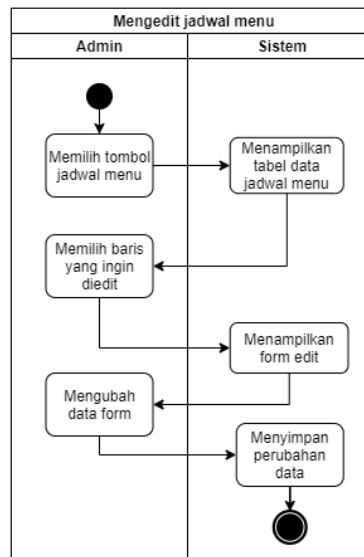
Gambar 5. 12 menjelaskan aktivitas membuat jadwal menu baru hanya dapat dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin dapat menambahkan menu baru beserta spesifikasinya yaitu tanggal & waktu menu dikirimkan, nama menu, keterangan, dan foto menu.



Gambar 5. 12 Activity Diagram Membuat Jadwal Menu Baru

5.1.1.13 Mengedit Jadwal Menu untuk Admin

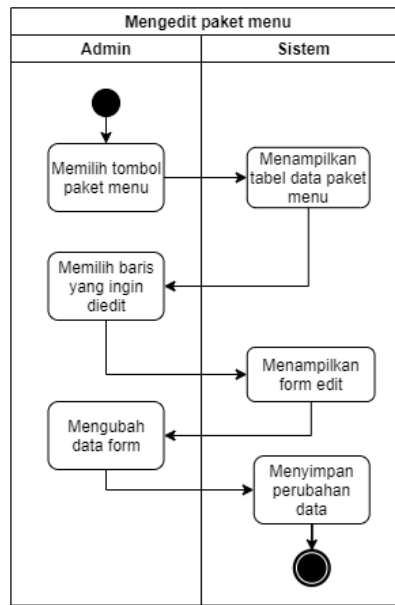
Gambar 5. 13 menjelaskan aktivitas mengedit jadwal menu hanya bisa dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin dapat melakukan perubahan data pada jadwal menu yang telah tersimpan.



Gambar 5. 13 Activity Diagram Mengedit Jadwal Menu

5.1.1.14 Mengedit Paket kupon untuk Admin

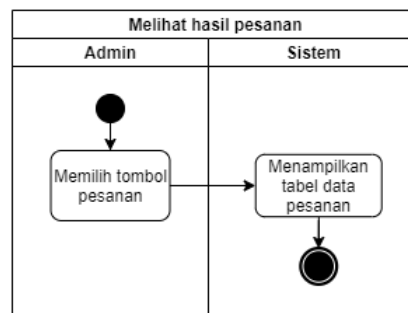
Gambar 5. 14 menjelaskan aktivitas mengedit paket kupon hanya bisa dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin dapat melakukan perubahan data pada tabel paket kupon.



Gambar 5. 14 Activity Diagram Mengedit Paket Kupon

5.1.1.15 Melihat Hasil Pesanan untuk Admin

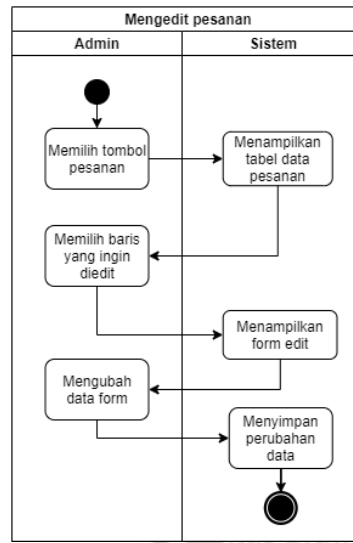
Gambar 5. 15 menjelaskan aktivitas melihat hasil pesanan hanya dapat dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin mendapatkan informasi penukaran kupon dengan menu yang dilakukan oleh pelanggan. Admin mendapatkan informasi kupon, yaitu nama pelanggan, status kupon, jenis paket kupon, tanggal & waktu pengiriman, biodata penerima, catatan makanan, catatan pengiriman, dan status pesanan.



Gambar 5. 15 Activity Diagram Melihat Hasil Pesanan

5.1.1.16 Mengedit Pesanan untuk Admin

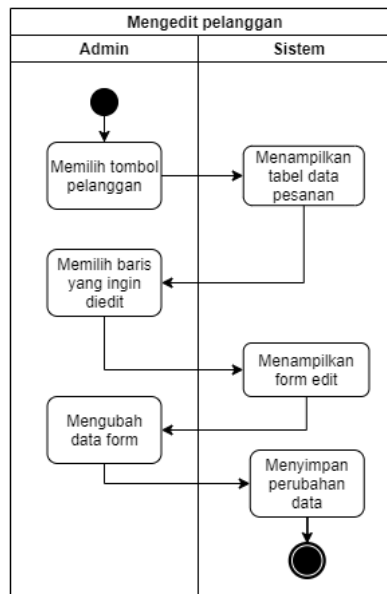
Gambar 5. 16 menjelaskan aktivitas mengedit pesanan hanya bisa dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin dapat melakukan perubahan data pada tabel pesanan yang telah diinput oleh pelanggan dengan cara penukaran kupon dengan menu.



Gambar 5. 16 Activity Diagram Mengedit Pesanan

5.1.1.17 Mengedit Pelanggan untuk Admin

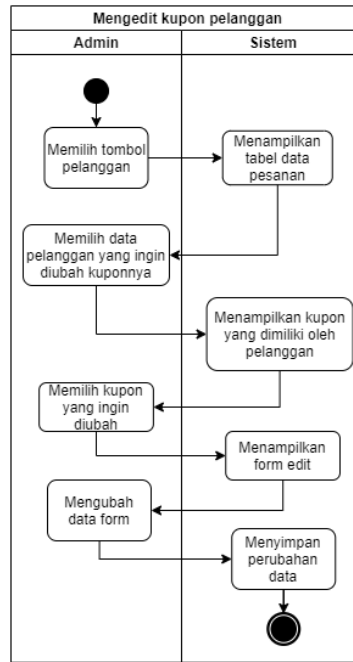
Gambar 5. 17 alur aktivitas mengedit pelanggan hanya bisa dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin dapat melakukan perubahan data pada tabel pelanggan yang diinput oleh pelanggan saat pendaftaran.



Gambar 5. 17 Activity Diagram Mengedit Pelanggan

5.1.1.18 Mengedit Kupon Pelanggan untuk Admin

Gambar 5. 18 alur aktivitas mengedit kupon pelanggan hanya bisa dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin dapat melakukan perubahan data pada tabel kupon pelanggan yang berasal dari pembelian kupon oleh pelanggan.



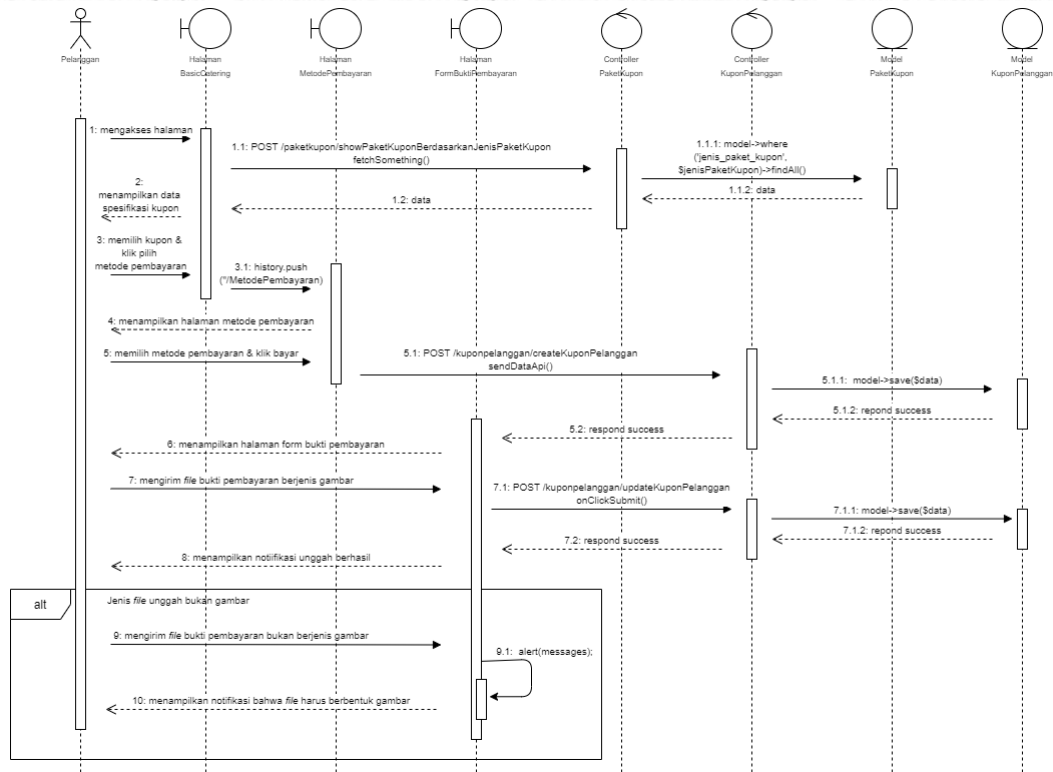
Gambar 5. 18 Activity Diagram Mengedit Kupon Pelanggan

5.1.2 Sequence Diagram

Setelah membuat *activity diagram* selanjutnya membuat *sequence diagram*. *Sequence diagram* adalah diagram yang menggambarkan komunikasi dinamis antar objek Ketika eksekusi sebuah tugas yang sedang berjalan (Pressman, 2010). Diagram ini berfungsi untuk memodelkan logika dari fungsi, operasi, atau prosedur dan memperlihatkan bagaimana sebuah objek dan komponen saling berinteraksi satu sama lain untuk menyelesaikan suatu proses. Terdapat 2 proses bisnis utama dalam Daily Catering yaitu pembelian kupon dan pemesanan menu. Proses pembelian kupon masuk dalam *use case* membeli kupon untuk pelanggan dan verifikasi bukti pembayaran untuk admin. Sedangkan, proses pemesanan menu masuk dalam *use case* menggunakan kupon untuk pelanggan dan melihat hasil pesanan untuk admin. Berikut penjelasan pemodelan *sequence diagram* untuk *use case* membeli kupon & menggunakan kupon untuk pelanggan dan *use case* verifikasi bukti pembayaran & melihat hasil pesanan untuk admin.

5.1.2.1 Sequence Diagram Membeli Kupon untuk Pelanggan

Aktivitas membeli kupon hanya bisa dilakukan oleh pelanggan saja. Membeli kupon berfungsi agar pelanggan dapat membeli kupon yang nantinya akan ditukarkan dengan menu sesuai dengan jadwalnya. Ada 2 alur, perbedaan alur terjadi saat pelanggan telah membayar kupon via transfer dan sedang ingin mengunggah bukti pembayaran. Alur utama pelanggan mengunggah file gambar sedangkan alur alternatif pelanggan mengunggah file bukan berbentuk gambar.



Gambar 5. 19 Sequence Diagram Membeli Kupon untuk Pelanggan

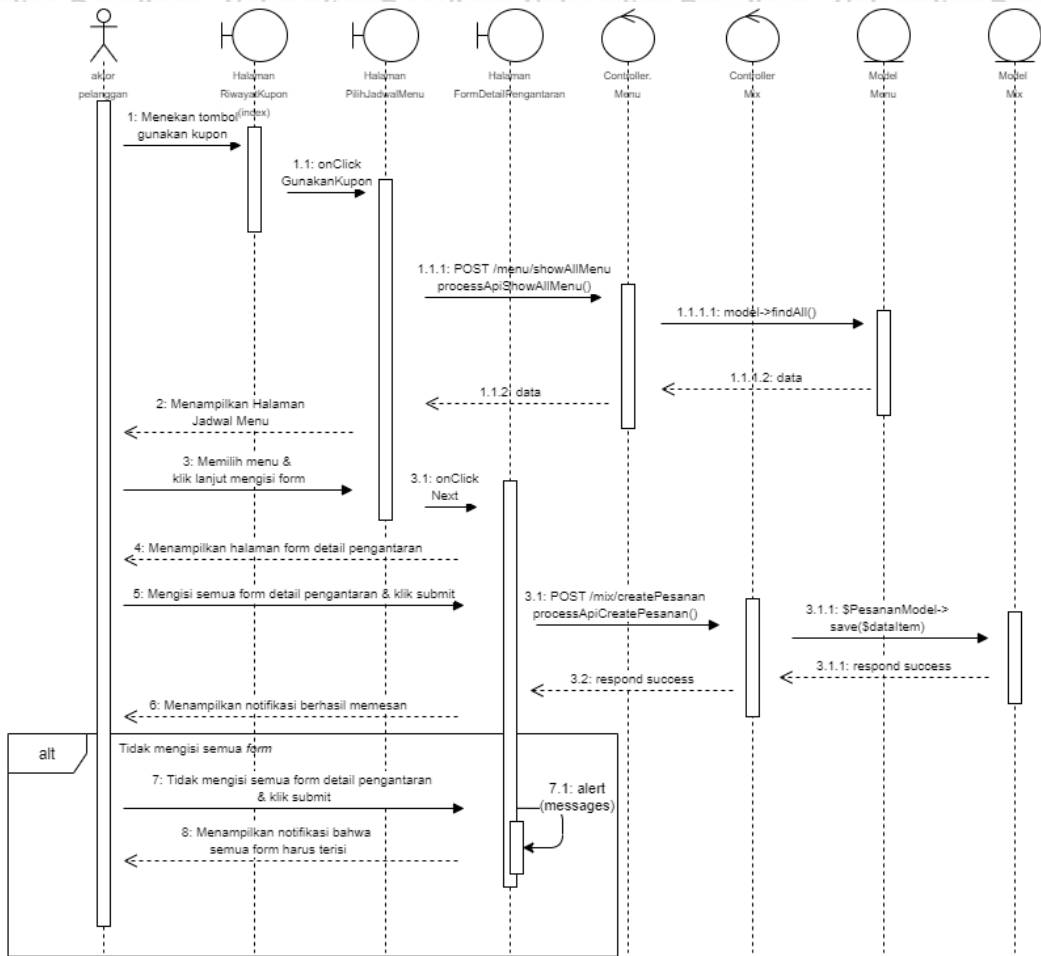
Dalam Gambar 5. 19 menjelaskan proses membeli kupon yang melibatkan aktor pelanggan, *boundary* halaman BasicCatering, *boundary* halaman MetodePembayaran, *boundary* halaman FormBuktiPembayaran, *control* PaketKupon, *control* KuponPelanggan, model PaketKupon, dan model KuponPelanggan. Awalnya pelanggan telah masuk ke dalam sistem dan berada di halaman BasicCatering, lalu memilih salah satu paket kupon yang tersedia. Kemudian, halaman BasicCatering menjalankan fungsi `fetchSomething()` yang berisi perintah *request* untuk mendapatkan data paket kupon dengan *method* `post` dan *url* `/paketkupon/showPaketKuponBerdasarkanJenisPaketKupon`. Lalu, diterima oleh *controller* PaketKupon yang menjalankan fungsi `model->where('jenis_paket_kupon', $jenisPaketKupon)->findAll()` ke model PaketKupon sehingga mendapatkan *return* sebuah data. Data tersebut diterima oleh *controller* PaketKupon lalu diteruskan ke halaman BasicCatering sebagai *respond*, sehingga aktor pelanggan mendapatkan informasi data tentang spesifikasi kupon (harga, jumlah, & jenis nasi) yang tersedia.

Setelah aktor pelanggan mendapatkan data kupon, pelanggan memilih salah satu kupon yang tersedia dan menekan tombol pilih metode pembayaran. Setelah itu, halaman BasicCatering akan mengarahkan ke halaman MetodePembayaran dengan menjalankan `history.push("/MetodePembayaran)`. Setelah aktor pelanggan masuk ke dalam halaman MetodePembayaran, pelanggan memilih salah satu metode pembayaran yang tersedia. Lalu, halaman MetodePembayaran menjalankan fungsi `sendDataApi()` melakukan *request method post* dengan *url* `/kuponpelanggan/createKuponPelanggan` bertujuan untuk membuat data di tabel kupon pelanggan. Setelah *request* diterima oleh *controller* KuponPelanggan, *controller* ini menjalankan `model->save($data)` ke model KuponPelanggan agar data baru dapat disimpan di tabel kupon pelanggan. Kemudian mengembalikan *respond* berhasil yang diterima oleh *controller* lalu membuka halaman FormBuktiPembayaran.

Setelah aktor pelanggan telah masuk ke halaman FormBuktiPembayaran, pelanggan mengunggah gambar bukti pembayaran dan menekan tombol *submit*. Lalu, halaman FormBuktiPembayaran akan menjalankan fungsi `onClickSubmit()` yang berisi *request post* dengan *link* `/kuponpelanggan/updateKuponPelanggan` bertujuan untuk melakukan menyimpan gambar bukti pembayaran. Setelah *request* diterima oleh *controller* KuponPelanggan, *controller* ini akan menjalankan `model->save($data)` ke model KuponPelanggan untuk disimpan ke *database*. Setelah berhasil menyimpan data, mengembalikan *respond* berhasil ke *controller*, lalu dilanjutkan ke FormBuktiPembayaran untuk menampilkan notifikasi berhasil mengunggah. Selain alur utama terdapat alur alternatif bila aktor pelanggan mengunggah *file* berjenis selain gambar, pengecekan dilakukan dalam halaman FormBuktiPembayaran lalu menampilkan notifikasi bahwa *file* yang diunggah harus berjenis gambar.

5.1.2.2 Sequence Diagram Menggunakan Kupon untuk Pelanggan

Aktivitas menggunakan kupon hanya bisa dilakukan oleh pelanggan saja. Menggunakan kupon berfungsi agar pelanggan dapat menukarkan kupon yang telah dibeli sebelumnya dengan menu makanan. Ada 2 alur, perbedaan alur saat pelanggan memasukkan *form* biodata diri dan catatan pengantaran untuk mempermudah pengiriman. Alur utama terjadi saat pelanggan mengisi semua *form* yang ada, sedangkan alur alternatif pelanggan tidak mengisi semua *form*.



Gambar 5. 20 Sequence Diagram Menggunakan Kupon untuk Pelanggan

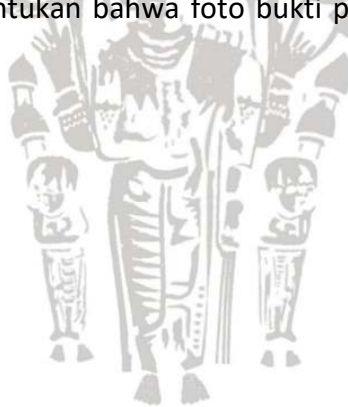
Dalam Gambar 5. 20 menjelaskan proses menggunakan kupon yang dilakukan oleh aktor pelanggan. Aktivitas ini melibatkan *boundary* halaman RiwayatKupon, *boundary* halaman PilihJadwalMenu, *boundary* halaman FormDetailPengantaran, *controller* menu, *controller* mix, *model* menu, dan *model* mix. Awalnya aktor pelanggan telah masuk ke dalam sistem dan berada di halaman RiwayatKupon, lalu menekan tombol gunakan kupon. Selanjutnya, halaman RiwayatKupon menjalankan fungsi `onClickGunakanKupon()` untuk membuka halaman PilihJadwalMenu. Halaman PilihJadwalMenu menjalankan fungsi `processApiShowAllMenu()` berisi *method* `post` dengan *link* `/menu/showAllMenu` bertujuan untuk mendapatkan semua data menu. Setelah *request* diterima oleh *controller* menu, *controller* ini menjalankan `model->findAll()` ke *model* menu. Selanjutnya, *model* menu akan mengembalikan data ke *controller* dan dilanjutkan ke halaman PilihJadwalMenu untuk ditampilkan.

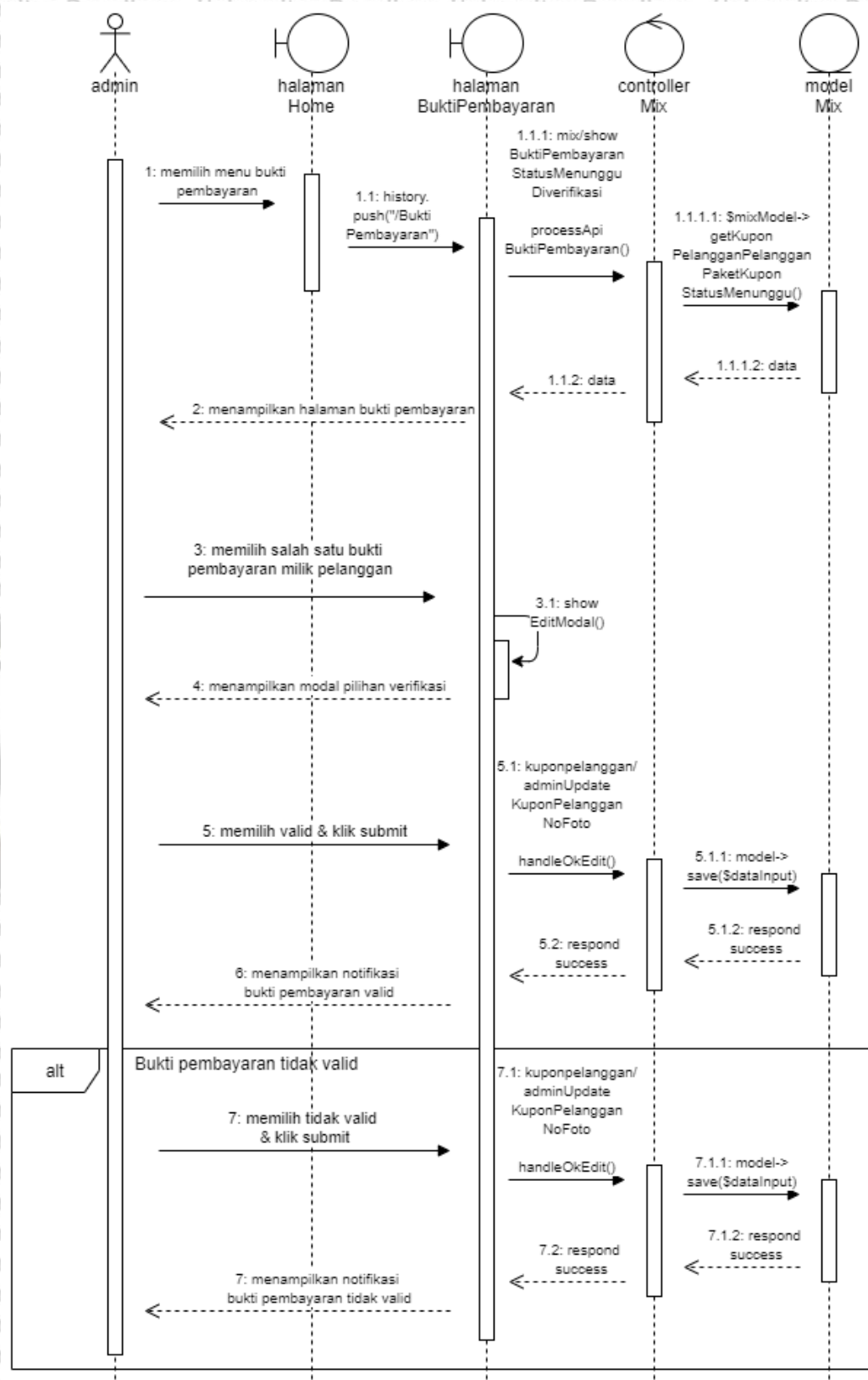
Setelah pelanggan mendapatkan informasi jadwal menu, pelanggan memilih menu beserta spesifikasinya (tanggal, waktu, dan nama makanan) dan menekan tombol lanjut. Setelah itu, halaman PilihJadwalMenu menjalankan fungsi `onClickNext` untuk membuka halaman FormDetailPengantaran. Setelah pelanggan masuk ke halaman FormDetailPengantaran, pelanggan mengisi semua

form seperti nama, alamat penerima, nomor *handphone*, catatan makanan, dan catatan pengantaran dan menekan tombol *submit*. Selanjutnya, halaman `FormDetailPengantaran` menjalankan fungsi `processApiCreatePesanan()` yang berisi *request method post* dengan *link /mix/createPesanan* bertujuan untuk membuat data pesanan baru. Setelah *controller* menerima *request* tersebut, *controller* ini menjalankan `$PesananModel->save($dataItem)` untuk menyimpan data, lalu mengembalikan *respond* berhasil ke *controller* dan dilanjutkan ke halaman `FormDetailPengantaran` untuk menampilkan notifikasi pemesanan berhasil. Selain alur utama terdapat alur alternatif dimana pelanggan tidak mengisi semua *form* yang ada dalam halaman `FormDetailPengantaran`. Oleh karena itu, halaman `FormDetailPengantaran` menjalankan fungsi `alert()` untuk menampilkan notifikasi bahwa semua *form* harus terisi.

5.1.2.3 Sequence Diagram Verifikasi Bukti Pembayaran untuk Admin

Aktivitas verifikasi bukti pembayaran hanya bisa dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin bisa menganalisis keaslian dari foto bukti pembayaran yang diunggah oleh pelanggan. Ada 2 alur, perbedaan alur saat admin telah selesai menganalisis foto bukti pembayaran. Alur utama saat admin menentukan bahwa foto bukti pembayaran tersebut asli, sedangkan alur alternatif saat admin menentukan bahwa foto bukti pembayaran tersebut tidak asli atau tidak cocok.





Gambar 5. 21 Sequence Diagram Verifikasi Bukti Pembayaran untuk Admin

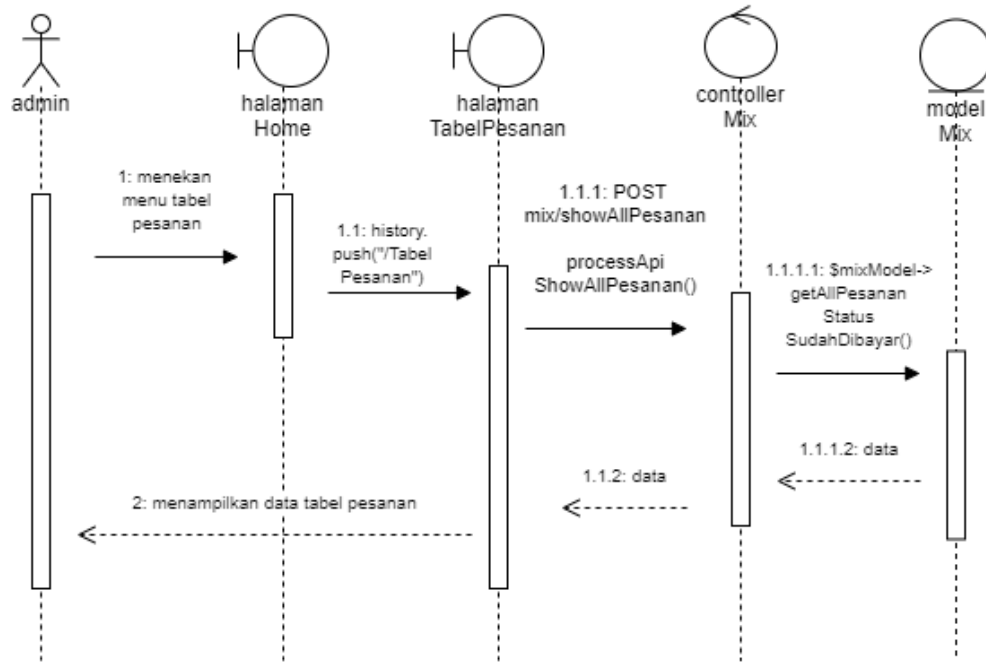
Dalam Gambar 5. 21 menjelaskan proses verifikasi bukti pembayaran yang dilakukan oleh aktor admin. Aktivitas ini melibatkan *boundary* halaman Home,

boundary halaman *BuktiPembayaran*, *controller* *mix*, dan *model* *mix*. Awalnya aktor pelanggan telah masuk ke dalam sistem dan berada di halaman *Home*, lalu memilih tombol menu verifikasi bukti pembayaran. Selanjutnya, halaman *Menu* akan menjalankan `history.push("/BuktiPembayaran")` agar masuk ke dalam halaman *BuktiPembayaran*. Kemudian, halaman *BuktiPembayaran* akan menjalankan fungsi `processApiBuktiPembayaran()` yang berisi *request method post* dengan *link* `mix/showBuktiPembayaranStatusMenungguDiverifikasi`. Setelah *controller* *mix* menerima *request* tersebut, ini menjalankan `$mixModel->getKuponPelangganPelangganPaketKuponStatusMenunggu()` untuk mendapatkan data tentang bukti pembayaran yang merupakan kumpulan dari tabel pelanggan dan kupon pelanggan. Kemudian *model* mengembalikan data yang diminta ke *controller*, lalu dilanjutkan ke halaman *BuktiPembayaran* untuk ditampilkan ke aktor admin.

Setelah admin mendapatkan informasi data bukti pembayaran berupa tabel di halaman *BuktiPembayaran*, admin memilih salah satu data dan menekan tombol periksa. Selanjutnya, halaman *BuktiPembayaran* menampilkan sebuah modal berisi gambar bukti pembayaran, biaya yang harus dibayarkan, nama pelanggan, tanggal pembelian, dan *form* verifikasi. Selanjutnya, admin memilih *form verifikasi* bernilai valid dan menekan tombol *submit*. Setelah itu, halaman *BuktiPembayaran* akan menjalankan fungsi `handleOkEdit()` yang berisi *request method post* dengan *link* `kuponpelanggan/adminUpdateKuponPelangganNoFoto`. Setelah *controller* *mix* menerima *request* tersebut, *controller* ini menjalankan fungsi `model->save($dataInput)` ke *model* *mix* untuk mengubah nilai status kupon pelanggan dari menunggu dikonfirmasi menjadi sudah dibayar. Setelah proses penyimpanan selesai, *model* mengirim *respond* berhasil ke *controller* lalu dilanjutkan ke halaman *BuktiPembayaran* untuk ditampilkan ke admin bahwa proses verifikasi valid berhasil. Selain alur utama terdapat alur alternatif dimana admin memilih bahwa bukti pembayaran tidak valid. Proses yang terjadi sama dengan verifikasi valid hanya saja nilai status kupon yang dimasukkan ke dalam database menjadi gagal dibayar dan notifikasi ke admin menjadi proses verifikasi tidak valid berhasil.

5.1.2.4 Sequence Diagram Melihat Hasil Pesanan untuk Admin

Aktivitas melihat hasil pesanan hanya dapat dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin mendapatkan informasi penukaran kupon dengan menu yang dilakukan oleh pelanggan. Admin mendapatkan informasi kupon, yaitu nama pelanggan, status kupon, jenis paket kupon, tanggal & waktu pengiriman, biodata penerima, catatan makanan, catatan pengiriman, dan status pesanan.

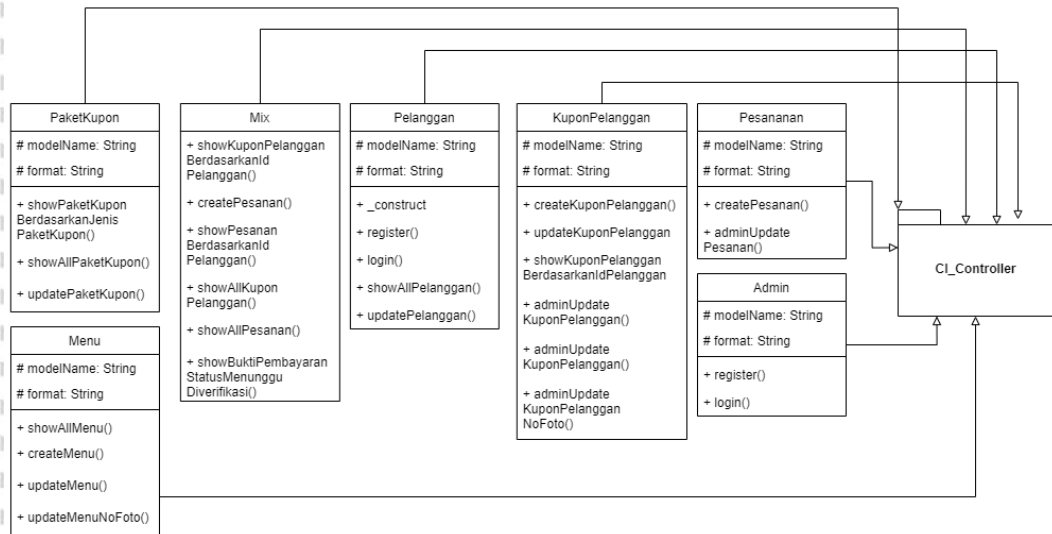


Gambar 5. 22 Sequence Diagram Melihat Hasil Pesanan untuk Admin

Dalam Gambar 5. 22 menjelaskan proses melihat hasil pesanan yang dilakukan oleh aktor admin. Aktivitas ini melibatkan *boundary* halaman Home, *boundary* halaman TabelPesanan, *controller* mix, dan *model* mix. Awalnya, admin yang telah masuk ke dalam sistem dan berada di halaman Home akan menekan tombol menu tabel pesanan. Kemudian, halaman Home akan menjalankan `history.push("/TabelPesanan")` untuk membuka halaman TabelPesanan. Selanjutnya, halaman TabelPesanan menjalankan fungsi `processApiShowAllPesanan()` yang berisi *request method post* dengan *link* `mix/showAllPesanan`. Setelah *controller* Mix mendapatkan *request* tersebut, *controller* ini menjalankan `$mixModel->getAllPesananStatusSudahDibayar()` ke *model* Mix untuk mendapatkan data pesanan yang berasal dari semua gabungan tabel dan status kupon pelanggan yang bernilai sudah dibayar saja. Setelah berhasil mendapatkan data pesanan, fungsi tersebut mengembalikan *respond* data ke *controller* dan dilanjutkan ke halaman TabelPesanan untuk ditampilkan ke aktor admin.

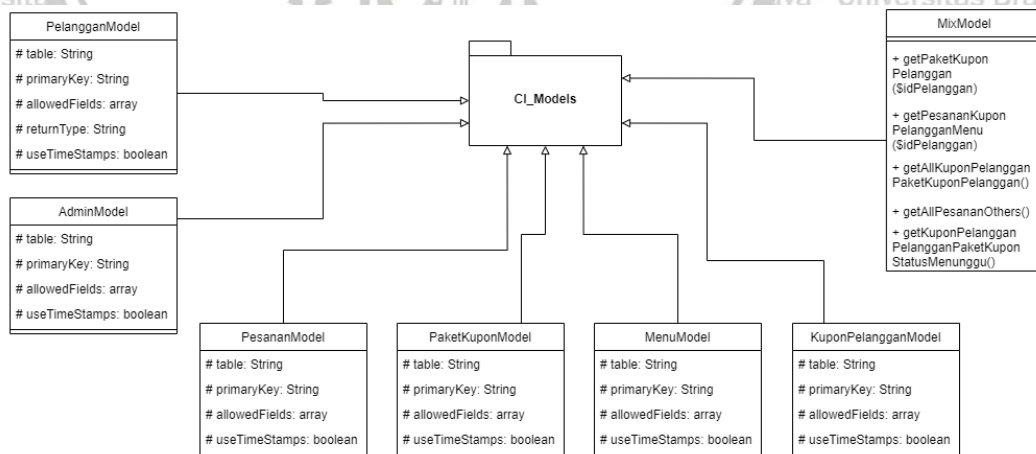
5.1.3 Class Diagram

Tahap selanjutnya adalah proses merancang *class diagram* yang akan digunakan dalam sistem. *Class diagram* adalah diagram yang menggambarkan aspek statis suatu sistem tentang kelas-kelas objek dari sistem, hubungan antar kelas, hingga batasan dari hubungan tersebut (Kim & David, 1999). Dalam penelitian ini terdiri dari *controller* dan *model* dikarenakan menggunakan *framework codeigniter*. Sedangkan tampilan *frontend* masuk dalam *Pages*.



Gambar 5. 23 Class Diagram Controller

Dalam Gambar 5. 23 berisi beberapa kelas *controller*, yaitu paket kupon, menu, mix, pelanggan, admin, kupon pelanggan, dan pesanan. Kelas pelanggan berisi logika yang memanajemen tabel admin dalam *database* seperti autentikasi saat admin melakukan *login*. Kelas pelanggan bertugas memanajemen tabel pelanggan dalam sistem admin. Kelas menu bertugas memanajemen tabel menu dalam *database* seperti menampilkan jadwal menu bagi pelanggan dan fitur edit menu bagi admin. Kelas paket kupon bertugas memanajemen tabel paket kupon dalam *database* seperti menampilkan paket kupon yang tersedia bagi pelanggan dan edit paket kupon bagi admin. Kelas kupon pelanggan bertugas memanajemen data kupon yang dimiliki oleh pelanggan, seperti menampilkan riwayat kupon milik pelanggan dan edit kupon pelanggan bagi admin. Kelas pesanan bertugas mencatat pesanan / penukaran kupon yang dilakukan oleh pelanggan. Kelas mix adalah kelas unik karena bertugas memanajemen berbagai tabel dalam *database*.



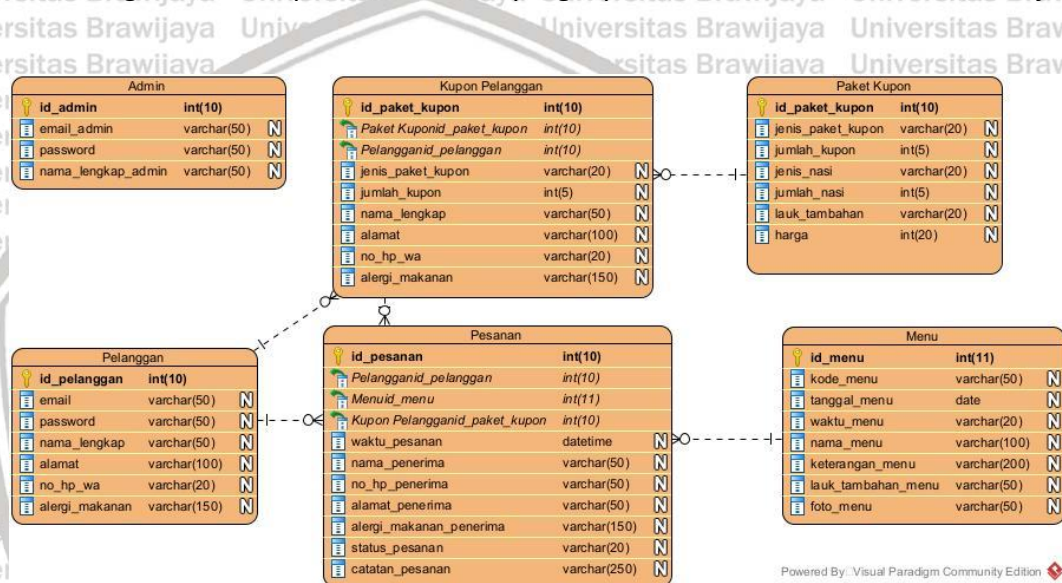
Gambar 5. 24 Class Diagram Models

Dalam Gambar 5. 24 berisi beberapa kelas model, yaitu admin model, pesanan model, menu model, kupon pelanggan model, pelanggan model, dan mix

model. Mix model merupakan kelas model khusus karena bertugas untuk melakukan *query* kepada lebih dari satu tabel di *database*. Sedangkan model lainnya yaitu PelangganModel, AdminModel, KuponPelangganModel, MenuModel, PaketKuponModel, dan PesananModel hanya bertugas menjadi penghubung antara *database* dan *controller* sendiri-sendiri.

5.1.4 Entity Relationship Diagram

Entity Relationship Diagram (ERD) adalah diagram struktural yang digunakan untuk memodelkan rancangan *database* (Visual Paradigm, 2021). Pada bagian ini merancang struktur data yang digunakan pada sistem. *Database* yang dirancang didasarkan pada kebutuhan yang diperlukan oleh sistem ketika bekerja.



Gambar 5. 25 Entity Relationship Diagram

Dalam Gambar 5. 25 *entity relationship diagram* terdiri dari 6 tabel, yaitu tabel admin, pelanggan, menu, paket kupon, kupon pelanggan, dan pesanan. Tabel admin berfungsi sebagai tempat penyimpanan data admin seperti *email* dan *password* yang telah di-*hash*, digunakan saat melakukan autentikasi *login* admin. Tabel pelanggan berfungsi sebagai tempat penyimpanan data pelanggan, data *email* dan *password* digunakan untuk autentikasi *login* pelanggan, sedangkan data nama lengkap, alamat, no hp wa, dan alergi makanan digunakan untuk nilai *default* saat pemesanan / penukaran kupon. Tabel menu berfungsi sebagai tempat penyimpanan data menu seperti nama menu, tanggal & waktu pengiriman, keterangan, lauk tambahan, dan foto, data dalam tabel ini diperlihatkan saat pelanggan melakukan *case* lihat jadwal menu dan memilih menu. Tabel paket kupon berisi data paket kupon (jenis, jumlah kupon, jenis nasi, dan harga) digunakan saat pelanggan dalam proses pembelian kupon. Setelah pelanggan memilih dan membeli kupon, data akan disimpan dalam tabel kupon pelanggan. Saat penukaran kupon dengan menu yang dilakukan oleh pelanggan, data akan tersimpan dalam tabel pesanan.

5.1.5 Desain Antarmuka

Tahap terakhir dalam perancangan sistem adalah desain antarmuka. Untuk meningkatkan peluang keberhasilan dan angka kepuasan pengguna, antarmuka pengguna harus memiliki prinsip seperti, perspektif dari user, inovasi, produk yang jelas, mengandung estetika desain, desain yang konsisten, familiar, dan elemen visual yang kuat (Oliver, 2020). Hasil desain antarmuka ini akan dijadikan patokan pembuatan sistem nantinya. Pada penelitian ini menggunakan teknik *wireframe* dalam pembuatan desain antarmuka. Melakukan tahap *wireframe* memiliki keuntungan, yaitu memastikan konsep fokus pada user, memperjelas fitur, dan cepat dan murah (Rahmalia, 2021)

5.1.5.1 Desain Antarmuka *Register* untuk Pelanggan

Gambar 5. 26 merupakan rancangan tampilan *register*, *register* hanya bisa dilakukan oleh pelanggan saja. *Register* berfungsi untuk menambahkan data akun pelanggan baru. Pelanggan harus mengisi *form* yaitu *email*, *password*, nama lengkap, alamat, nomor HP, dan catatan makanan. Pendaftaran akun pelanggan baru hanya bisa sukses bila *email* yang digunakan belum pernah terdaftar sebelumnya.

X Daftar

logo

Buat akun Senjanimu

Email
baariqazhar@gmail.com

Password

Ulangi password

Nama Lengkap
Baariq Fairuz Azhar

Alamat
Jl. Candi 2C No.557 (Kos Rahman 99),
Karangbesuki, Kec. Sukun, Kota Malang, Jawa
Timur 65149

Nomor HP (Whatsapp)
08123xxxxxx

Catatan makanan
mis. Alergi kacang tanah, tidak suka pedas

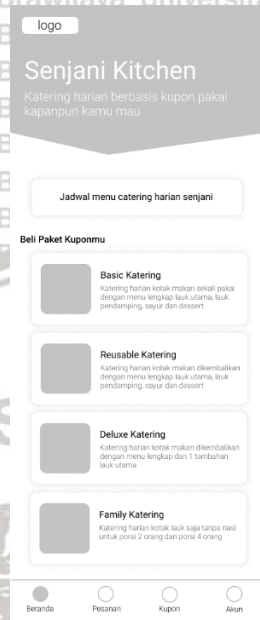
Daftar

Sudah punya akun Senjani? Masuk

Gambar 5. 26 Desain Antarmuka *Register* untuk Pelanggan

5.1.5.2 Desain Antarmuka Utama untuk Pelanggan

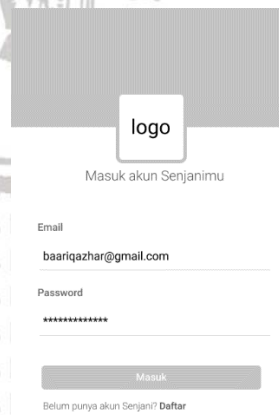
Gambar 5. 27 merupakan rancangan antarmuka halaman utama yang digunakan oleh pelanggan untuk masuk ke dalam fitur-fitur yang tersedia dan gambar sebelah kanan saat pelanggan belum mengunggah bukti pembayaran dalam pembelian kupon.



Gambar 5. 27 Desain Antarmuka Utama untuk Pelanggan

5.1.5.3 Desain Antarmuka Login untuk Pelanggan

Gambar 5. 28 merupakan rancangan antarmuka halaman *login*, *login* berfungsi agar pelanggan dapat masuk kedalam sistem menggunakan *email* dan *password* yang telah terdaftar sebelumnya.



Gambar 5. 28 Desain Antarmuka Login untuk Pelanggan

5.1.5.4 Desain Antarmuka Jadwal Menu untuk Pelanggan

Gambar 5. 29 merupakan rancangan antarmuka halaman jadwal menu, aktivitas melihat jadwal menu hanya dapat dilakukan oleh pelanggan saja. Melihat jadwal menu berfungsi agar pelanggan mendapatkan informasi spesifikasi menu, yaitu jadwal dan waktu menu dikirimkan, nama menu, dan keterangan menu.

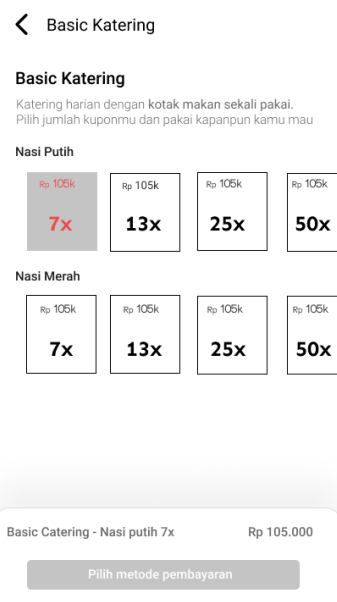


Gambar 5. 29 Desain Antarmuka Jadwal Menu untuk Pelanggan

5.1.5.5 Desain Antarmuka Pilih Paket Kupon untuk Pelanggan

Gambar 5. 30 merupakan rancangan antarmuka halaman pilih paket kupon yang digunakan oleh pelanggan untuk mendapatkan informasi spesifikasi kupon yang ingin dibeli.

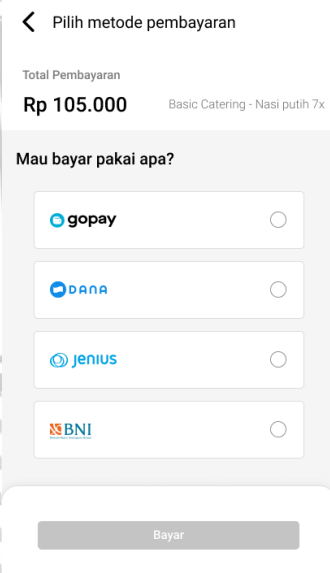




Gambar 5. 30 Desain Antarmuka Pilih Paket Kupon untuk Pelanggan.

5.1.5.6 Desain Antarmuka Metode Pembayaran untuk Pelanggan

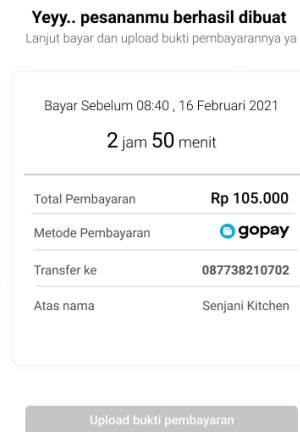
Gambar 5. 31 merupakan rancangan antarmuka halaman metode pembayaran yang digunakan oleh pelanggan untuk mendapatkan informasi metode pembayaran yang tersedia dalam proses pembelian kupon.



Gambar 5. 31 Desain Antarmuka Metode Pembayaran untuk Pelanggan

5.1.5.7 Desain Antarmuka Form Bukti Pembayaran bagi Pelanggan

Gambar 5. 32 merupakan rancangan antarmuka halaman metode pembayaran yang digunakan oleh pelanggan untuk mendapatkan informasi metode pembayaran yang tersedia dalam proses pembelian kupon.



Gambar 5. 32 Desain Antarmuka Form Bukti Pembayaran bagi Pelanggan

5.1.5.8 Desain Antarmuka Pilih Jadwal Menu bagi Pelanggan

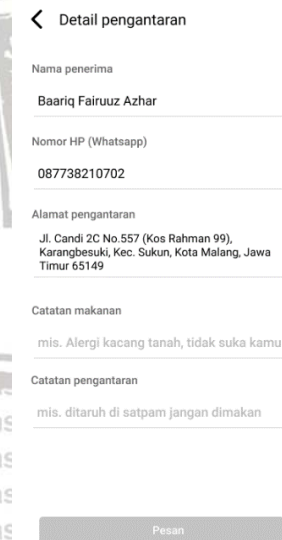
Gambar 5. 33 merupakan rancangan antarmuka halaman pilih jadwal menu yang digunakan oleh pelanggan untuk mendapatkan informasi jadwal menu beserta spesifikasi seperti tanggal, waktu, nama menu, pelengkap menu, menu penutup, dan lain-lain yang tersedia dalam proses penggunaan kupon.



Gambar 5. 33 Desain Antarmuka Pilih Jadwal Menu bagi Pelanggan

5.1.5.9 Desain Antarmuka Form Detail Pengantaran bagi Pelanggan

Gambar 5. 34 merupakan rancangan antarmuka halaman detail form detail pengantaran yang digunakan oleh pelanggan dalam proses penggunaan kupon. Pelanggan diharuskan mengisi form agar pengantaran catering dapat berjalan dengan baik.



Gambar 5. 34 Desain Antarmuka Form Detail Pengantaran bagi Pelanggan

5.1.5.10 Desain Antarmuka Riwayat Pesanan bagi Pelanggan

Gambar 5. 35 merupakan rancangan antarmuka halaman riwayat pesanan yang hanya dapat dilakukan oleh pelanggan saja. Melihat riwayat pesanan milik

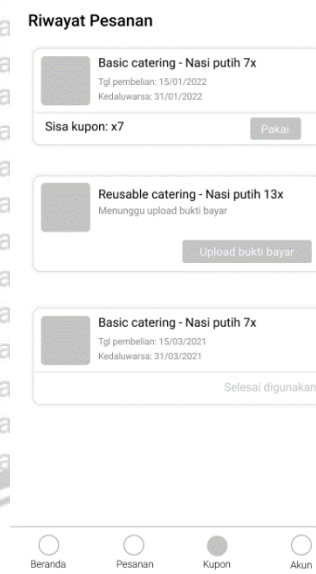
pelanggan itu sendiri berfungsi agar pelanggan mendapatkan informasi pesanan yang akan terjadi ataupun yang telah terjadi informasi yang didapatkan pelanggan, yaitu tanggal & waktu pengiriman, nama menu, keterangan menu, dan status pesanan.



Gambar 5. 35 Desain Antarmuka Riwayat Pesanan bagi Pelanggan

5.1.5.11 Desain Antarmuka Riwayat Kupon Pelanggan

Gambar 5. 36 merupakan rancangan antarmuka halaman riwayat kupon yang hanya dapat dilakukan oleh pelanggan saja. Melihat riwayat kupon milik pelanggan itu sendiri berfungsi agar pelanggan mendapatkan informasi kupon yang dimilikinya beserta spesifikasinya (jumlah kupon tersisa, jenis paket kupon, tanggal pembelian, tanggal kedaluwarsa, dan status kupon).



Gambar 5. 36 Desain Antarmuka Riwayat Kupon Pelanggan

5.1.5.12 Desain Antarmuka *Login* bagi Admin

Gambar 5. 37 merupakan rancangan antarmuka halaman *login* yang dilakukan oleh admin untuk bisa masuk ke dalam sistem. Admin harus memasukkan *email* dan *password* untuk masuk ke dalam system.

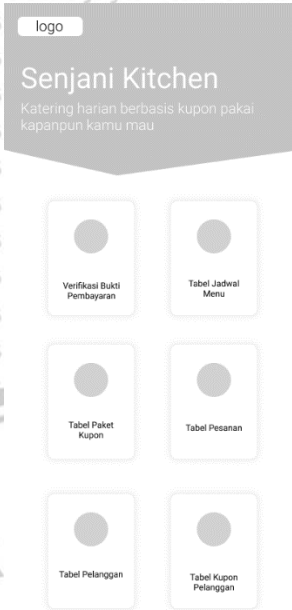


Gambar 5. 37 Desain Antarmuka *Login* bagi Admin

5.1.5.13 Desain Antarmuka Utama bagi Admin

Gambar 5. 38 merupakan rancangan antarmuka halaman utama yang dilakukan oleh admin. Halaman ini berguna agar admin dapat menggunakan fitur-fitur yang telah disediakan oleh sistem. Fitur-fitur yang dapat dipilih oleh admin

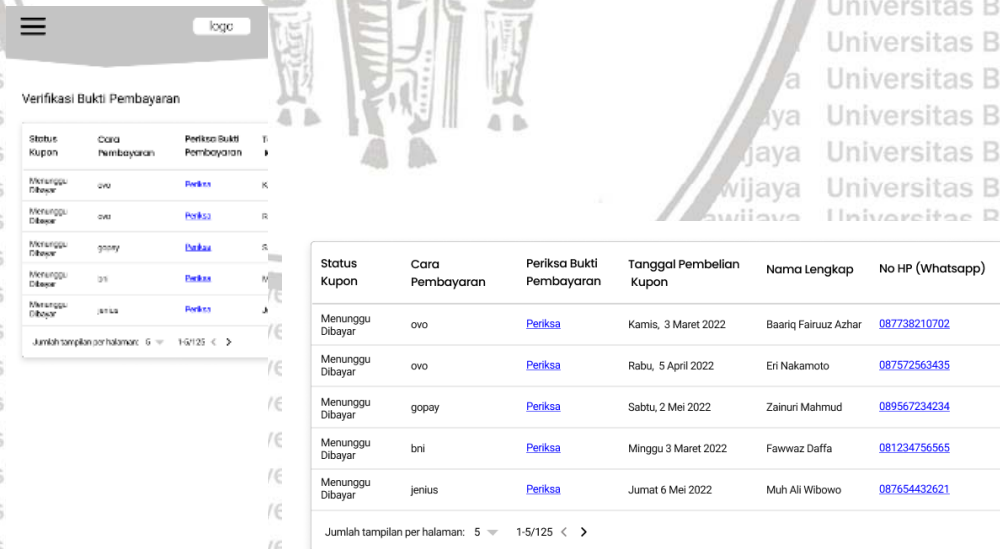
adalah verifikasi bukti pembayaran, tabel jadwal menu, tabel pesanan, tabel paket kupon, tabel pelanggan, dan tabel kupon pelanggan.



Gambar 5. 38 Desain Antarmuka Utama bagi Admin

5.1.5.14 Desain Antarmuka Verifikasi Bukti Pembayaran bagi Admin

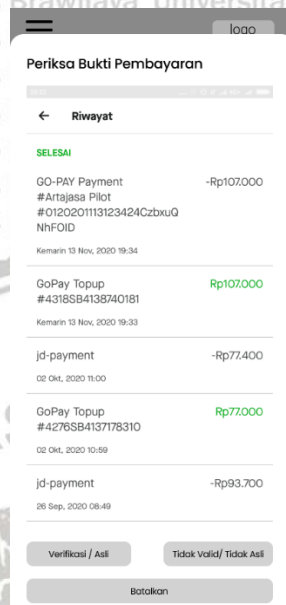
Kedua Gambar 5. 39 merupakan rancangan antarmuka halaman verifikasi bukti pembayaran yang dilakukan oleh admin. Halaman ini berguna agar admin mendapatkan informasi bukti pembayaran yang telah diunggah oleh pelanggan.



Gambar 5. 39 Desain Antarmuka Verifikasi Bukti Pembayaran bagi Admin

5.1.5.15 Desain Antarmuka Form Verifikasi Bukti Pembayaran bagi Admin

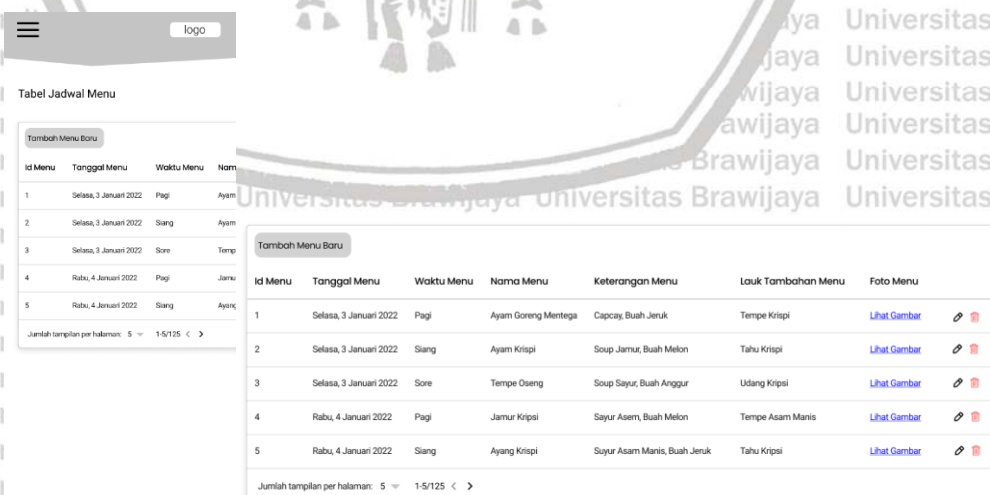
Gambar 5. 40 merupakan rancangan antarmuka halaman verifikasi bukti pembayaran yang dilakukan oleh admin. Halaman ini berguna agar admin mendapatkan informasi bukti pembayaran yang telah diunggah oleh pelanggan dan melakukan verifikasi apakah bukti pembayaran valid atau tidak.



Gambar 5. 40 Desain Antarmuka Form Verifikasi Bukti Pembayaran bagi Admin

5.1.5.16 Desain Antarmuka Tabel Jadwal Menu bagi Admin

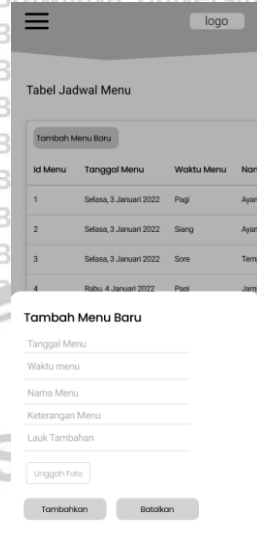
Gambar Gambar 5. 41 merupakan rancangan antarmuka halaman tabel jadwal menu yang dilakukan oleh admin. Halaman ini berguna agar admin mendapatkan informasi tentang tabel jadwal menu.



Gambar 5. 41 Desain Antarmuka Tabel Jadwal Menu bagi Admin

5.1.5.17 Desain Antarmuka Form Tambah Tabel Jadwal Menu bagi Admin

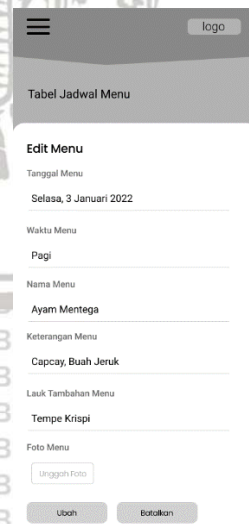
Gambar 5. 42 merupakan rancangan antarmuka halaman form tambah menu pada tabel jadwal menu yang dilakukan oleh admin. Halaman ini berguna agar admin bisa menambahkan menu baru pada tabel.



Gambar 5. 42 Desain Antarmuka Form Tambah Tabel Jadwal Menu bagi Admin

5.1.5.18 Desain Antarmuka Form Edit Tabel Jadwal Menu bagi Admin

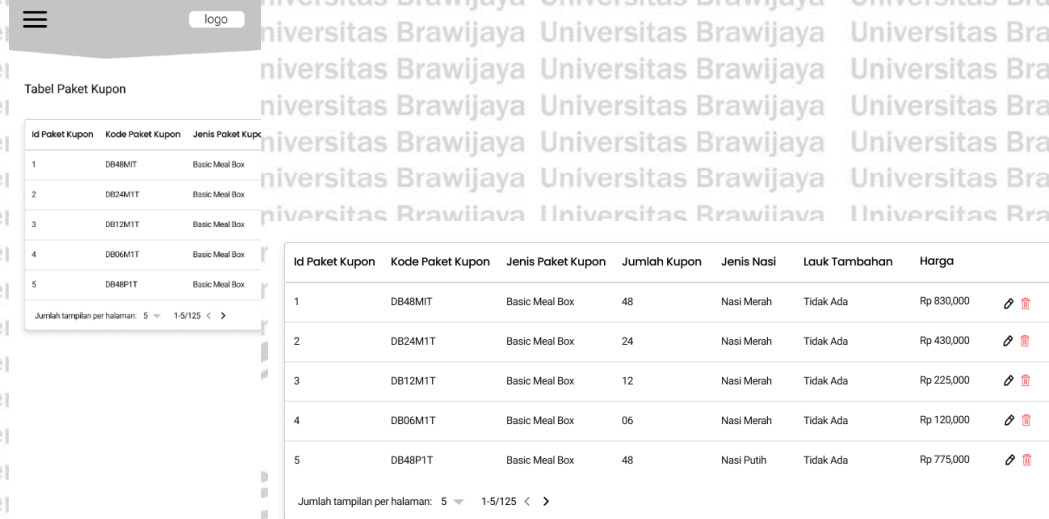
Gambar 5. 43 merupakan rancangan antarmuka halaman form edit menu pada tabel jadwal menu yang dilakukan oleh admin. Halaman ini berguna agar admin dapat mengubah menu pada tabel.



Gambar 5. 43 Desain Antarmuka Form Edit Tabel Jadwal Menu bagi Admin

5.1.5.19 Desain Antarmuka Tabel Paket Kupon bagi Admin

Gambar 5. 44 merupakan rancangan antarmuka halaman tabel paket kupon yang dilakukan oleh admin. Halaman ini berguna agar admin mendapatkan informasi tentang tabel paket kupon.



Gambar 5. 44 Desain Antarmuka Tabel Paket Kupon bagi Admin

5.1.5.20 Desain Antarmuka Form Edit Tabel Paket Kupon bagi Admin

Gambar 5. 45 merupakan rancangan antarmuka halaman form edit paket kupon pada tabel paket kupon yang dilakukan oleh admin. Halaman ini berguna agar admin dapat mengubah paket kupon pada tabel.

Gambar 5. 45 Desain Antarmuka Form Edit Tabel Paket Kupon bagi Admin

5.1.5.21 Desain Antarmuka Tabel Pesanan

Gambar 5. 46 merupakan rancangan antarmuka halaman tabel pesanan yang dilakukan oleh admin. Halaman ini berguna agar admin mendapatkan informasi tentang tabel pesanan.

ID Pesanan	Pelanggan (Nama)	Kupon Pelanggan (id,status)	Menu (Sat, tanggal/waktu)	Paket Kupon (Kode, jml)	Waktu Penawaran	Nama Pelanggan	No HP Pelanggan (WhatsApp)	Alamat Penitip	Lokasi Makanan Restoro	Status Pesanan	Opsi Pengiriman
1	1.Bearq Fauzan Akhir	1, sudah dibayar	71.Selam, 1. Januari 2022 7.Pagi	1.0418MIT Basic Meal Box	Senin, 3. Januari 2022 10.30	Beary Fauzan Akhir	08778212192	Jl. Cendek No.101 (Ruko Palmeras 02), Karangbala, Kec. Sukra, Kota Madiun	Suka Lado	Tidak Pkikan	Thp. delivery lebih
2	1.Bearq Fauzan Akhir	1, sudah dibayar	71.Selam, 1. Januari 2022 7.Sore	1.0418MIT Basic Meal Box	Rabu, 3. Januari 2022 17.30	Beary Fauzan Akhir	08702934835	Jl. Cendek No.101 (Ruko Palmeras 02), Karangbala, Kec. Sukra, Kota Madiun	Suka Lado	Tidak Pkikan	Insta sda
3	2.Zainu Mahmud	3, sudah dibayar	45.Selam, 1. Januari 2022 7.Pagi	1.0418MIT Basic Meal Box	Senyga, 3. Januari 2022 17.30	Zainu Mahmud	08952948234	Jl. Cendek No.101 (Ruko Palmeras 02), Karangbala, Kec. Sukra, Kota Madiun	Suka Lado	Sudah dikikan	Insta sda
4	2.Zainu Mahmud	3, sudah dibayar	45.Selam, 1. Januari 2022 7.Sore	1.0418MIT Basic Meal Box	Selam, 3. Januari 2022 17.30	Zainu Mahmud	08193879366	Jl. Cendek No.101 (Ruko Palmeras 02), Karangbala, Kec. Sukra, Kota Madiun	Suka Lado	Sudah dikikan	Tidak sda
5	3.Febrian Adji Febrian	6, sudah dibayar	71.Luarut, 8. Januari 2022 7.Pagi	1.0418MIT Basic Meal Box	Senin, 1. Februari 2022 10.30	Febrian Adji Febrian	081658457871	Jl. Cendek No.101 (Ruko Palmeras 02), Karangbala, Kec. Sukra, Kota Madiun	Suka Lado	Sudah dikikan	Tidak sda

Jumlah tampilan per halaman: 5 1-5/125 <

Gambar 5. 46 Desain Antarmuka Tabel Pesanan

5.1.5.22 Desain Antarmuka Form Edit Tabel Pesanan bagi Admin

Gambar 5. 47 merupakan rancangan antarmuka halaman form edit pesanan pada tabel pesanan yang dilakukan oleh admin. Halaman ini berguna agar admin dapat mengubah pesanan pada tabel.



Gambar 5. 47 Desain Antarmuka Form Edit Tabel Pesanan bagi Admin

5.1.5.23 Desain Antarmuka Tabel Pelanggan bagi Admin

Gambar 5. 48 merupakan rancangan antarmuka halaman tabel pelanggan yang dilakukan oleh admin. Halaman ini berguna agar admin mendapatkan informasi tentang tabel pelanggan.

Id Pelanggan	Email	Nama Lengkap	Alamat	No HP (Whatsapp)	Catatan Makanan
1	baariqazhar@gmail.com	Baariq Fairuz	Jl. Candi 2C No 557 (Kos Rahman 99), Karangbesuki, Kec. Sukun, Kota Malang	087738210702	tidak ada
2	erier@gmail.com	Eri Nakamoto	Jl. Candi 2C No 557 (Kos Rahman 99), Karangbesuki, Kec. Sukun, Kota Malang	087572563435	tidak suka pedas
3	zainurimahmud@gmail.com	Zainuri Mahmud	Jl. Candi 2C No 557 (Kos Rahman 99), Karangbesuki, Kec. Sukun, Kota Malang	089567234234	tidak mau sayur
4	fawwazdaffa@gmail.com	Fawwaz Daffa	Jl. Candi 2C No 557 (Kos Rahman 99), Karangbesuki, Kec. Sukun, Kota Malang	081234756565	tidak ada
5	alibowoo@gmail.com	Muh Ali Wibowo	Jl. Candi 2C No 557 (Kos Rahman 99), Karangbesuki, Kec. Sukun, Kota Malang	087654432621	tidak ada

Gambar 5. 48 Desain Antarmuka Tabel Pelanggan bagi Admin

5.1.5.24 Desain Antarmuka Form Edit Tabel Pelanggan bagi Admin

Gambar 5. 49 merupakan rancangan antarmuka halaman form edit isi data pelanggan pada tabel pelanggan yang dilakukan oleh admin. Halaman ini berguna agar admin dapat mengubah data pelanggan pada tabel.

Gambar 5. 49 Desain Antarmuka Form Edit Tabel Pelanggan bagi Admin

5.1.5.25 Desain Antarmuka Tabel Kupon Pelanggan

Gambar 5. 50 merupakan rancangan antarmuka halaman tabel kupon pelanggan yang dilakukan oleh admin. Halaman ini berguna agar admin mendapatkan informasi tentang tabel kupon yang dimiliki oleh pelanggan.

Id Kupon Pelanggan	Paket Kupon (cdkode, jenis)	Pelanggan (idnama)	Tanggal Pembelian	Tanggal Kedaluwarsa	Jumlah Kupon Tersisa	Status Kupon	Cara Pembayaran	Bukti Pembayaran	Waktu Batas Pembayaran
1	10848M1 Basic-Meal Box	1 Baariq Fairuz Azhar	Senin, 3 Januari 2022	Kamis, 3 Maret 2022	10	Sudah Dibayar	Duo	Lihat Gambar	Senin, 3 Januari 2022 (15:30)
2	20848M1 Basic-Meal Box	1 Baariq Fairuz Azhar	Rabu, 5 Februari 2022	Rabu, 5 April 2022	48	Sudah Dibayar	Gopay	Lihat Gambar	Rabu, 5 Februari 2022 (12:30)
3	30848M1 Basic-Meal Box	2 Zamru Mahmud	Minggu, 2 Maret 2022	Sabtu, 2 Mei 2022	34	Sudah Dibayar	Bni	Lihat Gambar	Minggu, 2 Maret 2022 (7:34)
4	50848M1 Basic-Meal Box	2 Zamru Mahmud	Selasa, 4 Januari 2022	Minggu, 3 Maret 2022	48	Sudah Dibayar	Gopay	Lihat Gambar	Selasa, 4 Januari 2022 (10:54)
5	70848M1 Basic-Meal Box	3 Febrian Aji Febian	Kamis, 4 Februari 2022	Jumat, 6 Mei 2022	48	Sudah Dibayar	Jenius	Lihat Gambar	Kamis, 4 Februari 2022 (1:24)

Gambar 5. 50 Desain Antarmuka Tabel Kupon Pelanggan

5.1.5.26 Desain Antarmuka Form Edit Tabel Kupon Pelanggan

Gambar 5. 51 merupakan rancangan antarmuka halaman form edit isi data kupon milik pelanggan pada tabel kupon pelanggan yang dilakukan oleh admin. Halaman ini berguna agar admin dapat mengubah data kupon pelanggan pada tabel.

Edit Kupon Pelanggan

Id Kupon Pelanggan
1

Id Paket Kupon
1

Id Pelanggan
1

Tanggal Pembelian
Senin, 3 Januari 2022

Tanggal Kedaluwarsa
Kamis, 3 Maret 2022

Jumlah Kupon tersedia
47

Status Kupon
Sudah dibayar

Cara Pembayaran
Ovo

Bukti Pembayaran
Unggah Foto

Waktu Batas Pembayaran
Kamis, 4 Maret 2022

Tambahkan Batalkan

Gambar 5. 51 Desain Antarmuka Form Edit Tabel Kupon Pelanggan

5.2 Implementasi Sistem

Pada tahap implementasi, rancangan sistem yang sudah dibuat diterjemahkan ke dalam bentuk kode menggunakan alat pengembangan aplikasi web yang telah ditentukan.

5.2.1 Spesifikasi Sistem

Spesifikasi sistem adalah penjelasan spesifikasi perangkat keras maupun perangkat lunak yang digunakan dalam merancang dan mengimplementasikan sistem. Dalam proses perancangan dan implementasi, penulis menggunakan komputer pribadi. Spesifikasi yang dijelaskan dalam Tabel 5. 1 terdiri dari CPU atau processor dan RAM atau *memory*. Tiap-tiap spesifikasi tersebut terdiri dari CPU atau *Processor*, *Memory* atau RAM.

Tabel 5. 1 Perangkat Keras

Perangkat	Deskripsi
CPU	Intel Core i7
RAM	8 GB DDR4

Selain perangkat keras juga dijelaskan perangkat lunak. Perangkat lunak yang digunakan dijelaskan pada Tabel 5. 2 ialah aplikasi dokumentasi, desain bagan, desain *wireframe*, *code editor*, bahasa pemrograman, *library & framework*, dan *browser*.

Tabel 5. 2 Perangkat Lunak

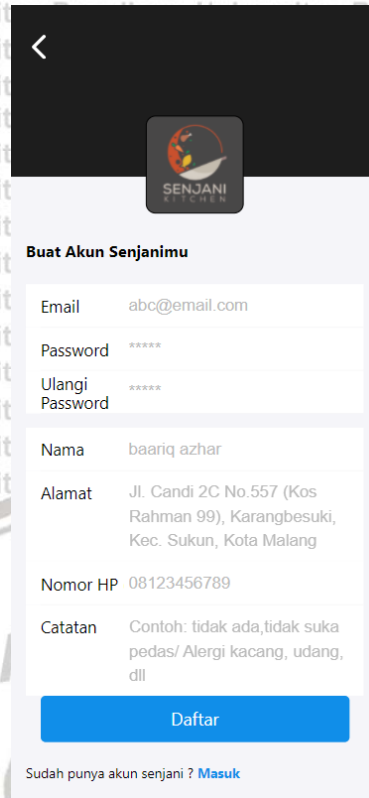
Perangkat	Deskripsi
Dokumentasi	Microsoft Office Word 2016
Desain Bagan	Draw Io dan Visual Paradigm
Desain <i>Wireframe</i>	Figma
<i>Code Editor</i>	Visual Studio Code
Bahasa Pemrograman	HTML, CSS, Javascript, dan PHP
<i>Library & Framework</i>	React Js versi 7 dan CodeIgniter versi 4
<i>Browser</i>	Chrome

5.2.2 Implementasi Kode

Sistem pemesanan Daily Catering merupakan sistem berbasis web. Untuk implementasi sistem berbasis web ini menggunakan konsep MVC (*Model View Controller*) dan optimasi web menggunakan *progressive web app*. Dalam MVC yang bertugas sebagai *view* adalah React Js Library berfungsi sebagai menampilkan dan berinteraksi dengan aktor. Sedangkan, *controller* dan *model* ditugaskan kepada *CodeIgniter Framework* berfungsi sebagai pengatur data sekaligus penghubung dengan *database*.

5.2.2.1 Implementasi *Register* untuk Pelanggan

Halaman Gambar 5. 52 *register* berfungsi sebagai proses pendaftaran pelanggan baru. Dalam proses pendaftaran akun pelanggan baru melibatkan halaman *register*, halaman utama, fungsi *register()* dalam *controller* pelanggan, dan model pelanggan. Halaman register untuk menampilkan dan memasukkan data bagi aktor yaitu pelanggan. Setelah pelanggan berhasil mendaftar, pelanggan akan masuk ke dalam halaman utama. Fungsi *register()* Tabel 5. 3 yang berada dalam *controller* pelanggan berfungsi memvalidasi dan memasukkan data baru atau akun pelanggan baru ke dalam tabel pelanggan. Model pelanggan berfungsi menjembatani *controller* pelanggan dengan *database*.



Gambar 5. 52 Hasil Implementasi Login untuk Pelanggan

Tabel 5. 3 Kode backend register() pada Controller Pelanggan

```

Pelanggan.php
1 public function register()
2     {
3         $data = $this->request->getPost();
4         if (!$this->model->where('email', $data['email'])) {
5             return $this->respond(['messages' => "email
telah terdaftar"]);
6         }
7         $validate = $this->validation->run($data,
'register');
8         $errors = $this->validation->getErrors();
9
10        if ($errors) {
11            // return $this->fail($errors);
12            return $this->respond($errors);
13        }
14
15        $pelanggan = new \App\Entities\Pelanggan();
16        $pelanggan->fill($data);
17        $pelanggan->created_by = 0;
18        $pelanggan->created_date = date("Y-m-d H:i:s");
19
20        if ($this->model->save($pelanggan)) {
21            $id_pelanggan = $this->model->getInsertId();
22            $dataPelangganFromDatabase = $this->model-
>find($id_pelanggan);
23            // return $this->respondCreated($pelanggan,
'pelanggan baru terbuat (created)');
    
```



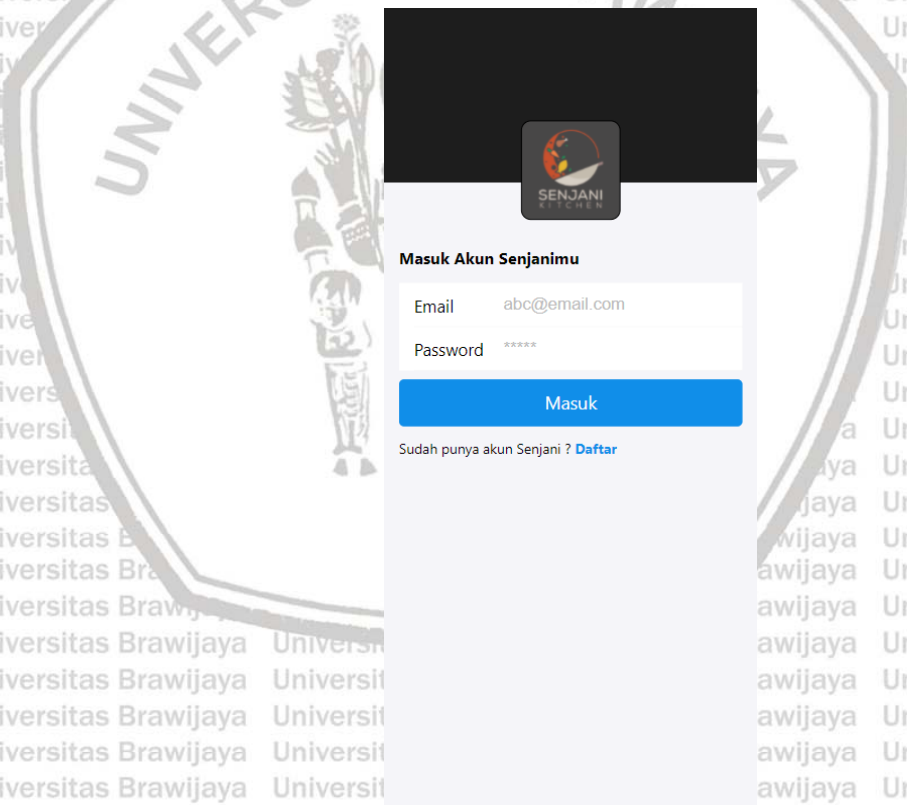
```

24
25     return $this->respondCreated(['id_pelanggan' =>
26     $id_pelanggan, 'status' => 'success', 'info' => 'register,
27     create', 'dataPelanggan' => $dataPelangganFromDatabase]);
    }
    }
    }

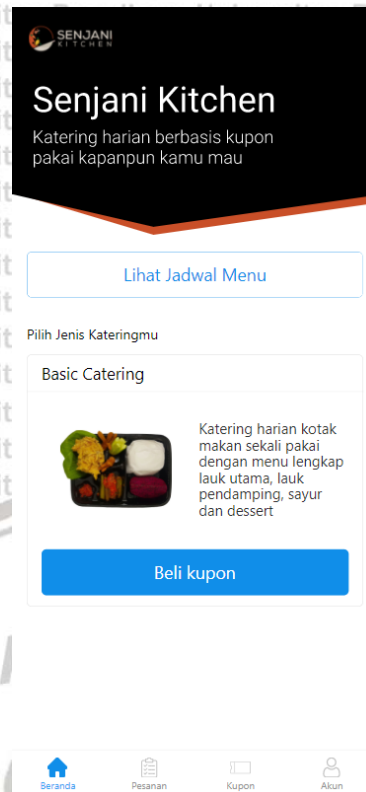
```

5.2.2.2 Implementasi *Login* untuk Pelanggan

Halaman *login* Gambar 5. 53 berfungsi sebagai proses autentikasi akun pelanggan yang sebelumnya telah terdaftar sehingga dapat masuk ke dalam halaman utama Gambar 5. 54. Dalam proses masuk pelanggan melibatkan halaman *login*, halaman utama, fungsi *login()* Tabel 5. 4 dalam *controller login*, dan model pelanggan. Halaman *login* berfungsi agar pelanggan dapat memasukkan *email* dan *password* yang telah terdaftar sebelumnya. Setelah pelanggan berhasil memasukkan *email* dan *password*, pelanggan akan masuk ke dalam halaman utama. Fungsi *login()* yang berada dalam *controller* pelanggan berfungsi memvalidasi data masuka ke dalam tabel pelanggan. Model pelanggan berfungsi menjembatani *controller* pelanggan dengan *database*.



Gambar 5. 53 Hasil Implementasi *Login* untuk Pelanggan



Gambar 5. 54 Hasil Implementasi Halaman Utama untuk Pelanggan

Tabel 5. 4 Kode Backend Login pada Controller Pelanggan

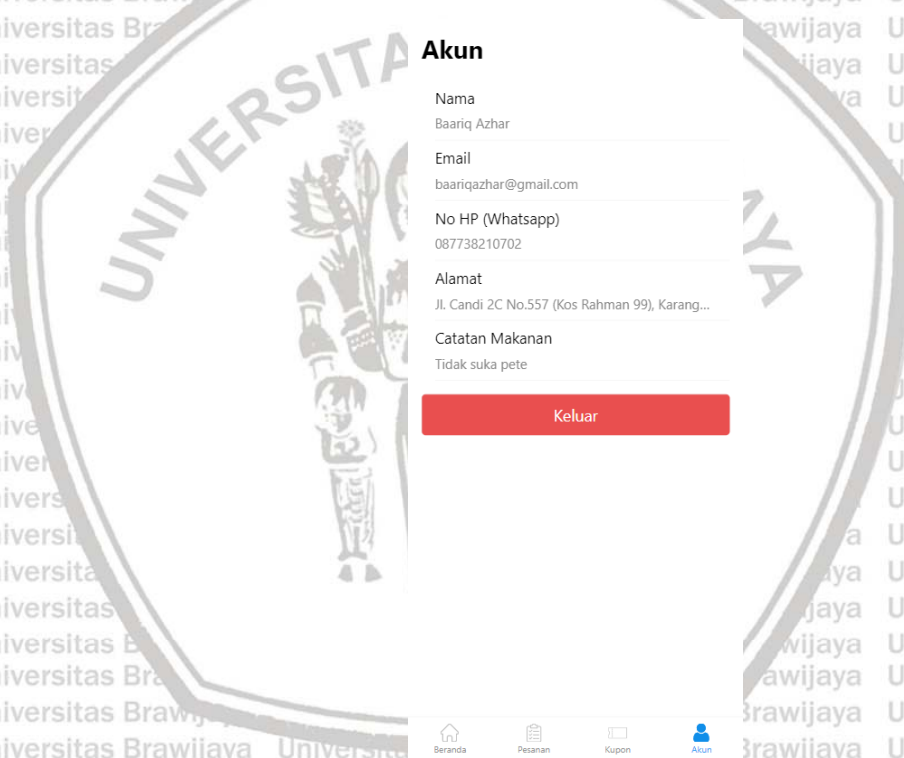
Pelanggan.php	
1	public function login()
2	{
3	\$data = \$this->request->getPost();
4	\$pelanggan = new \App\Entities\Pelanggan();
5	\$pelanggan->fill(\$data);
6	\$pelangganIdFromDatabase = \$this->model->
7	>where('email', \$pelanggan->email)->
8	>findColumn('id_pelanggan');
9	\$passwordFromDatabase = \$this->model->where('email',
10	\$pelanggan->email)->findColumn('password');
11	if (empty(\$passwordFromDatabase)) {
12	return \$this->respond(['id_pelanggan' =>
13	'status' => 'fail', 'info' => 'Login, login, email not
14	found']);
15	}
16	\$passwordInput = \$pelanggan->password;
17	if (implode(\$passwordFromDatabase) ===
18	\$passwordInput) {
19	\$dataPelangganFromDatabase = \$this->model->
20	>find(\$pelangganIdFromDatabase);
21	return \$this->respond(['id_pelanggan' =>
22	implode(\$pelangganIdFromDatabase), 'status' => 'success',

```

20 'info' => 'Login, login', "dataPelanggan" =>
21 $dataPelangganFromDatabase]);
18 } else {
19 return $this->respond(['id_pelanggan' =>
    implode($pelangganIdFromDatabase), 'status' => 'fail',
    'info' => 'Login, login, wrong password']);
20 }
21 }
    
```

5.2.2.3 Implementasi Logout untuk Pelanggan

Halaman *logout* Gambar 5. 55 berfungsi agar pelanggan dapat keluar dari sistem yang berjalan. Dalam proses *logout* ini melibatkan tab akun dalam halaman utama dan halaman *login*. Tab akun merupakan bagian dari halaman utama, berfungsi untuk menampilkan informasi tentang pelanggan dan tombol *logout* dengan kode pada tabel Tabel 5. 5. Setelah pelanggan menekan tombol *logout*, pelanggan akan masuk ke dalam halaman *login*.



Gambar 5. 55 Hasil Implementasi Logout untuk Pelanggan

Tabel 5. 5 Kode Frontend Akun

```

    Akun.jsx
1  const Akun = () => {
2  const onClickKeluar = () => {
3  localStorage.clear();
4  history.push("/Login");
    
```



```

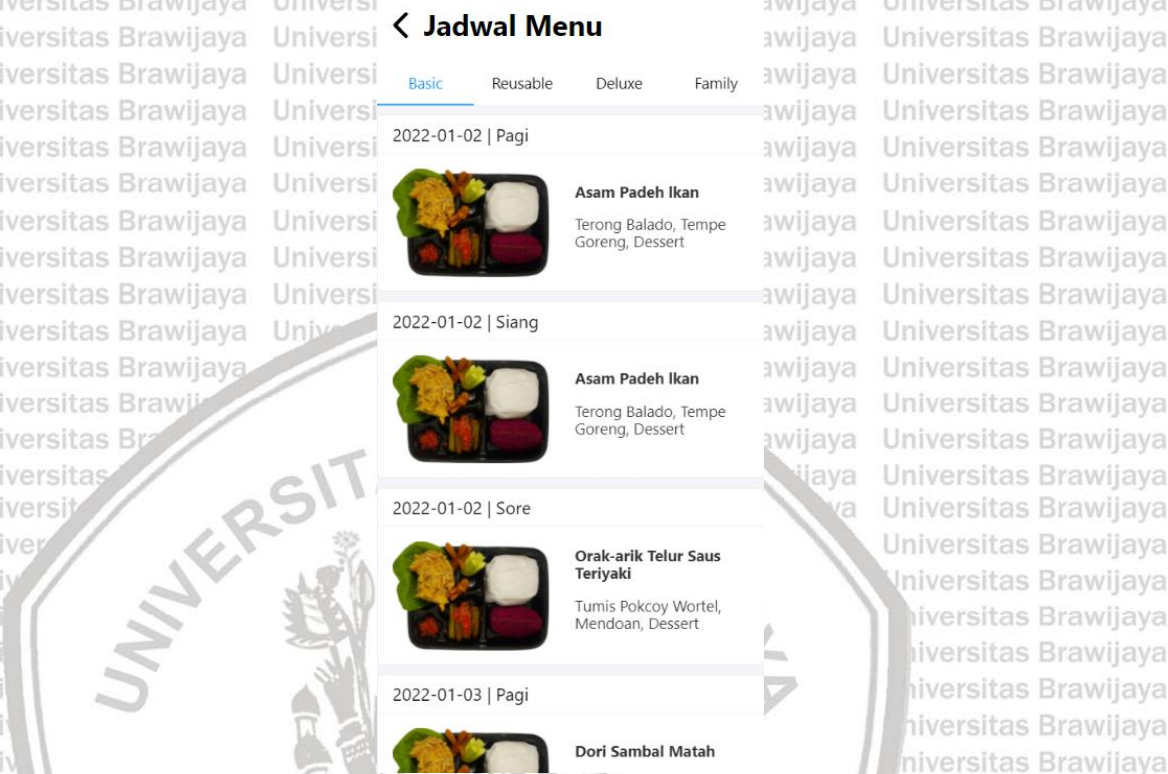
5     };
6     return (
7     <div>
8     <WingBlank>
9     <h1>Akun</h1>
10    <List.Item>
11      Nama <List.Item.Brief>
12      {pelanggan.nama_lengkap}</List.Item.Brief>
13    </List.Item>
14    <List.Item>
15      Email <List.Item.Brief>
16      {pelanggan.email}</List.Item.Brief>
17    </List.Item>
18      No HP (Whatsapp){ " " }
19    <List.Item.Brief>
20      {pelanggan.no_hp_wa}</List.Item.Brief>
21    </List.Item>
22      Alamat <List.Item.Brief>
23      {pelanggan.alamat}</List.Item.Brief>
24    </List.Item>
25      Catatan Makanan{" " }
26    <List.Item.Brief>
27      {pelanggan.alergi_makanan}</List.Item.Brief>
28    </List.Item>
29    <WhiteSpace size="lg" />
30    <Button type="warning" onClick={onClickKeluar}>
31      Keluar
32    </Button>
33  </WingBlank>
34 </div>

```

5.2.2.4 Implementasi Melihat Jadwal Menu untuk Pelanggan

Halaman melihat jadwal menu Gambar 5. 56 berfungsi agar pelanggan mendapatkan informasi tentang jadwal menu. Dalam proses melihat jadwal ini melibatkan tombol lihat jadwal menu dalam halaman utama, halaman jadwal menu, fungsi *index* dalam *controller menu*, dan model menu. Halaman utama berfungsi menampilkan tombol lihat menu yang dapat ditekan oleh pelanggan. Halaman jadwal menu menampilkan menu beserta spesifikasi yaitu tanggal,

waktu, lauk utama, lauk pelengkap, dan makanan penutup. Fungsi *indeks* Tabel 5.6 dalam *controller* menu berfungsi untuk mengambil semua data yang ada dalam tabel menu, lalu mengirimkan ke halaman jadwal menu. Model menu berfungsi menjembatani *controller* menu dengan *database*.



Gambar 5. 56 Hasil Implementasi Melihat Jadwal Menu untuk Pelanggan

Tabel 5. 6 Kode *Backend Index* pada *Controller Menu*

Menu.php	
1	public function index()
2	{
3	return \$this->respond(\$this->model->findAll());
4	}

5.2.2.5 Implementasi Membeli Kupon untuk Pelanggan

Aktivitas membeli kupon berfungsi agar pelanggan dapat membeli kupon yang tersedia sesuai dengan jenis paket kuponnya. Dalam proses membeli kupon melibatkan tombol beli kupon pada halaman utama, halaman pilih kupon, halaman metode pembayaran, halaman bukti pembayaran, fungsi *showPaketKuponBerdasarkanJenisPaketKupon* dalam *controller* paket kupon, fungsi *createKuponPelanggan* dalam *controller* kupon pelanggan, dan model paket kupon.

Basic Catering



Katering harian dengan kotak makan sekali pakai.
Pilih jumlah kuponmu dan pakai kapanpun kamu mau

Nasi Putih

- 48
Rp 775000
- 24
Rp 405000
- 12
Rp 210000
- 6
Rp 110000

Nasi Merah

40

Gambar 5. 57 Hasil Implementasi Pilih Paket Kupon untuk Pelanggan

Dalam halaman Gambar 5. 57 pilih paket kupon menampilkan informasi kupon beserta spesifikasi seperti harga dan jumlah kupon yang dapat ditukarkan.

Tabel 5. 7 Kode Backend ShowPaketKuponBerdasarkanJenisPaketKupon pada Controller PaketKupon

PaketKupon.php	
1	public function
	showPaketKuponBerdasarkanJenisPaketKupon ()
2	{
3	\$data = \$this->request->getPost ();
4	\$jenisPaketKupon = \$data['jenis_paket_kupon'];
5	\$dataFromDatabase =
	\$this->model->where('jenis_paket_kupon',
	\$jenisPaketKupon)->findAll ();
6	return \$this->respond (\$dataFromDatabase);
7	}

< Metode Pembayaran

Total Pembayaran

Rp 110000

Kupon Basic Meal Box

Nasi putih
6x

Mau bayar pakai apa?

GoPay

Dana

Jenius

BNI

Bayar

Gambar 5. 58 Hasil Implementasi Metode Pembayaran untuk Pelanggan

Dalam metode pembayaran Gambar 5. 58, sistem menampilkan total harga yang harus dibayarkan, spesifikasi paket kupon yang dibeli (jenis paket kupon, jenis nasi, jumlah kupon), dan jenis pembayaran. Pelanggan dapat memilih salah satu dari jenis pembayaran.

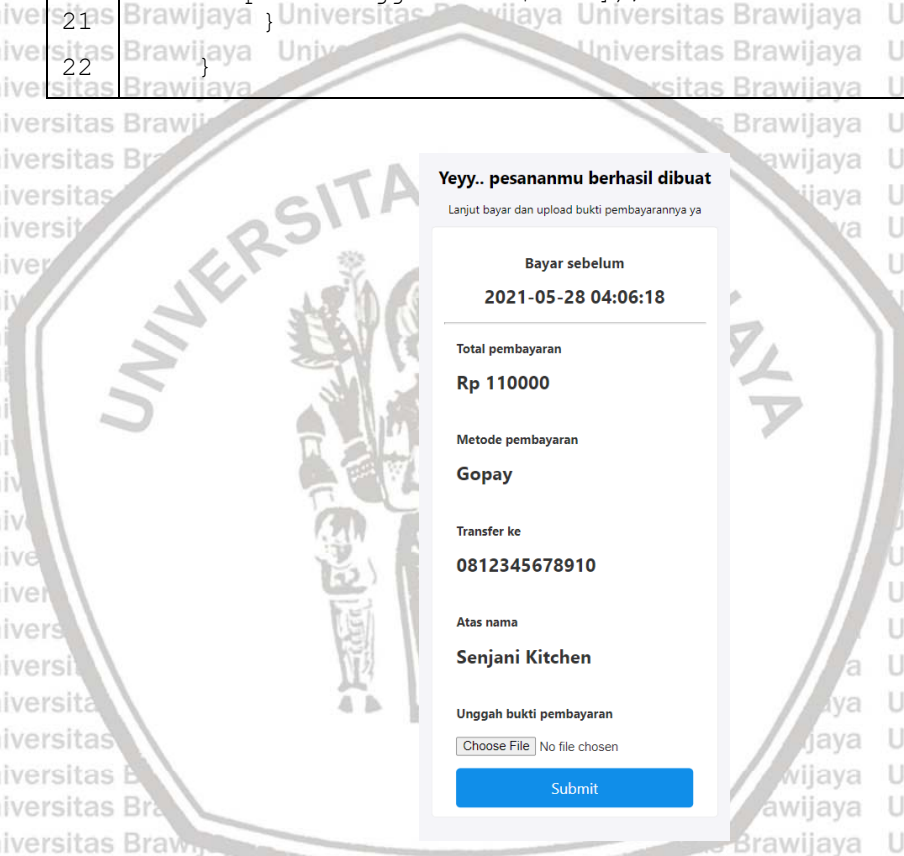
Tabel 5. 8 Kode Backend KuponPelanggan pada Controller Kupon Pelanggan

KuponPelanggan.php	
1	public function createKuponPelanggan()
2	{
3	\$data = \$this->request->getPost();
4	date_default_timezone_set('asia/jakarta');
5	\$kode_kupon_pelanggan = "KP" .
6	rand(0, 999) . "-PK" . \$data['kode_paket_kupon']
7	. "-D" . date("YmdHis");
8	unset(\$data['kode_paket_kupon']);
9	\$data['id_paket_kupon'] =
10	(int) \$data['id_paket_kupon'];
11	\$data['id_pelanggan'] = (int)
12	\$data['id_pelanggan'];
13	\$data['kode_kupon_pelanggan'] =
14	\$kode_kupon_pelanggan;
15	\$data['tanggal_pembelian_kupon'] = date("Y-m-
16	d H:i:s");
17	\$data['tanggal_kedaluwarsa'] = date("Y-m-d H:i:s",
18	strtotime("+1 year"));
19	\$data['waktu_batas_pembayaran'] =
20	date("Y-m-d H:i:s",
21	strtotime("+3 hour"));
22	\$data['jumlah_kupon_tersisa'] =




```

14 (int) $data['jumlah_kupon'];
15 unset($data['jumlah_kupon']);
16 $data['status_kupon'] = "belum_dibayar";
17 $data['created_by'] = 0;
18 $data['created_date'] = date("Y-m-d H:i:s");
19 if ($this->model->save($data)) {
20     $id_kupon_pelanggan =
21     $this->model->getInsertId();
22     return $this->respondCreated
    ([ 'id_kupon_pelanggan' =>
    $id_kupon_pelanggan, 'status' => 'success',
    'info' => 'create (kupon pelanggan)',
    'dataKuponPelanggan' => $data]);
    }
    }
    
```



Gambar 5. 59 Hasil Implementasi Unggah Bukti Pembayaran untuk Pelanggan

Pada halaman unggah bukti pembayaran Gambar 5. 59, sistem menampilkan total pembayaran, metode pembayaran yang telah dipilih, alamat transfer, dan batas tanggal & waktu pembayaran.

Tabel 5. 9 Kode Backend UpdateKuponPelanggan pada Controller Kupon Pelanggan

```

KuponPelanggan.php
1 public function updateKuponPelanggan ()
2 {
3     date_default_timezone_set('asia/jakarta');
    
```

```

4     helper(['form']);
5     $data = $this->request->getPost();
6     if (!$this->model->
7     >find($data['id_kupon_pelanggan'])) {
8         return $this->fail('id tidak ditemukan');
9     };
10    $file = $this->request->getFile('bukti_pembayaran');
11    $newName = "BP" . rand(0, 999) . date("YmDHis");
12    $newName = $newName . ".jpg";
13    $file->move('./assets/UploadBuktiPembayaran',
14    $newName);
15    $data['bukti_pembayaran'] = $newName;
16    $data['updated_by'] = 0;
17    $data['updated_date'] = date("Y-m-d H:i:s");
18    if ($this->model->save($data))
19    return $this->respondUpdated(['id_kupon_pelanggan' =>
20    $data['id_kupon_pelanggan'],
21    'status' => 'success', 'info' => 'update (kupon pelanggan)',
22    'data' => $data]);
23    }
24    }

```

5.2.2.6 Implementasi Menggunakan Kupon untuk Pelanggan

Aktivitas menggunakan kupon berfungsi agar pelanggan dapat menggunakan kupon yang telah dibeli sebelumnya. Dalam proses penggunaan kupon melibatkan halaman riwayat kupon, halaman jadwal menu, halaman form detail pengantaran, fungsi *showAllMenu* dalam *controller* menu, fungsi *createPesanan* dalam *controller* pesanan, model menu, dan model pesanan.

Gunakan Kupon

Jumlah kupon tersedia : 6

Kupon terpakai : 2

Lanjutkan Mengisi Form Pemesanan

2022-01-02 | Pagi



Asam Padeh Ikan

Terong Balado, Tempe Goreng, Dessert

2022-01-02 | Siang



Asam Padeh Ikan

Terong Balado, Tempe Goreng, Dessert

2022-01-02 | Sore



Orak-arik Telur Saus Teriyaki

Tumis Pokcoy Wortel, Mendoan, Dessert

Gambar 5. 60 Hasil Implementasi Pilih Menu untuk Pelanggan

Pada halaman pilih menu Gambar 5. 60, sistem menampilkan jadwal menu beserta spesifikasi, jumlah kupon tersedia dan kupon terpakai. Pelanggan memilih jadwal menu yang tersedia agar dapat melanjutkan ke form detail pengantaran atau pemesanan.

Tabel 5. 10 Kode Backend *showAllMenu* pada Controller Menu

Menu.php	
1	public function showAllMenu()
2	{
3	return \$this->respond(\$this->model->findAll());
4	}

< Form Pengantaran

Mohon isi sesuai dengan data penerima menu

Biodata Penerima

Nama Penerima Baariq Azhar

Alamat Jl. Candi 2C No.557 (Kos Rahman 99), Karangbesuki, Kec. Sukun, Kota Malang, Jawa Timur 65149

Nomor HP Penerima 087738210702

Catatan

Makanan Tidak suka pete

Pesanan Contoh: tidak ada/ ditaruh di satpam/ gantung di gerbang

Pesan

Gambar 5. 61 Hasil Implementasi Form Pengantaran untuk Pelanggan

Dalama halaman form pengantaran/ pemesanan Gambar 5. 61, pelanggan diharuskan mengisi semua informasi yang berkaitan tentang pengantaran dan penerima seperti, nama, alamat, nomor HP (whatsapp), catatan makanan, hingga catatan pesanan.

Tabel 5. 11 Kode Backend CreatePesanan pada Controller Pesanan

```

Pesanan.php
1 public function createPesanan()
2     {
3         date_default_timezone_set('asia/jakarta');
4         $data = $this->request->getPost();
5         $data['id_menu'] = explode(',', $data['ids_menu']);
6         $newData = [];
7         foreach ($data['id_menu'] as $item) {
8             $dataItem['id_pelanggan'] =
9             (int) $data['id_pelanggan'];
10            $dataItem['id_menu'] = (int) $item;
11            $dataItem['id_kupon_pelanggan'] =
12            (int) $data['id_kupon_pelanggan'];
13            $dataItem['kode_pesanan'] = "P" . rand(0, 999) . "-" .
14            "KP" . $data['id_kupon_pelanggan'] . "-D" . date("YmdHis");
15            $dataItem['waktu_pemesanan'] = date("Y-m-d H:i:s");
16            $dataItem['nama_penerima'] = $data['nama_penerima'];
17            $dataItem['no_hp_wa_penerima'] =
18            $data['no_hp_wa_penerima'];

```




```

15     $dataItem['alamat_penerima'] =
16     $data['alamat_penerima'];
17     $dataItem['alergi_makanan_penerima'] =
18     $data['alergi_makanan_penerima'];
19     $dataItem['status_pesanan'] = "belum dikirim";
20     $dataItem['catatan_pesanan'] =
21     $data['catatan_pesanan'];
22     $dataItem['created_by'] = 0;
23     $dataItem['created_date'] = date("Y-m-d H:i:s");
24     array_push($newData, $dataItem);
25     $this->model->save($dataItem);
26 }
27 $data = $newData;
28 return $this->respondCreated($data);
29 }

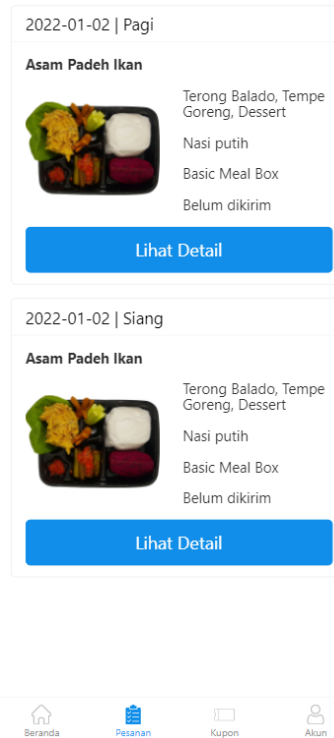
```

5.2.2.7 Implementasi Melihat Riwayat Pesanan untuk Pelanggan

Aktivitas melihat riwayat pesanan berfungsi agar pelanggan mendapatkan informasi tentang pesanan yang akan datang maupun yang telah selesai. Dalam proses melihat riwayat pesanan melibatkan tombol tab pesanan pada halaman utama, halaman riwayat pesanan, fungsi `showPesananBerdasarkanIdPelanggan` dalam *controller* mix, dan fungsi `getPesananKuponPelangganMenu` dalam model mix.



Riwayat Pesanan



Gambar 5. 62 Hasil Implementasi Riwayat Pesanan untuk Pelanggan

Halaman riwayat pesanan Gambar 5. 62 menampilkan informasi pesanan yang akan datang maupun yang telah selesai. Sistem menampilkan informasi pesanan berasal dari gabungan tabel pesanan, kupon pelanggan, paket kupon dan menu.

Tabel 5. 12 Kode Backend GetPesananKuponPelangganMenu pada Model Menu

```

MixModel.php
1 public function getPesananKuponPelangganMenu ($idPelanggan)
2 {
3     return $this->db->table('pesanan')
4         ->join('menu', 'menu.id_menu=pesanan.id_menu')
5         ->join('kupon_pelanggan', 'pesanan.
id kupon_pelanggan=kupon_pelanggan.id kupon_pelanggan')
6         ->join('paket_kupon', 'paket_kupon.
id paket_kupon=kupon_pelanggan.id paket_kupon')
7         ->getWhere(['pesanan.id_pelanggan' => $idPelanggan]);
8         >getResultArray();
9     }
    
```

Tabel 5. 13 Kode Backend ShowPesananBerdasarkanIdPelanggan pada Controller Mix

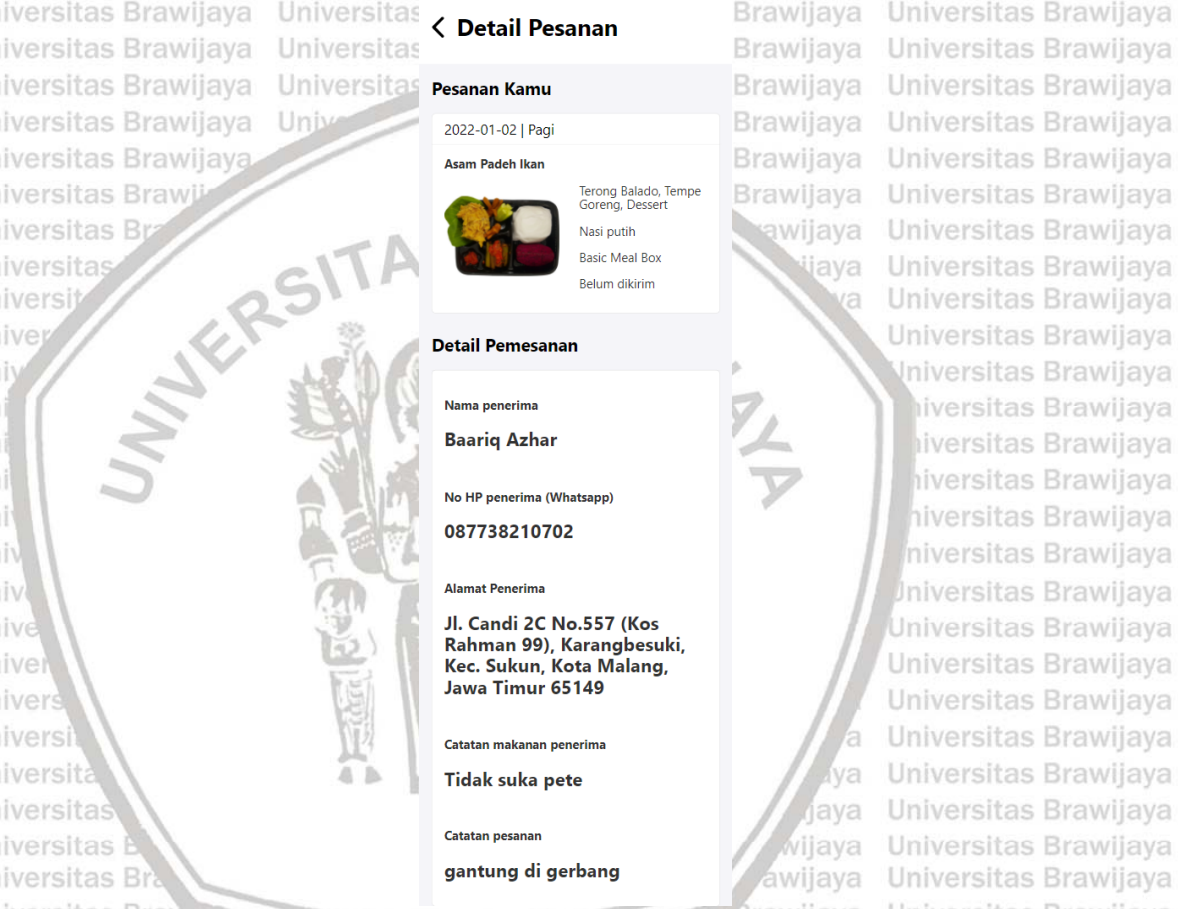
```

Mix.php
    
```



```

1 public function showPesananBerdasarkanIdPelanggan()
2 {
3     $dataRequest = $this->request->getPost();
4     $mixModel = new MixModel();
5     $data = $mixModel->
6     >getPesananKuponPelangganMenu($dataRequest['id_pelanggan']);
7     return $this->respond($data);
    }
    
```



Gambar 5. 63 Hasil Implementasi Detail Pesanan untuk Pelanggan

Halaman detail pesanan Gambar 5. 63 menampilkan informasi lebih lengkap tentang pesanan yang dipilih. Pelanggan mendapatkan informasi dari gabungan tabel pesanan, kupon pelanggan, paket kupon dan menu.

Tabel 5. 14 Kode Frontend DetailPesanan

```

DetailPesanan.jsx
1 function DetailPesanan() {
2     ...
3     const [apiMixPesanan, setApiMixPesanan] = useState(
4     JSON.parse(localStorage.getItem("apiMixPesanan"))
    )
    }
    
```

```
5   });
6   const [selectedIndexPesanan, setSelectedIndexPesanan] = use
State(
7     parseInt(localStorage.getItem("selectedIndexPesanan"))
8   );
9   return (
10    ...
11    <h1>Detail Pesanan</h1>
12    ...
13    <h2>Pesanan Kamu</h2>
14    <Card>
15    ...
16    <strong>{apiMixPesanan[selectedIndexPesanan].nama_menu}</st
rong>
17    <img src={` ${UrlApi}assets/fotoMenu/
${apiMixPesanan[selectedIndexPesanan].foto_menu}`
18    style={{ width: "90%" }}
19    />
20    ...
21    <p>{apiMixPesanan[selectedIndexPesanan].keterangan_menu}</p
>
22    <p>{jenisNasi(apiMixPesanan[selectedIndexPesanan].jen
is_nasi)}</p>
23    <p>{jenisPaketKupon(apiMixPesanan
[selectedIndexPesanan].jenis_paket_kupon)}</p>
24    <p>{statusPesanan(apiMixPesanan
[selectedIndexPesanan].status_pesanan)}</p>
25    ...
26    </Card>
27    <h2>Detail Pemesanan</h2>
28    <Card>
29    <Card.Body>
30    <h4>Nama penerima</h4>
31    <h2>{apiMixPesanan[selectedIndexPesanan].nama_penerima}</h2
>
32    <h4>No HP penerima (Whatsapp)</h4>
33    <h2>{apiMixPesanan[selectedIndexPesanan].no_hp_wa_penerima}
</h2>
34    <h4>Alamat Penerima</h4>
35    <h2>{apiMixPesanan[selectedIndexPesanan].alamat_penerima}
</h2>
36    <h4>Catatan makanan penerima</h4>
```



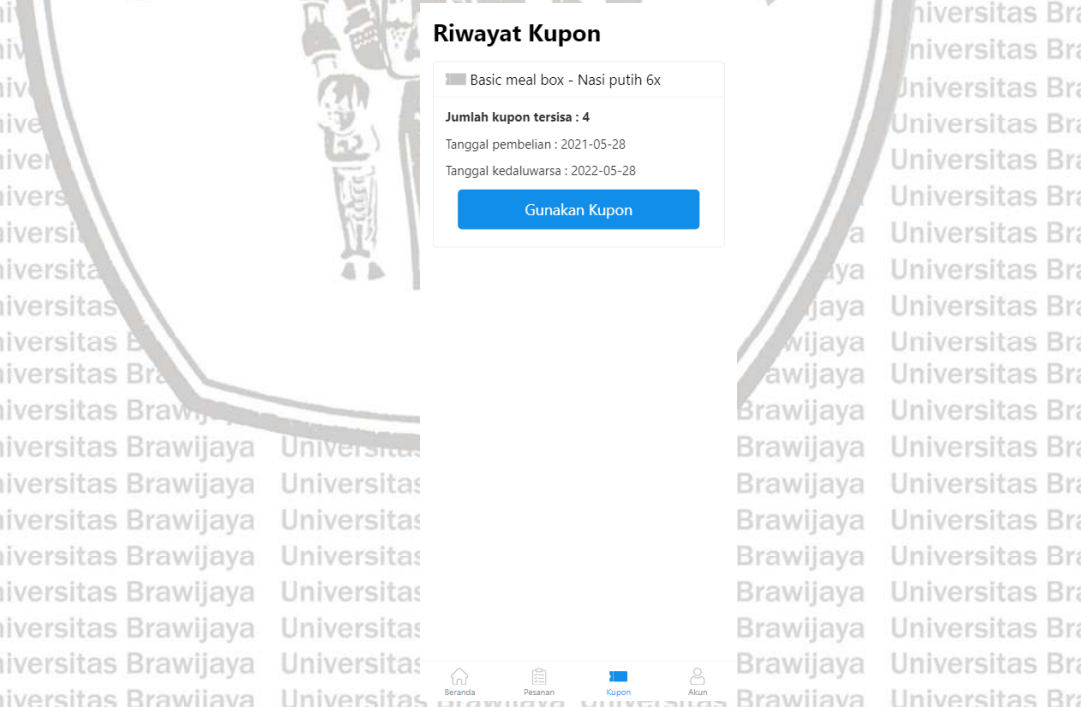
```

37 <h2>{apiMixPesanan[selectedIndexPesanan].alergi_makanan
38 _penerima}</h2>
39 <h4>Catatan pesanan</h4>
40 <h2>{apiMixPesanan[selectedIndexPesanan].catatan_pesanan}</
41 h2>
42 </Card.Body>
43 </Card>
44 ...

```

5.2.2.8 Implementasi Melihat Riwayat Kupon untuk Pelanggan

Halaman riwayat kupon Gambar 5. 64 berfungsi sebagai informasi bagi pelanggan tentang kupon yang telah mereka beli sebelumnya. Dalam riwayat kupon ini melibatkan tombol tab kupon pada halaman utama, halaman riwayat kupon, fungsi `showKuponPelangganBerdasarkanIdPelanggan` dalam `mix` controller, dan fungsi `getPaketKuponPelanggan` dalam model `mix`. Halaman riwayat kupon berfungsi memberi informasi kupon dan cara menggunakan kupon yang telah dibeli sebelumnya menggunakan tombol gunakan kupon. Fungsi `showKuponPelangganBerdasarkanIdPelanggan` mengambil data dan mengirimkan data kupon pelanggan ke halaman riwayat kupon. Fungsi `getPaketKuponPelanggan` dalam model `mix` berfungsi mengambil data dari database menggunakan `query`.



Gambar 5. 64 Hasil Implementasi Riwayat Kupon untuk Pelanggan

Tabel 5. 15 Kode Backend GetPaketKuponPelanggan pada Model Mix

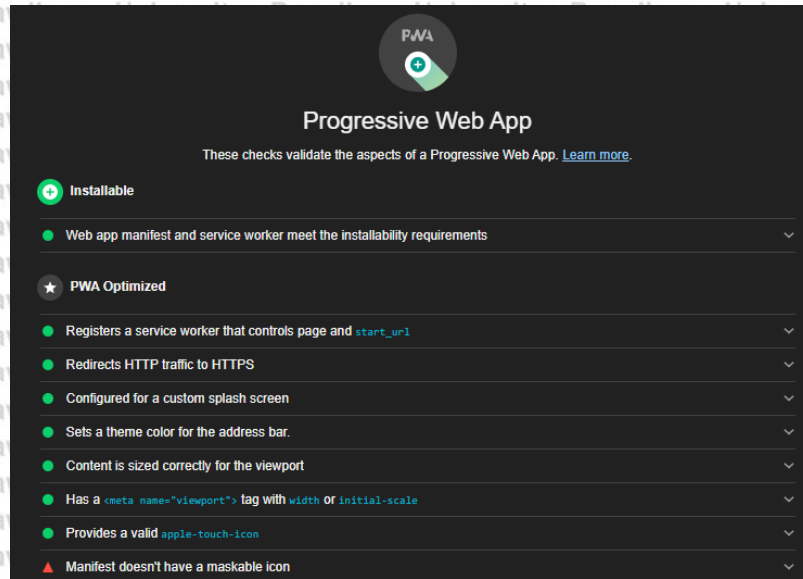
MixModel.php	
1	public function getPaketKuponPelanggan(\$idPelanggan)
2	{
3	return \$this->db->table('kupon_pelanggan')->
	join('paket_kupon','paket_kupon.id_paket_kupon=kupon_pelanggan.id_paket_kupon')->
	>getWhere(['id_pelanggan' => \$idPelanggan])
	>getResultArray();
4	}

Tabel 5. 16 Kode Backend ShowKuponPelangganBerdasarkanIdPelanggan pada Controller Mix

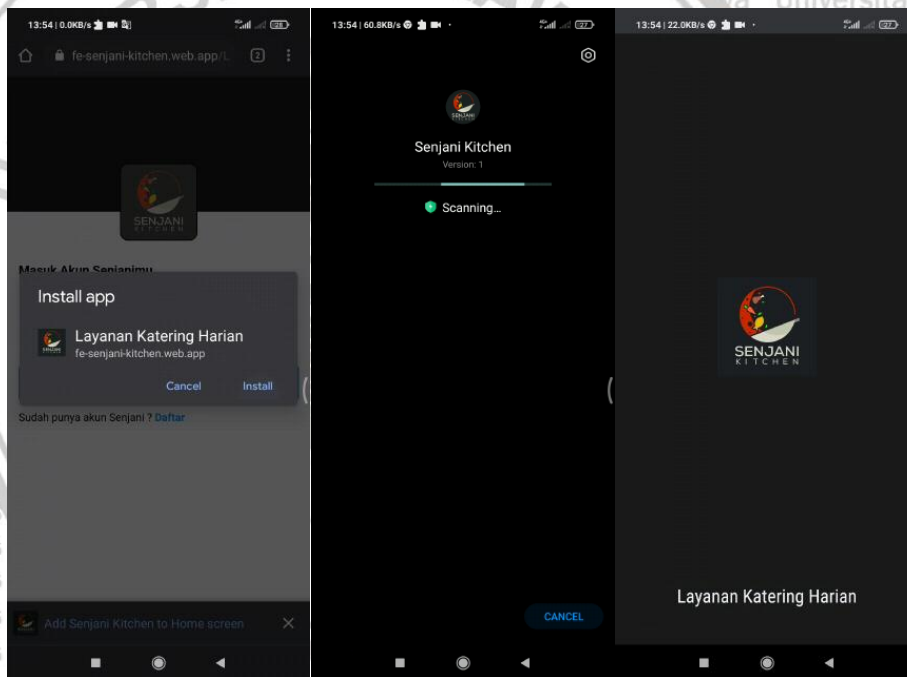
Mix.php	
1	public function showKuponPelangganBerdasarkanIdPelanggan()
2	{
3	\$dataRequest = \$this->request->getPost();
4	\$model = new MixModel();
5	\$data = \$model->
	getPaketKuponPelanggan(\$dataRequest['id_pelanggan']);
6	return \$this->respond(\$data);
7	}

5.2.2.9 Implementasi Progressive Web App untuk Pelanggan

PWA berfungsi agar pelanggan mendapatkan opsi memasang aplikasi web layaknya aplikasi *native* di perangkatnya. *Service worker* adalah *script* yang berjalan di *background* terpisah dari halaman web dan bagian dari *progressive web app*. Tangkapan layar Gambar 5. 65 membuktikan bahwa PWA berhasil diimplementasikan dan tangkapan layar Gambar 5. 66 percobaan melakukan pemasangan aplikasi web pada perangkat Android. Dalam implementasi PWA ini melibatkan *file manifest*, *index*, *service-worker*, dan *serviceWorkerRegistration*. Manifest dalam Tabel 5. 17 berisi informasi nama web, ikon dan ukuran yang digunakan, warna *default*, dan *splash screen*. Index dalam Tabel 5. 18 berfungsi sebagai pemanggilan registrasi *service worker*. *Service worker* berisi *script* untuk manajemen *file* yang ingin dimasukkan ke dalam *cache*. *Service worker registration* berfungsi sebagai pengecekan apakah perangkat yang digunakan bisa menjalankan *service worker* atau tidak.



Gambar 5. 65 Bukti Implementasi PWA untuk Pelanggan



Gambar 5. 66 Proses Pemasangan PWA untuk Pelanggan

Tabel 5. 17 Kode Frontend Manifest untuk Pelanggan

Manifest.json
1 {
2 "short_name": "Admin Senjani ",
3 "name": "Layanan Katering Harian",
4 "icons": [

```

5 {
6   "src": "favicon.ico",
7   "sizes": "64x64 32x32 24x24 16x16",
8   "type": "image/x-icon"
9 },
10 {
11   "src": "logo192.png",
12   "type": "image/png",
13   "sizes": "192x192"
14 },
15 {
16   "src": "logo512.png",
17   "type": "image/png",
18   "sizes": "512x512"
19 }
20 ],
21 "start_url": ".",
22 "display": "standalone",
23 "theme_color": "#000000",
24 "background_color": "#ffffff"
25 }

```

Tabel 5. 18 Kode *Frontend Index* untuk Pelanggan

```

Index.js
1 ...
2 import * as serviceWorkerRegistration from
  './serviceWorkerRegistration';
3 ...
4 serviceWorkerRegistration.register();

```

5.2.2.10 Implementasi *Login* untuk Admin

Halaman *login* Gambar 5. 67 berfungsi sebagai proses autentikasi akun admin yang sebelumnya telah terdaftar. Dalam proses masuk pelanggan melibatkan halaman *login*, halaman utama, fungsi *login()* dalam *controller* admin, dan model admin. Halaman *login* berfungsi agar admin dapat memasukkan *email* dan *password* yang telah terdaftar sebelumnya. Setelah admin berhasil memasukkan *email* dan *password*, admin akan masuk ke dalam halaman utama Gambar 5. 68. Fungsi *login()* yang berada dalam *controller* admin berfungsi memvalidasi data masukan ke dalam tabel admin. Model admin berfungsi menjembatani *controller* admin dengan *database*.



Gambar 5. 67 Hasil Implementasi *Login* untuk Admin



Gambar 5. 68 Hasil Implementasi Halaman Utama untuk Admin

Tabel 5. 19 Kode *Backend* login untuk Admin

```

Admin.php
1 public function login()
2 {
3     $dataInput = $this->request->getPost();
4     $dataDatabase = $this->model->where('email_admin',
5     $dataInput['email_admin'])->findAll();
6     if (count($dataDatabase) == 0) {

```

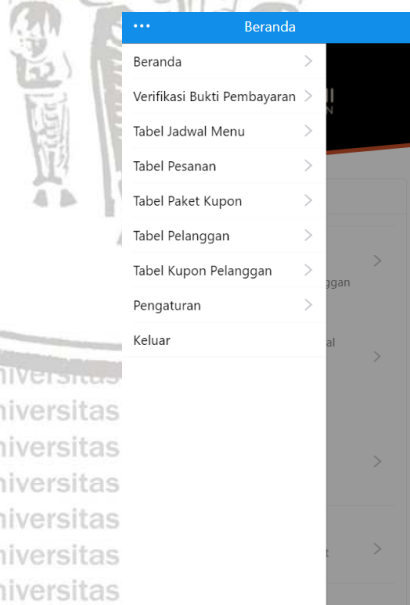
```

6         return $this->respond(['id_admin'=>'', 'status'
=>'failed', 'info'=>'email is not registered', 'dataAdmin'
=>'']);
7     } else {
8         if (md5($dataInput['password']) ==
$dataDatabase[0]['password']) {
9             return $this->respond(['id_admin'=>
$dataDatabase[0]['id_admin'], 'status'=>'success', 'info'
=>'email & password are correct', 'dataAdmin'=>
$dataDatabase[0]]);
10        } else {
11            return $this->respond(['id_admin'=>
$dataDatabase[0]['id_admin'], 'status'=>'failed', 'info'=>
'email is correct, password is incorrect', 'dataAdmin'=>
$dataDatabase[0]]);
12        }
13    }
14 }

```

5.2.2.11 Implementasi Logout untuk Admin

Halaman *logout* Gambar 5. 69 berfungsi agar admin dapat keluar dari sistem yang berjalan. Dalam proses *logout* melibatkan halaman utama, tombol keluar dalam *drawerComponents*, dan halaman *login*. Tabel 5. 20 berisi kode *frontend* untuk menampilkan tombol keluar dan fungsi *onClickKeluar* untuk mengeluarkan admin dari sistem yang sedang berjalan.



Gambar 5. 69 Hasil Implementasi Logout untuk Admin

Tabel 5. 20 Kode Frontend DrawerComponents

```

DrawerComponents.jsx
1  ...
2  const onClickKeluar = () => {
3    localStorage.clear();
4    history.push("/Login");
5  };
6
7  ...
8  const Sidebar = (
9    <List>
10   <List.Item
11     arrow="horizontal" onClick={() => {
12       history.push("/"); }}
13   >
14     Beranda
15   </List.Item>
16   ...
17   <List.Item
18     onClick={() => {
19       onClickKeluar();
20     }}
21   >
22     Keluar
23   </List.Item>
24 </List>
25 );
26 ...

```

5.2.2.12 Implementasi Verifikasi Bukti Pembayaran untuk Admin

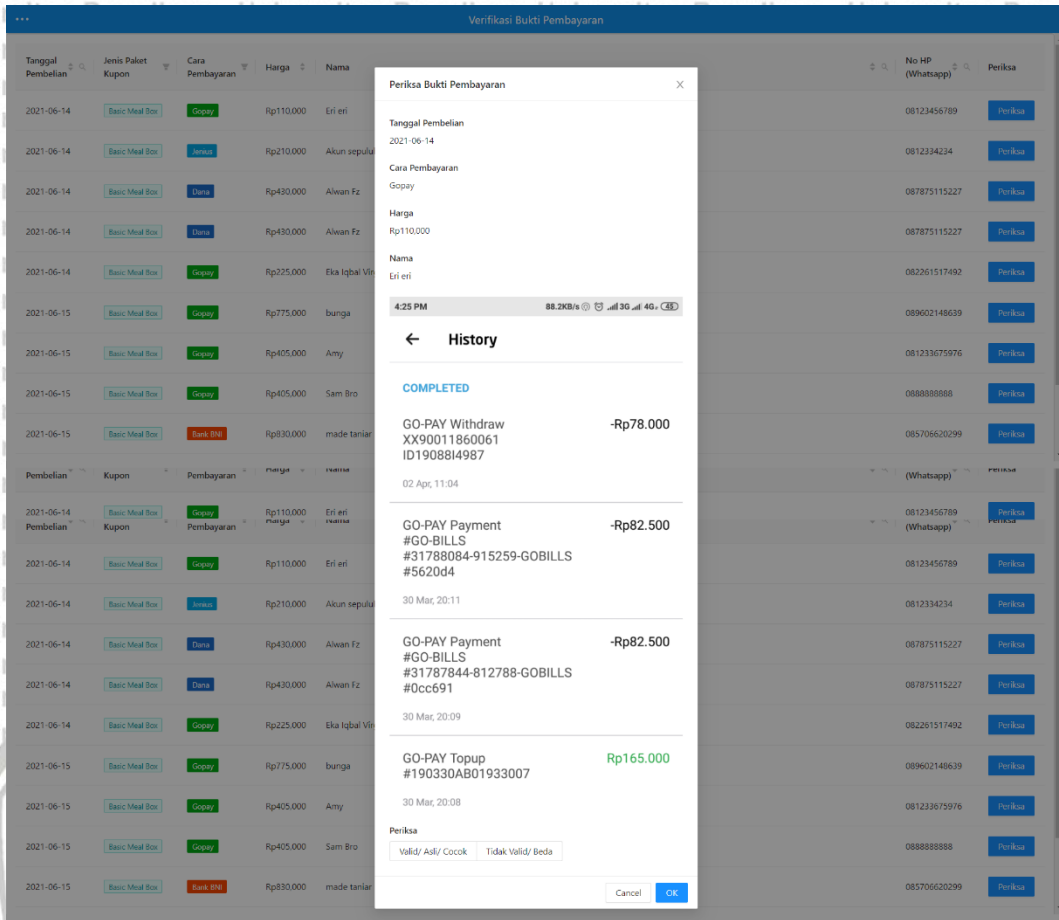
Halaman tabel bukti pembayaran Gambar 5. 70 berfungsi agar admin mendapatkan informasi dan melakukan verifikasi bukti pembayaran yang telah diunggah oleh pelanggan dalam modal form Gambar 5. 71. Aktivitas ini berfungsi agar admin bisa menganalisis keaslian dari foto bukti pembayaran yang diunggah oleh pelanggan. Dalam proses verifikasi bukti pembayaran melibatkan tombol “Verifikasi bukti pembayaran” dalam halaman utama dan menu sidebar, halaman tabel bukti pembayaran, modal periksa bukti pembayaran, fungsi showBuktiPembayaranStatusMenungguDiverifikasi() pada controller mix untuk menampilkan semua bukti pembayaran yang belum diverifikasi, fungsi adminUpdateKuponPelangganNoFoto() pada cotroller kupon pelanggan untuk

melakukan perubahan status kupon pelanggan, fungsi `getKuponPelangganPelangganPaketKuponStatusMenunggu()` dalam model `mix` untuk melakukan pengambilan data dengan `query`, dan model kupon pelanggan.

Tanggal Pembelian	Jenis Paket Kupon	Cara Pembayaran	Harga	Nama	No HP (Whatsapp)	Periksa
2021-06-14	Basic Meal Box	Gopay	Rp110.000	Eri eri	08123456789	Periksa
2021-06-14	Basic Meal Box	Jenius	Rp210.000	Akun sepuluh	081234234	Periksa
2021-06-14	Basic Meal Box	Dana	Rp430.000	Aliwan Fz	087875115227	Periksa
2021-06-14	Basic Meal Box	Dana	Rp430.000	Aliwan Fz	087875115227	Periksa
2021-06-14	Basic Meal Box	Gopay	Rp225.000	Eka Iqbal Virgilawan	082261517492	Periksa
2021-06-15	Basic Meal Box	Gopay	Rp775.000	Ibanga	089602148639	Periksa
2021-06-15	Basic Meal Box	Gopay	Rp405.000	Amy	081233675976	Periksa
2021-06-15	Basic Meal Box	Gopay	Rp405.000	Sam Bro	088888888	Periksa
2021-06-15	Basic Meal Box	Bank BNI	Rp830.000	made tanjar	085706620299	Periksa
2021-06-15	Basic Meal Box	Bank BNI	Rp775.000	made tanjar	085706620299	Periksa

Gambar 5. 70 Hasil Implementasi Tampilan Tabel Bukti Pembayaran untuk Admin





Gambar 5. 71 Hasil Implementasi Tampilan Form Verifikasi untuk Admin

Tabel 5. 21 Kode Backend Model Kupon Pelanggan

```
MixModel.php
1 public function
  getKuponPelangganPelangganPaketKuponStatusMenunggu()
2 {
3     return $this->db->table('kupon_pelanggan')
4         ->join('paket_kupon',
5             'paket_kupon.id_paket_kupon=kupon_pelanggan.id_paket_kupon')
6         ->join('pelanggan',
7             'pelanggan.id_pelanggan=kupon_pelanggan.id_pelanggan')
8         ->getWhere(['kupon_pelanggan.status_kupon'=>
9             'menunggu_diverifikasi'])->getResultArray();
10 }
```

Tabel 5. 22 Kode Backend showBuktiPembayaranStatusMenungguDiverifikasi

```
Mix.php
1 public function
  showBuktiPembayaranStatusMenungguDiverifikasi()
2 {
```

```

3     $mixModel = new MixModel();
4     $data      = $mixModel->
>getKuponPelangganPelangganPaketKuponStatusMenunggu();
5     return $this->respond($data);
6     }

```

Tabel 5. 23 Kode Backend adminUpdateKuponPelanggan

```

KuponPelanggan.php
1 public function adminUpdateKuponPelangganNoFoto()
2 {
3     date_default_timezone_set('asia/jakarta');
4     helper(['form']);
5
6     $dataInput = $this->request->getPost();
7     $dataInput['updated_by'] = 10;
8     $dataInput['updated_date'] = date("Y-m-d H:i:s");
9
10    if ($this->model->save($dataInput)) {
11        $dataMenuFromDatabase = $this->model->
>find($dataInput['id_kupon_pelanggan']);
12        return $this->respond(['id_kupon_pelanggan' =>
13        $dataInput['id_kupon_pelanggan'], 'status' => 'success',
14        'info' => 'update kupon pelanggan successfully',
15        'dataKuponPelanggan' => $dataMenuFromDatabase]);
16    }
17 }

```

5.2.2.13 Implementasi Membuat Jadwal Menu Baru untuk Admin

Membuat jadwal menu baru hanya dapat dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin dapat menambahkan menu baru beserta spesifikasinya yaitu tanggal & waktu menu dikirimkan, nama menu, keterangan, dan foto menu. Halaman Tabel jadwal menu Gambar 5. 72 berfungsi agar admin mendapatkan informasi bukti pembayaran yang telah diunggah oleh pelanggan namun belum diverifikasi. *Form* tambah menu baru Gambar 5. 73 bertujuan agar admin dapat menambahkan menu baru beserta spesifikasinya (tanggal& waktu, nama menu, keterangan, lauk tambahan, dan foto). Dalam proses ini melibatkan halaman tabel bukti pembayaran yang belum diverifikasi, fungsi showAllMenu() Tabel 5. 24 dalam *controller* menu menampilkan semua data dalam tabel Menu, fungsi createMenu() Tabel 5. 25 dalam *controller* menu untuk membuat data menu baru, dan model menu untuk menghubungkan *controller* menu dengan *database*.

Tanggal	Waktu	Nama	Keterangan Menu	Lauk Tambahan	Foto Menu	Edit
2022-01-02	Pagi	Asam Padéh Ikan	Terong Balado, Tempe Goreng, Dessert	Tempe Goreng		
2022-01-02	Siang	Asam Padéh Ikan	Terong Balado, Tempe Goreng, Dessert	Tempe Goreng		
2022-01-02	Sore	Orak-arik Telur Saus Teriyaki	Tamis Pkcozy Wortel, Mendoban, Dessert	Tahu Goreng		
2022-01-03	Pagi	Dori Sambal Matah	Kangkung Masak Santan, Bakwan Tahu, Dessert	Tahu Krispy		
2022-01-03	Siang	Chicken Crispy	Semur Kacang Panjang, Tempe Goreng, Dessert	Tahu Krispy		
2022-01-03	Sore	Orak-arik Telur S Tiram	Sup Jagung, Perkedel, Dessert	Tempe Kripi		
2022-01-04	Pagi	Usus Bumbu Kuning	Pecak Terong, Mendol, Dessert	Dadar Jagung		
2022-01-04	Siang	Usus Bumbu Kuning	Pecak Terong, Mendol, Dessert	Dadar Jagung		
2022-01-04	Sore	Rolade Ayam Lidang	Sayur Kelor Bening, Tahu Bumbu Kecap, Dessert	Dadar Jagung		
2022-01-05	Pagi	Ikan Gurami Panggang	Sayur sopi, Buah Anggur	Tempe Oseng		

Gambar 5. 72 Hasil Implementasi Tampilan Tabel Menu untuk Admin

Tambah Menu Baru

Tanggal Menu:

Waktu Menu: Pagi Siang Sore

Nama Menu:

Keterangan Menu:

Lauk Tambahan:

Unggah Foto: No file chosen

Gambar 5. 73 Hasil Implementasi Tampilan Form Tambah Menu untuk Admin

Tabel 5. 24 Kode Backend showAllMenu pada Controller Menu

```

Menu.php
1 public function showAllMenu()
2 {
3     return $this->respond($this->model->findAll());
4 }
    
```

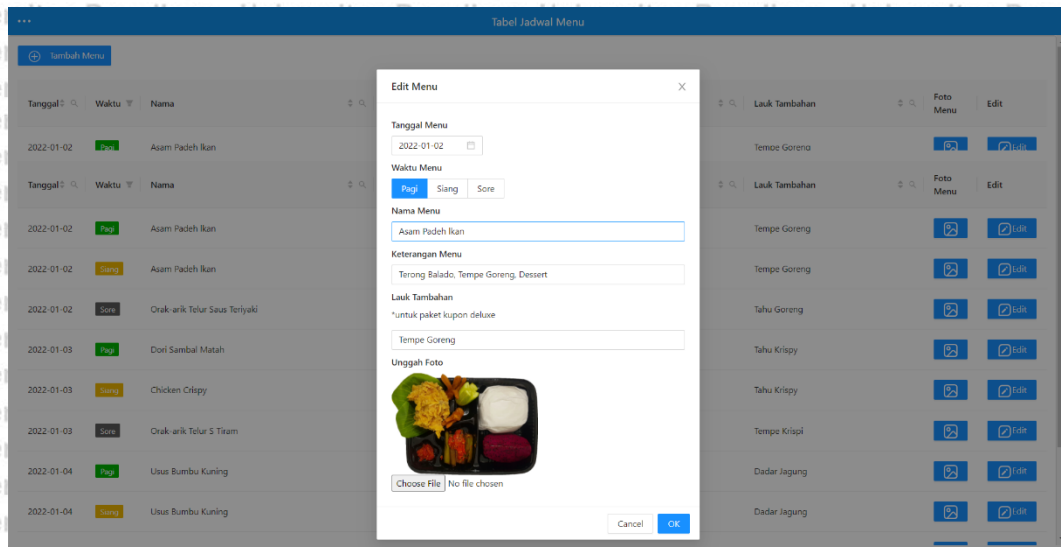
Tabel 5. 25 Kode Backend createMenu pada Controller Menu

```

Menu.php
1 public function createMenu()
2 {
3     date_default_timezone_set('asia/jakarta');
4     helper(['form']);
5
6     $dataInput = $this->request->getPost();
7     $file = $this->request->getFile('foto_menu');
8     $newName = "FM" . rand(0, 999) . date("YmdHis");
9     $newName = $newName . ".jpg";
10    $file->move('./assets/fotoMenu', $newName);
11
12    $dataInput['foto_menu'] = $newName;
13    $dataInput['created_by'] = 10;
14    $dataInput['created_date'] = date("Y-m-d H:i:s");
15
16    if ($this->model->save($dataInput)) {
17        $idMenu = $this->model->getInsertId();
18        $dataMenuFromDatabase = $this->model->
19        >find($idMenu);
20        return $this->respond(['id_menu' => $idMenu,
21        'status' => 'success', 'info' => 'new menu insert
22        successfully', 'dataMenu' => $dataMenuFromDatabase]);
23    }
24 }
    
```

5.2.2.14 Implementasi Mengedit Jadwal Menu untuk Admin

Mengedit jadwal menu hanya bisa dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin dapat melakukan perubahan data pada jadwal menu yang telah tersimpan. *Form* edit menu berfungsi agar admin dapat mengubah data yang telah tersimpan sebelumnya di *database*. Proses ini melibatkan *form* edit menu Gambar 5. 74, fungsi `updateMenu` Tabel 5. 26 untuk melakukan perubahan data sesuai dengan permintaan admin, dan model menu untuk menghubungkan *controller* menu dengan *database*.



Gambar 5. 74 Hasil Implementasi *Form* Edit untuk Admin

Tabel 5. 26 Kode *Backend* updateMenu pada *Controller* Menu

```

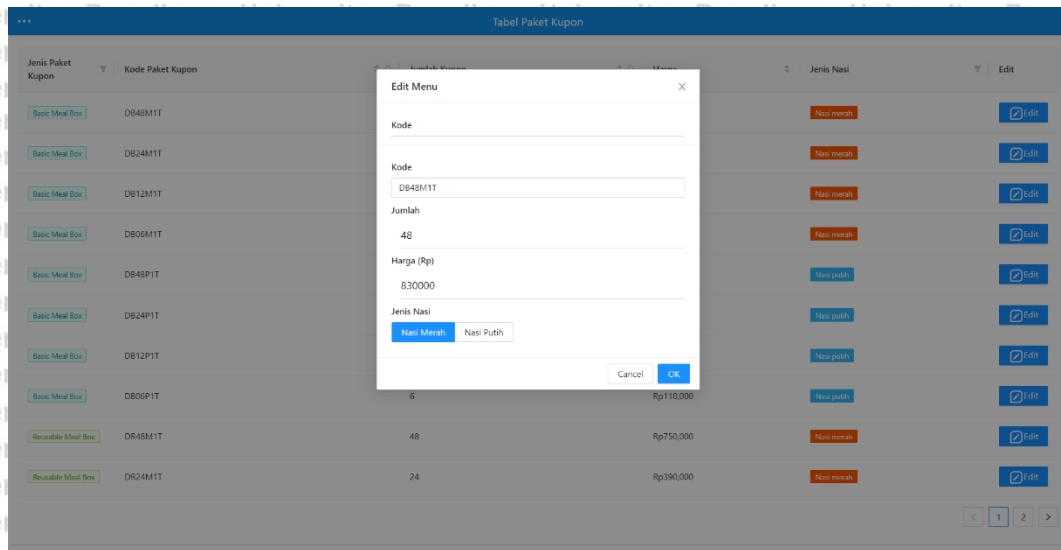
Menu.php
1 public function updateMenu()
2 {
3     date_default_timezone_set('asia/jakarta');
4     helper(['form']);
5
6     $dataInput = $this->request->getPost();
7     $file = $this->request->getFile('foto_menu');
8     $newName = "FM" . rand(0, 999) . date("YmdHis");
9     $newName = $newName . ".jpg";
10    $file->move('./assets/fotoMenu', $newName);
11
12    $dataInput['foto_menu'] = $newName;
13    $dataInput['updated_by'] = 10;
14    $dataInput['updated date'] = date("Y-m-d H:i:s");
15
16    if ($this->model->save($dataInput)) {
17        $dataMenuFromDatabase = $this->model->
18        >find($dataInput['id_menu']);
19        return $this->respond(['id_menu' =>
20        $dataInput['id_menu'], 'status' => 'success', 'info' =>
21        'update menu successfully', 'dataMenu' =>
22        $dataMenuFromDatabase]);
23    }
24 }
    
```

5.2.2.15 Implementasi Mengedit Paket Kupon untuk Admin

Mengedit paket kupon hanya bisa dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin dapat melakukan perubahan data pada tabel paket kupon. Halaman Gambar 5. 75 berfungsi agar admin mendapatkan informasi paket kupon yaitu kode, jenis, jumlah kupon, jenis nasi, lauk tambahan, dan harga dan halaman Gambar 5. 76 berfungsi agar admin dapat mengedit data paket kupon yang telah tersimpan di *database*. Dalam proses ini melibatkan halaman tabel paket kupon, *form* edit, fungsi `showAllPaketKupon` Tabel 5. 27 dalam *controller* paket kupon berfungsi untuk menampilkan semua data dari tabel Paket Kupon, fungsi Tabel 5. 28 dalam *controller* `updatePaketKupon` dalam *controller* paket kupon berfungsi untuk mengubah data sesuai dengan masukan admin, dan model paket kupon untuk menghubungkan *controller* dengan *database*.

Jenis Paket Kupon	Kode Paket Kupon	Jumlah Kupon	Harga	Jenis Nasi	Edit
Basic Meal Box	D048M1T	48	Rp830.000	Nasi merah	Edit
Basic Meal Box	D024M1T	24	Rp430.000	Nasi merah	Edit
Basic Meal Box	D012M1T	12	Rp225.000	Nasi merah	Edit
Basic Meal Box	D006M1T	6	Rp120.000	Nasi merah	Edit
Basic Meal Box	D048P1T	48	Rp775.000	Nasi putih	Edit
Basic Meal Box	D024P1T	24	Rp405.000	Nasi putih	Edit
Basic Meal Box	D012P1T	12	Rp210.000	Nasi putih	Edit
Basic Meal Box	D006P1T	6	Rp110.000	Nasi putih	Edit
Reusable Meal Box	D048M1T	48	Rp750.000	Nasi merah	Edit
Reusable Meal Box	D024M1T	24	Rp390.000	Nasi merah	Edit

Gambar 5. 75 Hasil Implementasi Tabel Paket Kupon untuk Admin



Gambar 5. 76 Hasil Implementasi *Form* Edit Menu

Tabel 5. 27 Kode *Backend* showAllPaketKupon pada *Controller* Paket Kupon

```
PaketKupon.php
1 public function showAllPaketKupon()
2 {
3     return $this->respond($this->model->findAll());
4 }
```

Tabel 5. 28 Kode *Backend* updatePaketKupon pada *Controller* Paket Kupon

```
PaketKupon.php
1 public function updatePaketKupon()
2 {
3     date_default_timezone_set('asia/jakarta');
4
5     $dataInput = $this->request->getPost();
6     $dataInput['updated_by'] = 10;
7     $dataInput['updated_date'] = date("Y-m-d H:i:s");
8
9     if ($this->model->save($dataInput)) {
10         $dataFromDatabase = $this->model->find($dataInput['id_paket_kupon']);
11         return $this->respond(['id_menu' => $dataInput['id_paket_kupon'], 'status' => 'success', 'info' => 'update paket kupon successfully', 'dataPaketKupon' => $dataFromDatabase]);
12     }
13 }
```

5.2.2.16 Implementasi Melihat Hasil Pesanan untuk Admin

Melihat hasil pesanan hanya dapat dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin mendapatkan informasi penukaran kupon dengan menu yang dilakukan oleh pelanggan. Admin mendapatkan informasi kupon, yaitu nama pelanggan, jenis paket kupon, tanggal & waktu pengiriman, biodata penerima, catatan makanan, catatan pengiriman, dan status pesanan. Dalam tabel ini hanya menampilkan pesanan dengan kupon yang telah dibayar. Dalam proses ini halaman tabel pesan Gambar 5. 77 berfungsi agar admin mendapatkan informasi pesanan yang telah dilakukan oleh pelanggan, fungsi `showAllPesanan()` Tabel 5. 29 dari *controller* pesanan berfungsi untuk menampilkan semua data dari tabel pesanan, dan fungsi `getKuponPelangganPelangganPaketKuponStatusMenunggu` dari model *mix* untuk melakukan *query* mencari pesanan yang telah dibayar saja.

Tanggal	Waktu	Status Pesanan	Jenis Paket Kupon	Jenis Nasi	Nama Penerima	No HP (Whatsapp)	Alamat Penerima	Catatan Makanan Penerima	Catatan Pengantaran	Edit
2022-01-03	Pagi	Belum dikirim	Basic Meal Box	Nasi putih	Amy	081233675976	Jalan Sunan Kalijaga 15	Alergi kacang	Ditaruh di meja taman kos	Edit
2022-01-03	Sore	Belum dikirim	Basic Meal Box	Nasi putih	Amy	081233675976	Jalan Sunan Kalijaga 15	Alergi kacang	Ditaruh di meja taman kos	Edit
2022-01-04	Pagi	Belum dikirim	Basic Meal Box	Nasi putih	Amy	081233675976	Jalan Sunan Kalijaga 15	Alergi kacang	Ditaruh di meja taman kos	Edit
2022-01-04	Sore	Belum dikirim	Basic Meal Box	Nasi putih	Amy	081233675976	Jalan Sunan Kalijaga 15	Alergi kacang	Ditaruh di meja taman kos	Edit
2022-01-05	Pagi	Belum dikirim	Basic Meal Box	Nasi putih	Amy	081233675976	Jalan Sunan Kalijaga 15	Alergi kacang	Ditaruh di meja taman kos	Edit
2022-01-02	Pagi	Belum dikirim	Basic Meal Box	Nasi putih	made tanlar	085706620299	Jl. Kemantren no 24	tidak suka kacang, tidak suka ikan2an-udang	titip ke kucing	Edit
2022-01-02	Pagi	Belum dikirim	Basic Meal Box	Nasi putih	made tanlar	085706620299	Jl. Kemantren no 24	tidak suka kacang, tidak suka ikan2an-udang	titip ke kucing	Edit
2022-01-02	Pagi	Belum dikirim	Basic Meal Box	Nasi putih	made tanlar	085706620299	Jl. Kemantren no 24	tidak suka kacang, tidak suka ikan2an-udang	pk rmtg	Edit
2022-01-02	Siang	Belum dikirim	Basic Meal Box	Nasi putih	made tanlar	085706620299	Jl. Kemantren no 24	tidak suka kacang, tidak suka ikan2an-udang	pk rmtg	Edit
2022-01-02	Sore	Sudah dikirim	Basic Meal Box	Nasi putih	made tanlar	085706620299	Jl. Kemantren no 24	tidak suka kacang, tidak suka ikan2an-udang	pk rmtg	Edit

Gambar 5. 77 Hasil Implementasi Tabel Pesanan untuk Admin

Tabel 5. 29 Kode *Backend* `showAllPesanan` pada *Controller* Pesanan

```

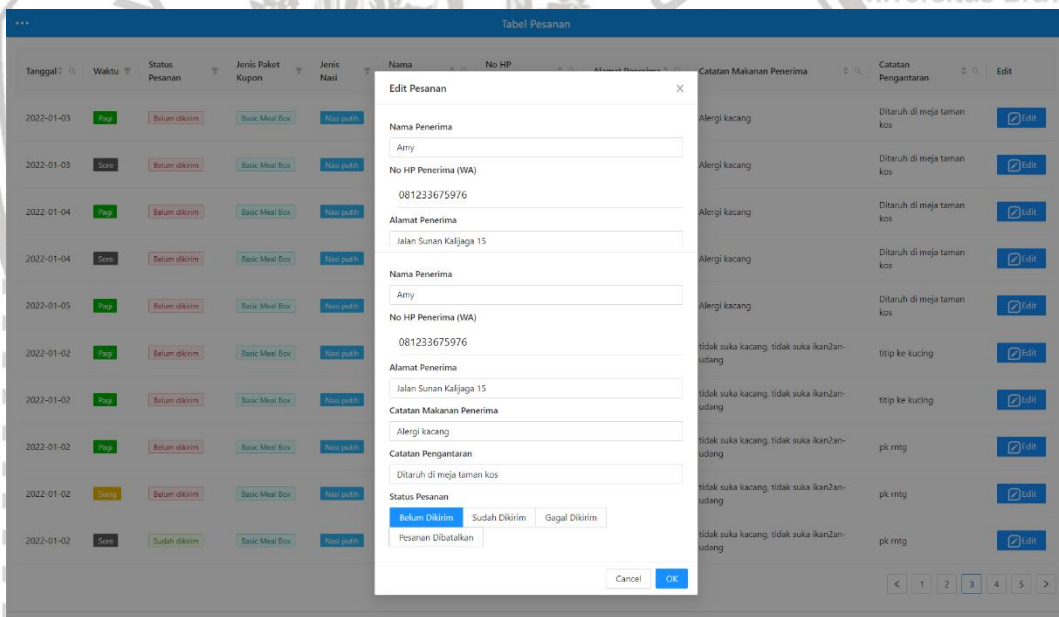
Pesanan.php
1 public function showAllPesanan()
2 {
3     $mixModel = new MixModel();
4     $data=$mixModel->
5     getKuponPelangganPelangganPaketKuponStatusMenunggu ( );
6     return $this->respond($data);
7 }
    
```



```
MixModel.php
1 public function getKuponPelangganPelangganPaketKuponStatusMenunggu()
2 {
3     return $this->db->table('kupon_pelanggan')
4         ->join('paket_kupon',
5             'paket_kupon.id_paket_kupon=kupon_pelanggan.id_paket_kupon')
6         ->join('pelanggan',
7             'pelanggan.id_pelanggan=kupon_pelanggan.id_pelanggan')
8         ->getWhere(['kupon_pelanggan.status_kupon' =>
9             'menunggu_diverifikasi'])->getResultArray();
10 }
```

5.2.2.17 Implementasi Mengedit Pesanan untuk Admin

Mengedit pesanan hanya bisa dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin dapat melakukan perubahan data pada tabel pesanan yang telah diinput oleh pelanggan dengan cara penukaran kupon dengan menu. Proses ini melibatkan *form* edit pesanan Gambar 5. 78, fungsi `adminUpdatePesanan` Tabel 5. 30 untuk melakukan perubahan data sesuai dengan permintaan admin, dan model pesanan untuk menghubungkan *controller* pesanan dengan *database*.



Gambar 5. 78 Hasil Implementasi *Form* Edit untuk Admin

Tabel 5. 30 Kode *Backend* `adminUpdatePesanan` pada *Controller* Pesanan

```
Pesanan.php
1 public function adminUpdatePesanan()
2 {
3     date_default_timezone_set('asia/jakarta');
4
5     $dataInput = $this->request->getPost();
6     $dataInput['updated by'] = 10;
```

```

7      $dataInput['updated_date'] = date("Y-m-d H:i:s");
8
9      if ($this->model->save($dataInput)) {
10         $dataMenuFromDatabase = $this->model-
>find($dataInput['id_pesanan']);
11         return $this->respond(['id_pesanan' =>
$dataInput['id_pesanan'], 'status' => 'success', 'info' =>
'update pesanan successfully', 'dataPesanan' =>
$dataMenuFromDatabase]);
12     }
13 }

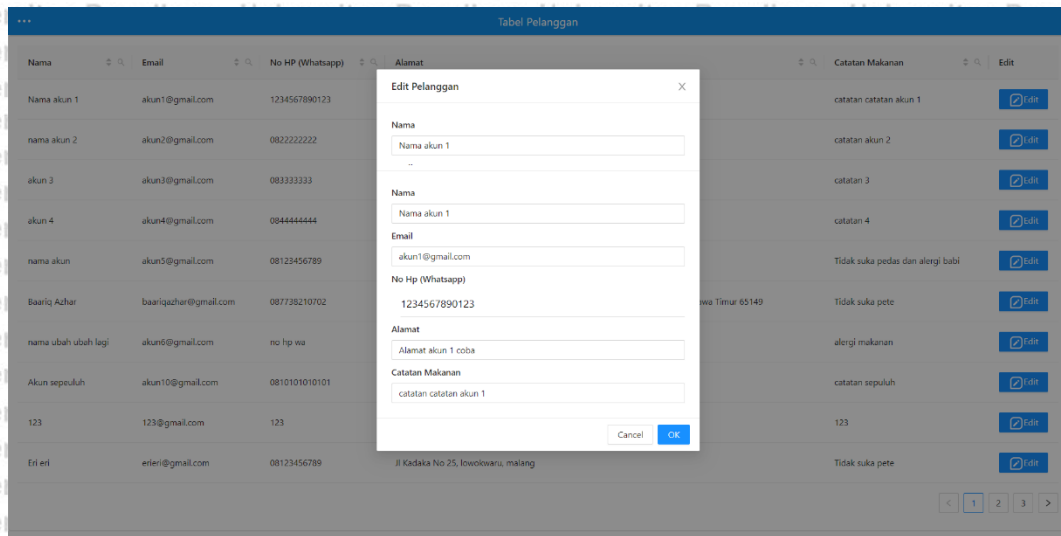
```

5.2.2.18 Implementasi Mengedit Pelanggan untuk Admin

Mengedit pelanggan hanya bisa dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin dapat melakukan perubahan data pada tabel pelanggan yang diinput oleh pelanggan saat pendaftaran. Halaman tabel pelanggan Gambar 5. 79 berfungsi agar admin mendapatkan informasi pelanggan. Jika admin ingin mengubah data pelanggan dapat menekan tombol “edit” lalu akan muncul *form* edit Gambar 5. 80. Dalam proses ini melibatkan, yaitu fungsi `showAllPelanggan()` Tabel 5. 31 dalam *controller* pelanggan berfungsi untuk menampilkan semua data dari tabel pelanggan, `updatePelanggan()` Tabel 5. 32 dalam *controller* pelanggan berfungsi untuk menyimpan perubahan data berdasarkan id pelanggan, dan model pelanggan sebagai penghubung antara *controller* pelanggan dengan database.

Nama	Email	No HP (Whatsapp)	Alamat	Catatan Makanan	Edit
Nama akun 1	akun1@gmail.com	1234567890123	Alamat akun 1 coba	catatan catatan akun 1	Edit
nama akun 2	akun2@gmail.com	0822222222	alamat akun 2	catatan akun 2	Edit
akun 3	akun3@gmail.com	083333333	alamat 3	catatan 3	Edit
akun 4	akun4@gmail.com	0844444444	alamat	catatan 4	Edit
nama akun	akun5@gmail.com	08123456789	Jl alamat bisa harus bisa akun no 5. Malang, Jawa Timur	Tidak suka pedas dan alergi babi	Edit
Baariq Azhar	baariqazhar@gmail.com	087738270702	Jl. Candi 2C No.557 (Kos Rahman 99), Karangbesuki, Kec. Sukun, Kota Malang, Jawa Timur 65149	Tidak suka pete	Edit
nama ubah lagi	akun6@gmail.com	no hp wa	alamat	alergi makanan	Edit
Akun sepuluh	akun10@gmail.com	0810101010101	alamat sepuluh	catatan sepuluh	Edit
123	123@gmail.com	123	123	123	Edit
Eri eri	erieri@gmail.com	08123456789	Jl Kadaka No 25, lowowaru, malang	Tidak suka pete	Edit

Gambar 5. 79 Hasil Implementasi Tabel Pelanggan untuk Admin



Gambar 5. 80 Hasil Implementasi Form Edit Pelanggan untuk Admin

Tabel 5. 31 Kode Backend showAllPelanggan pada Controller Pelanggan

```
Pelanggan.php
1 public function showAllPelanggan()
2 {
3     return $this->respond($this->model->findAll());
4 }
```

Tabel 5. 32 Kode Backend updatePelanggan pada Controller Pelanggan

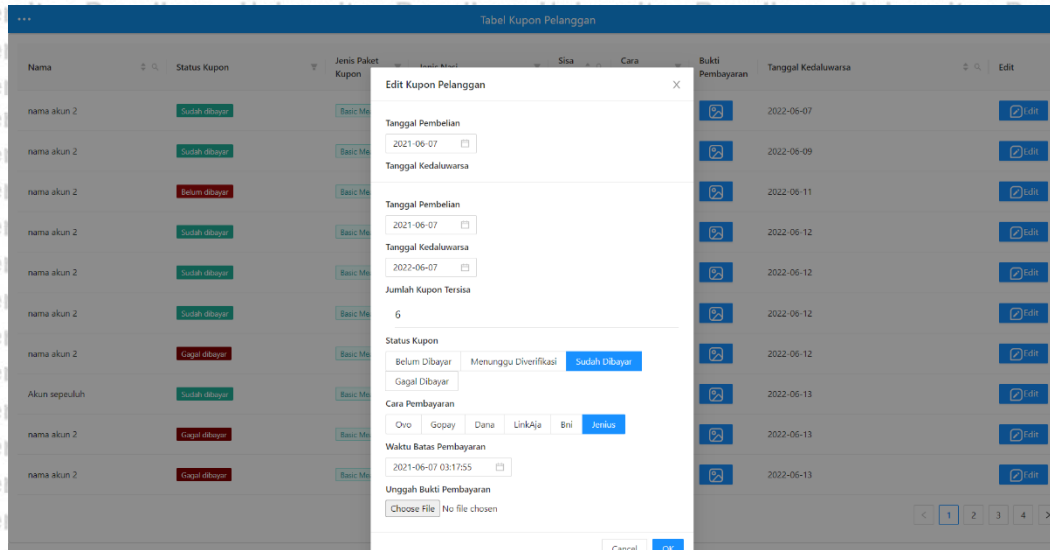
```
Pelanggan.php
1 public function updatePelanggan()
2 {
3     $data = $this->request->getPost();
4
5     $data['updated_by'] = 10;
6     $data['updated_date'] = date("Y-m-d H:i:s");
7
8     if ($this->model->save($data)) {
9         $dataPelangganFromDatabase = $this->model->find($data['id_pelanggan']);
10        return $this->respondCreated(['id_pelanggan' => $data['id_pelanggan'], 'status' => 'success', 'info' => 'update pelanggan', 'dataPelanggan' => $dataPelangganFromDatabase]);
11    }
12 }
```

5.2.2.19 Implementasi Mengedit Kupon Pelanggan

Mengedit kupon pelanggan hanya bisa dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin dapat melakukan perubahan data pada tabel kupon pelanggan yang berasal dari pembelian kupon oleh pelanggan. Halaman tabel kupon pelanggan Gambar 5. 81 berfungsi agar admin mendapatkan informasi kupon milik pelanggan yang telah dibeli. Jika admin ingin mengubah data kupon pelanggan, admin dapat menekan tombol “edit” sehingga sistem menampilkan form edit Gambar 5. 82. Dalam proses ini melibatkan fungsi `showAllKuponPelanggan()` Tabel 5. 33 dalam *controller* mix yang memanggil fungsi `getAllKuponPelangganPaketKuponPelanggan()` Tabel 5. 34 dari model mix, fungsi-fungsi tersebut bertujuan untuk menampilkan semua data kupon milik pelanggan. Sedangkan fungsi `adminUpdateKuponPelanggan()` Tabel 5. 35 dalam *controller* kupon pelanggan bertujuan untuk mengubah perubahan data sesuai dengan masukan admin.

Nama	Status Kupon	Jenis Paket Kupon	Jenis Nasi	Sisa Kupon	Cara Pembayaran	Bukti Pembayaran	Tanggal Kedaluwarsa	Edit
nama akun 2	Sudah dibayar	Basic Meal Box	Nasi putih	6	Jemas	[Bukti]	2022-06-07	[Edit]
nama akun 2	Sudah dibayar	Basic Meal Box	Nasi putih	12	Dana	[Bukti]	2022-06-09	[Edit]
nama akun 2	Belum dibayar	Basic Meal Box	Nasi putih	6	Jemas	[Bukti]	2022-06-11	[Edit]
nama akun 2	Sudah dibayar	Basic Meal Box	Nasi putih	12	Gopay	[Bukti]	2022-06-12	[Edit]
nama akun 2	Sudah dibayar	Basic Meal Box	Nasi putih	24	Dana	[Bukti]	2022-06-12	[Edit]
nama akun 2	Sudah dibayar	Basic Meal Box	Nasi putih	48	Gopay	[Bukti]	2022-06-12	[Edit]
nama akun 2	Gagal dibayar	Basic Meal Box	Nasi putih	48	Jemas	[Bukti]	2022-06-12	[Edit]
Akun sepuluh	Sudah dibayar	Basic Meal Box	Nasi putih	0	Gopay	[Bukti]	2022-06-13	[Edit]
nama akun 2	Gagal dibayar	Basic Meal Box	Nasi merah	6	Jemas	[Bukti]	2022-06-13	[Edit]
nama akun 2	Gagal dibayar	Basic Meal Box	Nasi putih	24	Bank BNI	[Bukti]	2022-06-13	[Edit]

Gambar 5. 81 Hasil Implementasi Tampilan Tabel Kupon Pelanggan



Gambar 5. 82 Hasil Implementasi Tampilan *Form* Edit Kupon Pelanggan

Tabel 5. 33 Kode *Backend* showAllKuponPelanggan pada *Controller* Mix

```
Mix.php
1 public function showAllKuponPelanggan ()
2 {
3     $mixModel = new MixModel ();
4     $data = $mixModel->
5     >getAllKuponPelangganPaketKuponPelanggan ();
6     return $this->respond ($data);
7 }
```

Tabel 5. 34 Kode *Backend* getAllKuponPelangganPaketKuponPelanggan pada *Model* Mix

```
MixModel.php
1 public function
2 getAllKuponPelangganPaketKuponPelanggan ()
3 {
4     return $this->db->table ('kupon_pelanggan')
5     ->join ('paket_kupon',
6     'paket_kupon.id_paket_kupon=kupon_pelanggan.id_paket_kupon')
7     ->join ('pelanggan',
8     'pelanggan.id_pelanggan=kupon_pelanggan.id_pelanggan')
9     ->get () ->getResultArray ();
10 }
```

Tabel 5. 35 Kode *Backend* adminUpdateKuponPelanggan pada *Controller* Kupon Pelanggan

```
KuponPelanggan.php
```

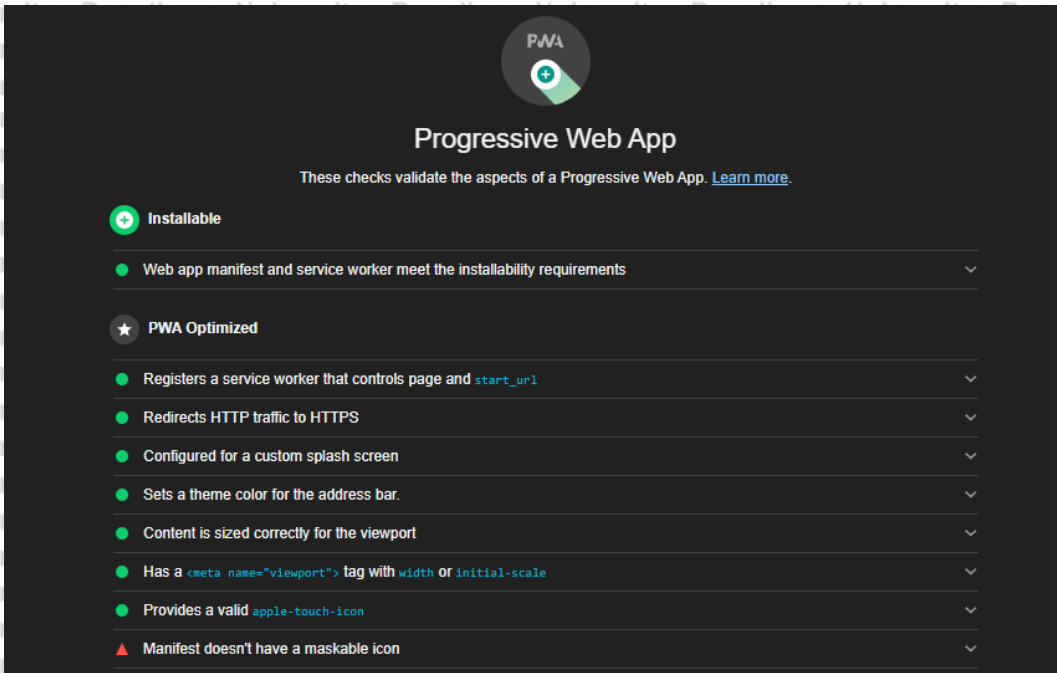
```

1 public function adminUpdateKuponPelanggan()
2 {
3     date_default_timezone_set('asia/jakarta');
4     helper(['form']);
5
6     $dataInput = $this->request->getPost();
7     $file = $this->request->getFile('bukti_pembayaran');
8     $newName = "BP".j.rand(0,999).date("YmdHis");
9     $newName = $newName . ".jpg";
10    $file->move('./assets/UploadBuktiPembayaran',
11    $newName);
12    $dataInput['bukti_pembayaran'] = $newName;
13    $dataInput['updated_by'] = 10;
14    $dataInput['updated_date'] = date("Y-m-d H:i:s");
15
16    if ($this->model->save($dataInput)) {
17        $dataMenuFromDatabase = $this->model-
18        >find($dataInput['id_kupon_pelanggan']);
19        return $this->respond(['id_kupon_pelanggan' =>
20        $dataInput['id_kupon_pelanggan'], 'status' => 'success',
21        'info' => 'update kupon pelanggan successfully',
22        'dataKuponPelanggan' => $dataMenuFromDatabase]);
23    }
24 }

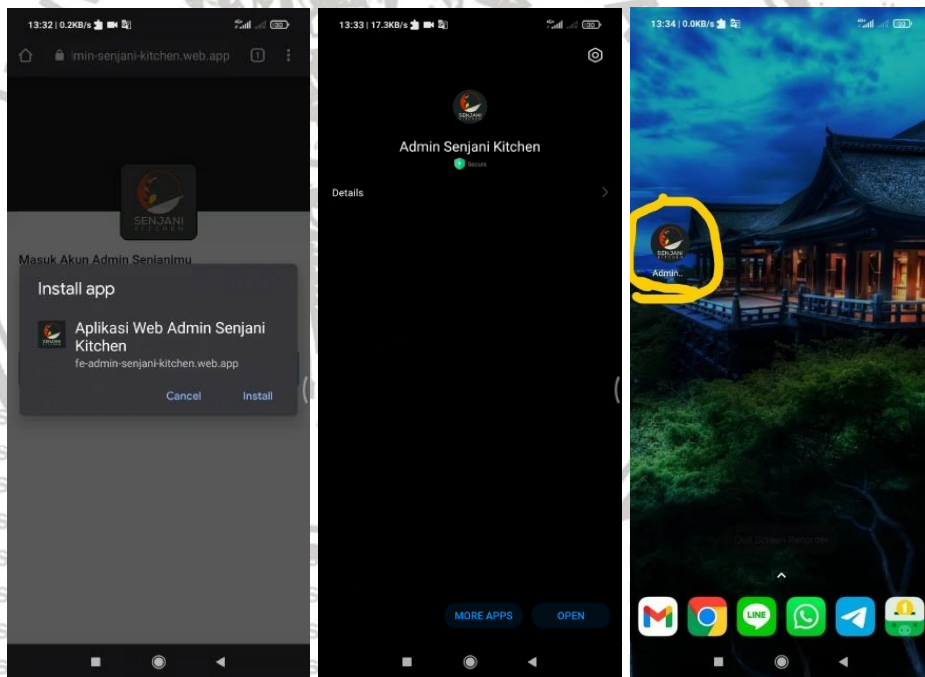
```

5.2.2.20 Implementasi Progressive Web App untuk Admin

PWA berfungsi agar admin mendapatkan opsi memasang aplikasi web layaknya aplikasi *native* di perangkatnya. *Service worker* adalah *script* yang berjalan di *background* terpisah dari halaman web dan bagian dari *progressive web app*. Tangkapan layar Gambar 5. 83 membuktikan bahwa PWA berhasil diimplementasikan dan tangkapan layar Gambar 5. 84 percobaan melakukan pemasangan aplikasi web pada perangkat Android. Dalam implementasi PWA ini melibatkan *file manifest*, *index*, *service-worker*, dan *serviceWorkerRegistration*. Manifest dalam Tabel 5. 36 berisi informasi nama web, ikon dan ukuran yang digunakan, warna *default*, dan *splash screen*. Index dalam Tabel 5. 37 berfungsi sebagai pemanggilan registrasi *service worker*. *Service worker* berisi *script* untuk manajemen *file* yang ingin dimasukkan ke dalam *cache*. *Service worker registration* berfungsi sebagai pengecekan apakah perangkat yang digunakan bisa menjalankan *service worker* atau tidak.



Gambar 5. 83 Bukti Implementasi PWA untuk Admin



Gambar 5. 84 Proses Pemasangan PWA untuk Admin

Tabel 5. 36 Kode *Frontend Manifest* untuk Admin

Manifest.json	
26	{
27	"short_name": "Admin Senjani Kitchen",

```

28 "name": "Aplikasi Web Admin Senjani Kitchen",
29 "icons": [
30 {
31   "src": "favicon.ico",
32   "sizes": "64x64 32x32 24x24 16x16",
33   "type": "image/x-icon"
34 },
35 {
36   "src": "logo192.png",
37   "type": "image/png",
38   "sizes": "192x192"
39 },
40 {
41   "src": "logo512.png",
42   "type": "image/png",
43   "sizes": "512x512"
44 }
45 ],
46 "start_url": ".",
47 "display": "standalone",
48 "theme_color": "#000000",
49 "background_color": "#ffffff"
50 }

```

Tabel 5. 37 Kode Frontend Index untuk Admin

```

Index.js
5 ...
6 import * as serviceWorkerRegistration from
  './serviceWorkerRegistration';
7
8 serviceWorkerRegistration.register();

```


BAB 6 PENGUJIAN

6.1 Pengujian Sistem

Tahap terakhir dalam metode *waterfall* adalah pengujian sistem. Pengujian dilakukan untuk memastikan semua yang telah dibuat dapat berjalan semestinya dan mencapai tujuan dengan performa yang baik. Terdapat beberapa jenis testing perangkat lunak yang dapat dilakukan, untuk penelitian ini penulis menggunakan dua pengujian, yaitu *validation testing* dan *system usability scale*.

6.1.1 Pengujian Validasi

Pengujian validasi atau *Validation testing* adalah pengujian untuk mencari tahu apakah sistem berjalan sesuai harapan atau tidak dengan melakukan beberapa kasus uji (Sommerville, 2011). Tiap fungsionalitas yang diuji akan disamakan alur dan keluarannya dengan kebutuhan. Pengujian dilakukan untuk semua alur baik itu alur utama maupun alur alternatif berdasarkan *use case* yang telah dibuat sebelumnya. Pengujian ini berfungsi untuk mengecek apakah sistem telah berjalan sesuai dengan hasil yang diharapkan atau belum. Ada 18 *use case* dan 31 skenario uji dalam pengujian ini.

6.1.1.1 Register untuk Pelanggan

Aktivitas *register* hanya bisa dilakukan oleh pelanggan saja. *Register* berfungsi untuk menambahkan data akun pelanggan baru. Pendaftaran akun pelanggan baru hanya bisa sukses bila *email* yang digunakan belum pernah terdaftar sebelumnya. Ada 2 pengujian, pengujian pertama Tabel 6. 1 dilakukan dengan *email* yang belum terdaftar sebelumnya, sedangkan pengujian kedua Tabel 6. 2 dilakukan dengan *email* yang telah terdaftar sebelumnya.

Tabel 6. 1 Validation Testing Register Email Belum Terdaftar

Kasus Uji	<i>Register</i>
Tujuan	Menambahkan akun pelanggan baru agar bisa masuk ke sistem
Prosedur	1. Pelanggan baru mengisi <i>email</i> , <i>password</i> , nama, alamat, nomor hp, dan catatan makanan 2. Pelanggan menekan tombol “daftar”
Harapan	Akun pelanggan baru terdaftar dan masuk dalam halaman beranda
Hasil	Akun pelanggan baru terdaftar dan masuk dalam halaman beranda
Status	Valid

Tabel 6. 2 Validation Testing Register Email Telah Terdaftar

Kasus Uji	<i>Register</i>
Tujuan	Menguji fungsionalitas <i>register</i> dengan memasukkan <i>email</i> yang telah terdaftar
Prosedur	1. Pelanggan baru mengisi <i>email</i> yang telah terdaftar , <i>password</i> , nama, alamat, nomor hp, dan catatan makanan 2. Pelanggan menekan tombol “daftar”
Harapan	Akun pelanggan baru gagal dan menampilkan notifikasi pendaftaran akun pelanggan baru gagal
Hasil	Akun pelanggan baru gagal dan menampilkan notifikasi “email telah terdaftar, tidak dapat membuat akun”
Status	Valid

6.1.1.2 Login untuk Pelanggan

Aktivitas *login* dapat dilakukan oleh pelanggan. *Login* berfungsi agar aktor dapat masuk kedalam sistem menggunakan *email* dan *password* yang telah terdaftar sebelumnya. Ada 2 pengujian pada aplikasi web sisi pelanggan, yaitu pengujian pertama Tabel 6. 3 dilakukan dengan *email* dan *password* yang telah terdaftar dan benar, sedangkan pengujian kedua Tabel 6. 4 menggunakan *email* dan *password* yang salah.

Tabel 6. 3 Validation Testing Pelanggan Berhasil

Kasus Uji	<i>Login</i>
Tujuan	Menguji fungsionalitas <i>login</i> dengan <i>email</i> dan <i>password</i> yang telah terdaftar
Prosedur	1. Pelanggan mengisi <i>email</i> dan <i>password</i> yang telah terdaftar 2. Pelanggan menekan tombol “masuk”
Harapan	Akun pelanggan berhasil dikenali dan masuk ke dalam halaman beranda
Hasil	Akun pelanggan berhasil dikenali dan masuk ke dalam halaman beranda
Status	Valid

Tabel 6. 4 Validation Testing Pelanggan Gagal

Kasus Uji	<i>Login</i>
-----------	--------------

Tujuan	Menguji fungsionalitas <i>login</i> dengan <i>email</i> dan <i>password</i> yang salah/ belum terdaftar
Prosedur	1. Pelanggan mengisi <i>email</i> dan <i>password</i> yang belum terdaftar 2. Pelanggan menekan tombol “masuk”
Harapan	Proses <i>login</i> gagal dan menampilkan notifikasi <i>email</i> dan <i>password</i> salah
Hasil	Proses <i>login</i> gagal dan menampilkan notifikasi “Email tidak ditemukan” atau “password salah”
Status	Valid

6.1.1.3 Logout untuk Pelanggan

Aktivitas *logout* dapat dilakukan oleh pelanggan. *Logout* berfungsi agar aktor dapat keluar sistem. Syarat pengujian yaitu admin telah masuk ke dalam sistem, pengujian dijelaskan dalam Tabel 6. 5.

Tabel 6. 5 Validation Testing Logout Pelanggan

Kasus Uji	<i>Logout</i>
Tujuan	Menguji fungsionalitas <i>logout</i> sehingga pelanggan dapat keluar dari sistem
Prosedur	1. Pelanggan menekan tombol “keluar”
Harapan	Akun pelanggan berhasil keluar dari sistem dan masuk ke halaman login
Hasil	Akun pelanggan berhasil keluar dari sistem dan masuk ke halaman login
Status	Valid

6.1.1.4 Melihat Jadwal Menu untuk Pelanggan

Aktivitas melihat jadwal menu hanya dapat dilakukan oleh pelanggan saja. Melihat jadwal menu berfungsi agar pelanggan mendapatkan informasi spesifikasi menu, yaitu jadwal dan waktu menu dikirimkan, nama menu, dan keterangan menu. Pengujian dilakukan dengan syarat pelanggan telah masuk ke dalam sistem, pengujian dijelaskan dalam Tabel 6. 6.

Tabel 6. 6 Validation Testing Melihat Jadwal Menu

Kasus Uji	Melihat Jadwal Menu
Tujuan	Menguji fungsionalitas lihat jadwal menu menampilkan jadwal menu yang telah tersimpan

Prosedur	1. Pelanggan menekan tombol “Lihat Jadwal Menu” 2. Pelanggan memilih jenis paket kupon
Harapan	Sistem menampilkan jadwal menu berdasarkan jenis paket kupon
Hasil	Sistem menampilkan jadwal menu berdasarkan jenis paket kupon
Status	Valid

6.1.1.5 Membeli Kupon untuk Pelanggan

Aktivitas membeli kupon hanya bisa dilakukan oleh pelanggan saja. Membeli kupon berfungsi agar kupon tersebut dapat ditukarkan dengan menu sesuai dengan jadwalnya. Ada 2 pengujian, pengujian dilakukan saat pelanggan telah membayar kupon via transfer dan sedang ingin mengunggah bukti pembayaran. Pengujian pertama pelanggan Tabel 6. 7 mengunggah file yang diinginkan yaitu file gambar sedangkan pengujian kedua Tabel 6. 8 pelanggan mengunggah file bukan berbentuk gambar.

Tabel 6. 7 Validation Testing Membeli Kupon Unggah Gambar

Kasus Uji	Membeli Kupon
Tujuan	Menguji fungsionalitas beli kupon dengan unggahan bukti pembayaran berjenis gambar
Prosedur	1. Pelanggan menekan tombol “Beli Kupon” 2. Pelanggan memilih salah satu paket kupon yang tersedia 3. Pelanggan memilih salah satu metode pembayaran yang tersedia 4. Pelanggan mengunggah <i>file</i> bukti pembayaran berbentuk gambar 5. Pelanggan menekan tombol “submit”
Harapan	Pembelian paket kupon berhasil dan pelanggan masuk ke beranda
Hasil	Pembelian paket kupon berhasil dan pelanggan masuk ke beranda
Status	Valid

Tabel 6. 8 Validation Testing Membeli Kupon Unggah Bukan Gambar

Kasus Uji	Membeli Kupon
Tujuan	Menguji fungsionalitas beli kupon dengan unggahan bukti pembayaran bukan berjenis gambar

Prosedur	<ol style="list-style-type: none"> 1. Pelanggan menekan tombol “Beli Kupon” 2. Pelanggan memilih salah satu paket kupon yang tersedia 3. Pelanggan memilih salah satu metode pembayaran yang tersedia 4. Pelanggan mengunggah <i>file</i> bukti pembayaran berjenis bukan gambar (dokumen / docx) 5. Pelanggan menekan tombol “submit”
Harapan	Pembelian paket kupon gagal dan menampilkan notifikasi unggahan harus berjenis gambar
Hasil	Pembelian paket kupon gagal dan menampilkan notifikasi “Unggah Bukti Gagal, Bukti Pembayaran Harus Berbentuk Gambar (jpg, png, dll)”
Status	Valid

6.1.1.6 Menggunakan Kupon untuk Pelanggan

Aktivitas menggunakan kupon hanya bisa dilakukan oleh pelanggan saja. Menggunakan kupon berfungsi agar pelanggan dapat menukarkan kupon yang telah dibeli sebelumnya dengan menu makanan. Ada 2 pengujian, pengujian dilakukan saat pelanggan memasukkan *form* biodata diri dan catatan pengantaran untuk mempermudah pengiriman. Pengujian pertama Tabel 6. 9 dilakukan pelanggan setelah pelanggan mengisi semua *form* yang ada, sedangkan pengujian kedua Tabel 6. 10 pelanggan tidak mengisi semua *form*.

Tabel 6. 9 Validation Testing Menggunakan Kupon Berhasil

Kasus Uji	Menggunakan Kupon
Tujuan	Menguji fungsionalitas gunakan kupon dengan mengisi semua <i>form</i> pengantaran
Prosedur	<ol style="list-style-type: none"> 1. Pelanggan menekan tab menu bawah “kupon” 2. Pelanggan menekan tombol “gunakan kupon” 3. Pelanggan memilih jadwal menu yang tersedia 4. Pelanggan mengisi semua <i>form</i> pengantaran 5. Pelanggan menekan “pesan”
Harapan	Pemesanan / penukaran kupon berhasil dan menampilkan notifikasi penggunaan kupon berhasil
Hasil	Pemesanan / penukaran kupon berhasil dan menampilkan notifikasi “Pemesanan Berhasil”
Status	Valid

Tabel 6. 10 Validation Testing Menggunakan Kupon Gagal

Kasus Uji	Menggunakan Kupon
Tujuan	Menguji fungsionalitas gunakan kupon tanpa mengisi semua form pengantaran
Prosedur	<ol style="list-style-type: none"> 1. Pelanggan menekan tab menu bawah “kupon” 2. Pelanggan menekan tombol “gunakan kupon” 3. Pelanggan memilih jadwal menu yang tersedia 4. Pelanggan mengisi sebagian form pengantaran dan mengkosongkan sebagian 5. Pelanggan menekan “pesan”
Harapan	Pemesanan / penukaran kupon gagal dan menampilkan notifikasi penggunaan kupon gagal
Hasil	Pemesanan / penukaran kupon gagal dan menampilkan notifikasi “Semua form harus terisi”
Status	Valid

6.1.1.7 Melihat Riwayat Kupon untuk Pelanggan

Aktivitas melihat riwayat kupon hanya dapat dilakukan oleh pelanggan saja. Melihat riwayat kupon milik pelanggan itu sendiri berfungsi agar pelanggan mendapatkan informasi kupon yang dimilikinya beserta spesifikasinya (jumlah kupon tersisa, jenis paket kupon, tanggal pembelian, tanggal kedaluwarsa, dan status kupon). Pengujian dijelaskan dalam Tabel 6. 11.

Tabel 6. 11 Validation Testing Melihat Riwayat Kupon

Kasus Uji	Melihat Riwayat Kupon
Tujuan	Menguji fungsionalitas lihat riwayat kupon menampilkan semua kupon milik pelanggan
Prosedur	1. Pelanggan menekan tab menu bawah “Kupon”
Harapan	Pelanggan mendapatkan informasi kupon baik yang masih bisa digunakan maupun sudah tidak bisa digunakan
Hasil	Pelanggan mendapatkan informasi kupon baik yang masih bisa digunakan maupun sudah tidak bisa digunakan
Status	Valid

6.1.1.8 Melihat Riwayat Pesanan untuk Pelanggan

Aktivitas melihat riwayat pesanan hanya dapat dilakukan oleh pelanggan saja. Melihat riwayat pesanan milik pelanggan itu sendiri berfungsi agar pelanggan mendapatkan informasi pesanan yang akan terjadi ataupun yang telah terjadi informasi yang didapatkan pelanggan, yaitu tanggal & waktu pengiriman, nama menu, keterangan menu, dan status pesanan. Pengujian dijelaskan dalam Tabel 6. 12.

Tabel 6. 12 Validation Testing Melihat Riwayat Pesanan

Kasus Uji	Melihat Riwayat Pesanan
Tujuan	Menguji fungsionalitas lihat riwayat pesanan menampilkan semua pesanan yang akan datang maupun yang telah selesai
Prosedur	1. Pelanggan menekan tab menu bawah “Pesanan”
Harapan	Pelanggan mendapatkan informasi pesanan yang akan datang maupun yang telah selesai
Hasil	Pelanggan mendapatkan informasi pesanan yang akan datang maupun yang telah selesai
Status	Valid

6.1.1.9 Progressive Web App untuk Pelanggan

PWA berfungsi agar pelanggan mendapatkan opsi memasang aplikasi web layaknya aplikasi *native* di perangkatnya. Aktivitas pemasangan aplikasi web agar pelanggan mendapatkan opsi kenyamanan dan kecepatan dalam melakukan akses ke aplikasi web. Data web tersimpan dalam *cache* perangkat sehingga perangkat tidak perlu mengunduh semua data dari server. Pengujian dilakukan untuk melihat apakah sistem berjalan sesuai dengan yang diharapkan atau tidak. Pengujian dijelaskan dalam Tabel 6. 13.

Tabel 6. 13 Validation Testing PWA untuk Pelanggan

Kasus Uji	Pemasangan aplikasi web dalam perangkat
Tujuan	Menguji kemampuan sistem untuk dipasang dalam perangkat pelanggan
Prosedur	1. Pelanggan menekan tombol <i>install</i> pada browser 2. Pelanggan menunggu proses instalasi dan menekan tombol “open”
Harapan	Sistem terpasang pada perangkat pelanggan
Hasil	Sistem terpasang pada perangkat pelanggan
Status	Valid

6.1.1.10 Login untuk Admin

Aktivitas *login* dapat dilakukan oleh admin. *Login* berfungsi agar aktor dapat masuk kedalam sistem menggunakan *email* dan *password* yang telah terdaftar sebelumnya. Ada 2 pengujian pada aplikasi web sisi admin, yaitu pengujian pertama Tabel 6. 14 dilakukan dengan *email* dan *password* yang telah terdaftar dan benar, sedangkan pengujian kedua Tabel 6. 15 menggunakan *email* dan *password* yang salah.

Tabel 6. 14 Validation Testing Login Admin Berhasil

Kasus Uji	<i>Login</i>
Tujuan	Menguji fungsionalitas <i>login</i> dengan <i>email</i> dan <i>password</i> yang telah terdaftar
Prosedur	1. Admin mengisi <i>email</i> dan <i>password</i> yang telah terdaftar 2. Admin menekan tombol “masuk”
Harapan	Akun admin berhasil dikenali dan masuk ke dalam halaman beranda
Hasil	Akun admin berhasil dikenali dan masuk ke dalam halaman beranda
Status	Valid

Tabel 6. 15 Validation Testing Login Admin Gagal

Kasus Uji	<i>Login</i>
Tujuan	Menguji fungsionalitas <i>login</i> dengan <i>email</i> dan <i>password</i> yang salah/ belum terdaftar
Prosedur	1. Admin mengisi <i>email</i> dan <i>password</i> yang belum terdaftar 2. Admin menekan tombol “masuk”
Harapan	Proses <i>login</i> gagal dan menampilkan notifikasi <i>email</i> dan <i>password</i> salah
Hasil	Proses <i>login</i> gagal dan menampilkan notifikasi “Email tidak ditemukan” atau “password salah”
Status	Valid

6.1.1.11 Logout untuk Admin

Aktivitas *logout* dapat dilakukan oleh admin. *Logout* berfungsi agar aktor dapat keluar sistem. Syarat pengujian yaitu admin telah masuk ke dalam sistem, pengujian dijelaskan dalam Tabel 6. 16.

Tabel 6. 16 Validation Testing Logout Admin

Kasus Uji	<i>Logout</i>
Tujuan	Menguji fungsionalitas <i>logout</i> sehingga admin dapat keluar dari sistem
Prosedur	1. Admin menekan tombol “keluar”
Harapan	Akun admin berhasil keluar dari sistem dan masuk ke halaman login
Hasil	Akun admin berhasil keluar dari sistem dan masuk ke halaman login
Status	Valid

6.1.1.12 Verifikasi Bukti Pembayaran untuk Admin

Aktivitas verifikasi bukti pembayaran hanya bisa dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin bisa menganalisis keaslian dari foto bukti pembayaran yang diunggah oleh pelanggan. Ada 2 pengujian, pengujian dilakukan saat admin telah selesai menganalisis foto bukti pembayaran. Pengujian pertama Tabel 6. 17 admin menentukan bahwa foto bukti pembayaran tersebut asli, sedangkan pengujian kedua Tabel 6. 18 admin menentukan bahwa foto bukti pembayaran tersebut tidak asli atau tidak cocok.

Tabel 6. 17 Validation Testing Verifikasi Bukti Pembayaran Valid

Kasus Uji	Verifikasi Bukti Pembayaran
Tujuan	Menguji fungsionalitas verifikasi bukti pembayaran dengan bukti pembayaran yang asli / valid
Prosedur	<ol style="list-style-type: none"> 1. Admin menekan tombol “Verifikasi Bukti Pembayaran” pada menu 2. Admin menekan tombol “periksa” 3. Admin menganalisis foto bukti pembayaran dan memilih “Valid/ Asli/ Cocok” pada <i>form</i> 4. Admin menekan tombol “Ok”

Harapan	Sistem mengubah status kupon pelanggan dari “menunggu diverifikasi” menjadi “sudah dibayar” dan pelanggan mendapatkan notifikasi verifikasi berhasil
Hasil	Sistem mengubah status kupon pelanggan dari “menunggu diverifikasi” menjadi “sudah dibayar” dan pelanggan mendapatkan notifikasi verifikasi berhasil
Status	Valid

Tabel 6. 18 Validation Testing Verifikasi Bukti Pembayaran Tidak Valid

Kasus Uji	Verifikasi Bukti Pembayaran
Tujuan	Menguji fungsionalitas verifikasi bukti pembayaran dengan bukti pembayaran yang tidak valid/ beda/ palsu
Prosedur	<ol style="list-style-type: none"> 1. Admin menekan tombol “Verifikasi Bukti Pembayaran” pada menu 2. Admin menekan tombol “periksa” 3. Admin menganalisis foto bukti pembayaran dan memilih “Tidak Valid/ Beda” pada <i>form</i> 4. Admin menekan tombol “Ok”
Harapan	Sistem mengubah status kupon pelanggan dari “menunggu diverifikasi” menjadi “gagal dibayar” dan pelanggan mendapatkan notifikasi verifikasi berhasil
Hasil	Sistem mengubah status kupon pelanggan dari “menunggu diverifikasi” menjadi “gagal dibayar” dan pelanggan mendapatkan notifikasi verifikasi berhasil
Status	Valid

6.1.1.13 Membuat Jadwal Menu Baru untuk Admin

Aktivitas membuat jadwal menu baru hanya dapat dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin dapat menambahkan menu baru beserta spesifikasinya yaitu tanggal & waktu menu dikirimkan, nama menu, keterangan, dan foto menu. Pengujian dilakukan untuk melihat apakah sistem berjalan sesuai yang diharapkan atau tidak. Pengujian dijelaskan dalam Tabel 6. 19.

Tabel 6. 19 Validation Testing Membuat Jadwal Menu Baru

Kasus Uji	Membuat Jadwal Menu Baru
Tujuan	Menguji fungsionalitas membuat jadwal menu baru untuk menambahkan data menu baru

Prosedur	<ol style="list-style-type: none"> 1. Admin menekan menu “Tabel Jadwal Menu” pada <i>sidebar menu</i> 2. Admin menekan tombol “Tambah Menu” 3. Admin mengisi <i>form</i> data menu (tanggal menu, waktu, nama makanan, keterangan, lauk tambahan, dan foto makanan) 4. Admin menekan tombol “Ok”
Harapan	Sistem menambahkan data menu baru dan menampilkan menu baru berhasil ditambahkan
Hasil	Sistem menambahkan data menu baru dan menampilkan menu baru berhasil ditambahkan
Status	Valid

6.1.1.14 Mengedit Jadwal Menu untuk Admin

Aktivitas mengedit jadwal menu hanya bisa dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin dapat melakukan perubahan data pada jadwal menu yang telah tersimpan. Ada 2 pengujian, pengujian dilakukan saat admin telah mengubah data yang ingin diubah dalam *form*. Pengujian pertama Tabel 6. 20 admin menekan tombol “ok” untuk menyimpan perubahan data, sedangkan pengujian kedua Tabel 6. 21 menekan tombol “cancel” untuk membatalkan.

Tabel 6. 20 Validation Testing Mengedit Jadwal Menu Berhasil

Kasus Uji	Mengedit Jadwal Menu
Tujuan	Menguji fungsionalitas mengedit jadwal menu dengan mengubah data yang telah ada
Prosedur	<ol style="list-style-type: none"> 1. Admin menekan menu “Tabel Jadwal Menu” pada <i>sidebar menu</i> 2. Admin menekan tombol “Edit” 3. Admin mengubah data yang ingin diedit 4. Admin menekan tombol “Ok”
Harapan	Sistem menyimpan perubahan data yang telah diedit dan menampilkan notifikasi bahwa data berhasil diubah
Hasil	Sistem menyimpan perubahan data yang telah diedit dan menampilkan notifikasi bahwa data berhasil diubah
Status	Valid

Tabel 6. 21 Validation Testing Mengedit Jadwal Menu Gagal

Kasus Uji	Mengedit Jadwal Menu
Tujuan	Menguji fungsionalitas mengedit jadwal menu dengan membatalkan perubahan
Prosedur	<ol style="list-style-type: none"> 1. Admin menekan menu "Tabel Jadwal Menu" pada <i>sidebar menu</i> 2. Admin menekan tombol "Edit" 3. Admin mengubah data yang ingin diedit 4. Admin menekan tombol "Cancel"
Harapan	Sistem membatalkan perubahan data dan admin masuk kembali ke tampilan tabel
Hasil	Sistem membatalkan perubahan data dan admin masuk kembali ke tampilan tabel
Status	Valid

6.1.1.15 Mengedit Paket Kupon untuk Admin

Aktivitas mengedit paket kupon hanya bisa dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin dapat melakukan perubahan data pada tabel paket kupon. Ada 2 pengujian, pengujian dilakukan saat admin telah mengubah data yang ingin diubah dalam *form*. Pengujian pertama Tabel 6. 22 admin menekan tombol "ok" untuk menyimpan perubahan data, sedangkan pengujian kedua Tabel 6. 23 menekan tombol "cancel" untuk membatalkan.

Tabel 6. 22 Validation Testing Mengedit Paket Kupon Berhasil

Kasus Uji	Mengedit Paket Kupon
Tujuan	Menguji fungsionalitas mengedit data paket kupon dengan mengubah data yang telah ada
Prosedur	<ol style="list-style-type: none"> 1. Admin menekan menu "Tabel Paket Kupon" pada <i>sidebar menu</i> 2. Admin menekan tombol "Edit" 3. Admin mengubah data yang ingin diedit 4. Admin menekan tombol "Ok"
Harapan	Sistem menyimpan perubahan data yang telah diedit dan menampilkan notifikasi bahwa data berhasil diubah
Hasil	Sistem menyimpan perubahan data yang telah diedit dan menampilkan notifikasi bahwa data berhasil diubah

Status	Valid
--------	-------

Tabel 6. 23 Validation Testing Mengedit Paket Kupon Gagal

Kasus Uji	Mengedit Paket Kupon
Tujuan	Menguji fungsionalitas membatalkan pengeditan data paket kupon
Prosedur	<ol style="list-style-type: none"> 1. Admin menekan menu “Tabel Paket Kupon” pada <i>sidebar menu</i> 2. Admin menekan tombol “Edit” 3. Admin mengubah data yang ingin diedit 4. Admin menekan tombol “Cancel”
Harapan	Sistem membatalkan perubahan data dan admin masuk kembali ke tampilan tabel
Hasil	Sistem membatalkan perubahan data dan admin masuk kembali ke tampilan tabel
Status	Valid

6.1.1.16 Melihat Hasil Pesanan untuk Admin

Aktivitas melihat hasil pesanan hanya dapat dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin mendapatkan informasi penukaran kupon dengan menu yang dilakukan oleh pelanggan. Admin mendapatkan informasi kupon, yaitu nama pelanggan, status kupon, jenis paket kupon, tanggal & waktu pengiriman, biodata penerima, catatan makanan, catatan pengiriman, dan status pesanan. Pengujian dilakukan untuk melihat apakah sistem berjalan sesuai yang diharapkan atau tidak. Pengujian dijelaskan dalam Tabel 6. 24.

Tabel 6. 24 Validation Testing Melihat Hasil Pesanan

Kasus Uji	Melihat Hasil Pesanan
Tujuan	Menguji fungsionalitas lihat hasil pesanan yang telah dipesan oleh pelanggan sebelumnya
Prosedur	<ol style="list-style-type: none"> 1. Admin menekan menu “Tabel Pesanan” pada <i>sidebar menu</i> 2. Admin melakukan <i>filter & sorting</i> sesuai dengan kebutuhan untuk mencari data yang diinginkan
Harapan	Admin mendapatkan informasi data pesanan yang telah dipesan oleh pelanggan sebelumnya

Hasil	Admin mendapatkan informasi data pesanan yang telah dipesan oleh pelanggan sebelumnya
Status	Valid

6.1.1.17 Mengedit Pesanan untuk Admin

Aktivitas mengedit pesanan hanya bisa dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin dapat melakukan perubahan data pada tabel pesanan yang telah diinput oleh pelanggan dengan cara penukaran kupon dengan menu. Ada 2 pengujian, pengujian dilakukan saat admin telah mengubah data yang ingin diubah dalam *form*. Pengujian pertama Tabel 6. 25 admin menekan tombol “ok” untuk menyimpan perubahan data, sedangkan pengujian kedua Tabel 6. 26 menekan tombol “cancel” untuk membatalkan.

Tabel 6. 25 Validation Testing Mengedit Pesanan Berhasil

Kasus Uji	Mengedit Pesanan
Tujuan	Menguji fungsionalitas mengedit data pesanan yang telah ada sebelumnya
Prosedur	1. Admin menekan menu “Tabel Pesanan” pada <i>sidebar menu</i> 2. Admin menekan tombol “Edit” 3. Admin mengubah data yang ingin diedit 4. Admin menekan tombol “Ok”
Harapan	Sistem menyimpan perubahan data yang telah diedit dan menampilkan notifikasi bahwa data berhasil diubah
Hasil	Sistem menyimpan perubahan data yang telah diedit dan menampilkan notifikasi bahwa data berhasil diubah
Status	Valid

Tabel 6. 26 Validation Testing Mengedit Pesanan

Kasus Uji	Mengedit Pesanan
Tujuan	Menguji fungsionalitas membatalkan pengeditan data pesanan
Prosedur	1. Admin menekan menu “Tabel Pesanan” pada <i>sidebar menu</i> 2. Admin menekan tombol “Edit” 3. Admin mengubah data yang ingin diedit 4. Admin menekan tombol “Cancel”
Harapan	Sistem membatalkan perubahan data dan admin masuk kembali ke tampilan tabel

Hasil	Sistem membatalkan perubahan data dan admin masuk kembali ke tampilan tabel
Status	Valid

6.1.1.18 Mengedit Pelanggan untuk Admin

Aktivitas mengedit pelanggan hanya bisa dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin dapat melakukan perubahan data pada tabel pelanggan yang diinput oleh pelanggan saat pendaftaran. Ada 2 pengujian, pengujian dilakukan saat admin telah mengubah data yang ingin diubah dalam form. Pengujian pertama Tabel 6. 27 admin menekan tombol “ok” untuk menyimpan perubahan data, sedangkan pengujian kedua Tabel 6. 28 menekan tombol “cancel” untuk membatalkan.

Tabel 6. 27 Validation Testing Mengedit Pelanggan Berhasil

Kasus Uji	Mengedit Pelanggan
Tujuan	Menguji fungsionalitas mengedit data pelanggan yang telah ada sebelumnya
Prosedur	1. Admin menekan menu “Tabel Pelanggan” pada <i>sidebar menu</i> 2. Admin menekan tombol “Edit” 3. Admin mengubah data yang ingin diedit 4. Admin menekan tombol “Ok”
Harapan	Sistem menyimpan perubahan data yang telah diedit dan menampilkan notifikasi bahwa data berhasil diubah
Hasil	Sistem menyimpan perubahan data yang telah diedit dan menampilkan notifikasi bahwa data berhasil diubah
Status	Valid

Tabel 6. 28 Validation Testing Mengedit Pelanggan Gagal

Kasus Uji	Mengedit Pelanggan
Tujuan	Menguji fungsionalitas membatalkan pengeditan data pelanggan
Prosedur	1. Admin menekan menu “Tabel Pelanggan” pada <i>sidebar menu</i> 2. Admin menekan tombol “Edit” 3. Admin mengubah data yang ingin diedit 4. Admin menekan tombol “Cancel”
Harapan	Sistem membatalkan perubahan data dan admin masuk kembali ke tampilan tabel

Hasil	Sistem membatalkan perubahan data dan admin masuk kembali ke tampilan tabel
Status	Valid

6.1.1.19 Mengedit Kupon Pelanggan untuk Admin

Aktivitas mengedit kupon pelanggan hanya bisa dilakukan oleh admin saja. Aktivitas ini berfungsi agar admin dapat melakukan perubahan data pada tabel kupon pelanggan yang berasal dari pembelian kupon oleh pelanggan. Ada 2 pengujian, pengujian dilakukan saat admin telah mengubah data yang ingin diubah dalam *form*. Pengujian pertama Tabel 6. 29 admin menekan tombol “ok” untuk menyimpan perubahan data, sedangkan pengujian kedua Tabel 6. 30 menekan tombol “cancel” untuk membatalkan.

Tabel 6. 29 Validation Testing Mengedit Kupon Pelanggan

Kasus Uji	Mengedit Kupon Pelanggan
Tujuan	Menguji fungsionalitas mengedit data kupon pelanggan yang telah ada sebelumnya
Prosedur	<ol style="list-style-type: none"> 1. Admin menekan menu “Tabel Kupon Pelanggan” pada <i>sidebar menu</i> 2. Admin menekan tombol “Edit” 3. Admin mengubah data yang ingin diedit 4. Admin menekan tombol “Ok”
Harapan	Sistem menyimpan perubahan data yang telah diedit dan menampilkan notifikasi bahwa data berhasil diubah
Hasil	Sistem menyimpan perubahan data yang telah diedit dan menampilkan notifikasi bahwa data berhasil diubah
Status	Valid

Tabel 6. 30 Validation Testing Mengedit Kupon Pelanggan

Kasus Uji	Mengedit Kupon Pelanggan
Tujuan	Menguji fungsionalitas membatalkan pengeditan data kupon milik pelanggan
Prosedur	<ol style="list-style-type: none"> 1. Admin menekan menu “Tabel Kupon Pelanggan” pada <i>sidebar menu</i> 2. Admin menekan tombol “Edit” 3. Admin mengubah data yang ingin diedit

	4. Admin menekan tombol “Cancel”
Harapan	Sistem membatalkan perubahan data dan admin masuk kembali ke tampilan tabel
Hasil	Sistem membatalkan perubahan data dan admin masuk kembali ke tampilan tabel
Status	Valid

6.1.1.20 Progressive Web App untuk Admin

PWA berfungsi agar admin mendapatkan opsi memasang aplikasi web layaknya aplikasi *native* di perangkatnya. Aktivitas pemasangan aplikasi web agar admin mendapatkan opsi kenyamanan dan kecepatan dalam melakukan akses ke aplikasi web, karena data web tersimpan dalam *cache* perangkat sehingga perangkat tidak perlu mengunduh semua data dari server. Pengujian dilakukan untuk melihat apakah sistem berjalan sesuai dengan yang diharapkan atau tidak. Pengujian dijelaskan dalam Tabel 6. 31.

Tabel 6. 31 Validation Testing PWA untuk Admin

Kasus Uji	Pemasangan aplikasi web dalam perangkat
Tujuan	Menguji kemampuan sistem untuk dipasang dalam perangkat admin
Prosedur	1. Admin menekan tombol <i>install</i> pada browser 2. Admin menunggu proses instalasi dan menekan tombol “open”
Harapan	Sistem terpasang pada perangkat admin
Hasil	Sistem terpasang pada perangkat admin
Status	Valid

6.1.2 Pengujian Usabilitas

6.1.2.1 Prosedur Pengujian Usabilitas

System Usability Scale (SUS) adalah alat yang handal untuk mengukur tingkat *usability* (Usability.gov, 2021). *System Usability* merupakan sebuah cara yang digunakan peneliti untuk melakukan evaluasi produk atau layanan perangkat keras, perangkat lunak, perangkat seluler, situs web, serta aplikasi. Pada pengujian usabilitas ini terdapat 2 jenis responden yaitu calon pelanggan dan calon admin. Responden untuk calon pelanggan adalah masyarakat, sedangkan responden admin adalah admin dan pemilik *senjani kitchen*. Mula-mula responden dijelaskan alur dari penggunaan aplikasi web ini, lalu responden disuruh untuk mencoba menggunakan aplikasi web sesuai dengan alur, selanjutnya responden menjawab 10 pertanyaan *system usability scale* Tabel 6. 33.

Tabel 6. 32 Jawaban Pertanyaan SUS

Pilihan Jawaban	Nilai	Keterangan
STS	1	Sangat Tidak Setuju
TS	2	Tidak Setuju
RG	3	Ragu-ragu
ST	4	Setuju
SS	5	Sangat Setuju

Tabel 6. 33 Pertanyaan System Usability Scale

No	Pertanyaan	Jawaban				
		STS	TS	RG	ST	SS
1	Saya pikir saya akan sering menggunakan sistem ini.					
2	Saya merasa sistem ini terlalu rumit padahal dapat dibuat lebih sederhana.					
3	Saya rasa sistem ini mudah untuk digunakan.					
4	Saya pikir saya membutuhkan bantuan dari orang teknis untuk dapat menggunakan sistem ini.					
5	Saya menemukan bahwa terdapat berbagai macam fitur yang terintegrasi dengan baik dalam sistem.					
6	Saya rasa banyak hal yang tidak konsisten terdapat pada sistem ini.					
7	Saya rasa mayoritas pengguna akan dapat mempelajari sistem ini dengan cepat.					
8	Saya menemukan bahwa sistem ini sangat tidak praktis ketika digunakan.					
9	Saya sangat yakin dapat menggunakan sistem ini.					
10	Saya harus belajar banyak hal terlebih dahulu sebelum saya dapat menggunakan sistem ini.					

6.1.2.2 Hasil dan Analisis Data Pengujian Usabilitas

Sesudah prosedur pengujian usabilitas telah selesai dilakukan oleh responden, proses selanjutnya adalah menghitung data tersebut, nilai didapat dari pertanyaan yang dipilih oleh responden dengan rentang 1 sampai 5. Tabel 6. 34 menjelaskan data hasil jawaban responden pelanggan yang ingin menggunakan jasa catering harian Senjani Kitchen, sedangkan Tabel 6. 35 menjelaskan data hasil jawaban responden admin yaitu admin dan pemilik Senjani Kitchen.

Tabel 6. 34 Hasil Jawaban Responden Pelanggan

Nama	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Responden 1	4	1	4	1	5	3	2	4	5	4
Responden 2	4	1	5	1	4	2	4	2	5	1
Responden 3	3	2	4	2	4	2	3	2	4	3
Responden 4	5	1	5	1	5	1	5	1	5	1
Responden 5	5	1	5	1	5	1	4	1	5	1
Responden 6	4	2	5	3	3	2	3	1	5	2
Responden 7	4	1	5	1	4	1	5	1	5	1
Responden 8	4	1	4	1	4	1	4	2	5	1
Responden 9	4	2	4	1	4	1	4	2	5	5
Responden 10	4	3	4	2	4	2	4	2	5	2

Tabel 6. 35 Hasil Jawaban Responden Admin

Nama	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Responden 1	5	4	3	1	5	1	3	3	5	1
Responden 2	5	4	3	5	5	2	4	2	5	2

Dari hasil yang telah didapatkan dari responden pelanggan dan admin, dilakukan perhitungan dengan mengikuti aturan SUS, yaitu setiap pernyataan dengan nomor ganjil maka skala jawaban responden dikurangi 1. Setiap pernyataan dengan nomor genap maka 5 dikurangi dengan skala jawaban responden. Kemudian skala jawaban responden dijumlah dan dikalikan dengan 2.5. Berikut merupakan perhitungan responden pelanggan Tabel 6. 36 dan responden admin Tabel 6. 37:

Tabel 6. 36 Hasil Perhitungan Responden Pelanggan

Nama	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Jumlah	Skor
Responden 1	3	4	3	4	4	2	1	1	4	1	27	67.5

Responden 2	3	4	4	4	3	3	3	3	4	4	35	87.5
Responden 3	2	3	3	3	3	3	2	3	3	2	27	67.5
Responden 4	4	4	4	4	4	4	4	4	4	4	40	100
Responden 5	4	4	4	4	4	4	3	4	4	4	39	97.5
Responden 6	3	3	4	2	2	3	2	4	4	3	30	75
Responden 7	3	4	4	4	3	4	4	4	4	4	38	95
Responden 8	3	4	3	4	3	4	3	3	4	4	35	87.5
Responden 9	3	3	3	4	3	4	3	3	4	0	30	75
Responden 10	3	2	3	3	3	3	3	3	4	3	30	75
Jumlah											827.5	
Rata-rata											82.75	

Tabel 6. 37 Hasil Perhitungan Responden Admin

Nama	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Jumlah	Skor
Responden 1	4	1	2	4	4	4	2	2	4	4	31	77.5
Responden 2	4	1	2	0	4	3	3	3	4	3	27	67.5
Jumlah											145	
Rata-rata											72.5	

Setelah dilakukan perhitungan, jumlah skor SUS untuk pelanggan yang didapatkan berjumlah 827.5 dan kemudian dibagi 10 maka didapatkan rata-rata sebesar 82.75 dimana termasuk dalam kategori *grade A* dengan *adjective ratings Excellent*. Sedangkan, jumlah skor SUS untuk admin adalah 827.5 dan kemudian dibagi 2 maka didapatkan rata-rata sebesar 72.5 dimana termasuk dalam kategori *grade B* dengan *adjective ratings Good*.

BAB 7 PENUTUP

7.1 Kesimpulan

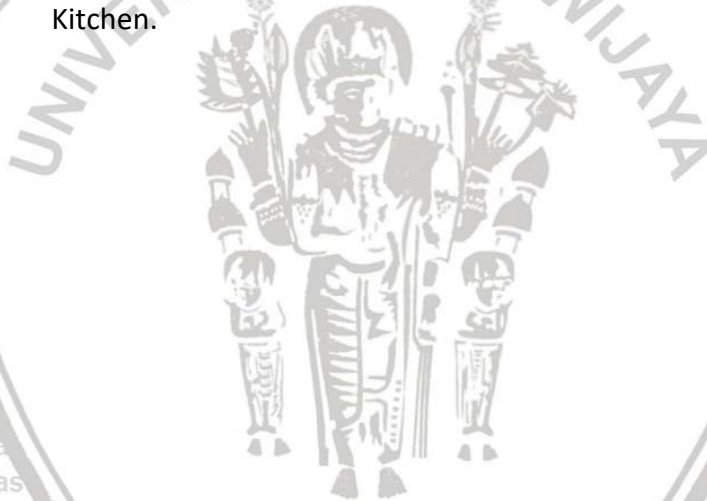
Setelah melakukan penelitian yang telah melalui tahapan-tahapan *System Development Life Cycle (SDLC)* menggunakan metode *waterfall*. Tahapan-tahapan tersebut ialah analisis kebutuhan, perancangan sistem, implementasi sistem, dan pengujian sistem. Penelitian ini menghasilkan kesimpulan sebagai berikut:

1. Hasil analisis definisi kebutuhan menghasilkan 2 proses bisnis utama yang digambarkan dalam BPMN yaitu pembelian kupon dan pemesanan menu. Kemudian analisis kebutuhan mengidentifikasi 2 aktor yaitu pelanggan dan admin, 18 persyaratan fungsional, dan 2 persyaratan non-fungsional. Dalam proses ini menghasilkan beberapa dokumen yaitu *use case diagram* dan *use case specification* sebanyak 18 *use case*.
2. Hasil perancangan sistem dilakukan berdasarkan hasil analisis kebutuhan. Tahap ini menghasilkan dokumen berupa 18 *activity diagram* untuk memodelkan skenario dari *use case*, 18 *sequence diagram* untuk menggambarkan komunikasi antar objek, *class diagram* untuk menggambarkan aspek statis sistem, *entity relationship diagram* dengan 6 tabel untuk menggambarkan bentuk *database*, dan 26 desain antarmuka sebagai patokan tampilan pembuatan sistem.
3. Implementasi sistem informasi pemesanan menggunakan konsep *Model View Controller (MVC)* dan optimasi web menggunakan *Progressive Web App (PWA)*. Dalam MVC yang bertugas sebagai *view* adalah *React Js Library* versi 17 berfungsi menampilkan dan berinteraksi dengan aktor. Sedangkan, *controller* dan *model* ditugaskan kepada *Codeigniter Framework* versi 4 berfungsi sebagai pengatur data sekaligus penghubung dengan *database*, terdapat 7 kelas *controller* dan 7 kelas *model*.
4. Pengujian sistem dilakukan menggunakan *validation testing* dan *system usability scale*. *Validation testing* dilakukan untuk mengetahui sistem berjalan sesuai dengan hasil yang diharapkan atau tidak, sedangkan *system usability scale* untuk mengukur tingkat *usability* atau kegunaan dari sistem yang telah dibuat. Dalam pengujian validasi menyatakan 31 dari 31 skenario uji berjalan sesuai harapan dan dapat disimpulkan seluruh fungsi sistem telah sesuai dengan persyaratan sistem. Sedangkan dalam pengujian *usability* menghasilkan tingkat *usability* sisi pelanggan menghasilkan nilai 82.75 (*adjective rating excellent*) dan sisi admin menghasilkan nilai 72.5 (*adjective rating good*).

7.2 Saran

Setelah penelitian pengembangan sistem informasi pemesanan Daily Catering Senjani Kitchen dilakukan, didapatkan beberapa saran untuk penelitian selanjutnya, yaitu:

1. Meskipun sistem informasi pemesanan Daily Catering ini berjalan dengan baik dan dapat mempermudah pekerjaan admin dan pelanggan. Sistem informasi ini masih belum mencakup pekerja lainnya yang ada di Senjani Kitchen, contohnya seperti belum adanya sistem informasi yang baik untuk menghubungkan antara pelanggan dengan kurir, ataupun staf dapur dengan pemasok bahan makanan. Diharapkan penelitian selanjutnya untuk memperluas cakupan permasalahan dan aktor yang terlibat.
2. Meskipun sistem informasi yang telah dibangun menggunakan teknologi *web* yaitu *Progressive Web App* telah memfasilitasi berbagai *platform*. Namun bila pelanggan Senjani Kitchen semakin banyak dan Senjani Kitchen lebih mempunyai kemampuan untuk memperluas *platform* pemesanan, sistem dapat dikembangkan khusus untuk *native mobile* (Android & iOS) untuk memberi pilihan dan kenyamanan lebih kepada pelanggan Senjani Kitchen.



DAFTAR REFERENSI

- Akbar, Mohammad. Pertimbangan Memilih Progressive Web Apps Ketimbang Merilis Aplikasi [online]. Tersedia di <<https://socs.binus.ac.id/2019/12/30/pertimbangan-memilih-progressive-web-apps-ketimbang-merilis-aplikasi/#:~:text=PWA%20memiliki%20kelebihan%20dapat%20dikunjungi,ketika%20perusahaan%20beroperasi%20di%20Indonesia>> [diakses 24 Mei 2021]
- Aknovia, 2018. Peranan Penting Sistem Informasi Dalam Perusahaan. [online] Tersedia di <<https://garudacyber.co.id/artikel/223-peranan-penting-sistem-informasi-dalam-perusahaan>> [diakses 2 Februari 2021]
- Arviana, Geofanni Nerissa. Kenali Apa Itu Aplikasi Web dan Kelebihannya Dibanding Aplikasi Mobile [online]. Tersedia di <<https://glints.com/id/lowongan/aplikasi-web-adalah/>> [diakses 24 Mei 2021]
- Booch, G., et al. 2007. Object Oriented Analysis and Design with Application. Boston.
- Codeigniter. 2021. Codeigniter Rocks. [online] Tersedia di <<https://www.codeigniter.com/>> [diakses: 4 Februari 2021]
- Dennis, A., Wixom, B.H. & Roth, R.M., 2012. System Analysis and Design. New York: John Wiley & Sons, Inc.
- Grechanik, M., McKinley, K. S., & Perry, D. E. (2007, September). Recovering and using use-case-diagram-to-source-code traceability links. In Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering (pp.95-104).
- Jogiyanto, H. M., 2008. Metodologi Penelitian Sistem Informasi. Yogyakarta: Andi Offset.
- Kim, S. K., & David, C. (1999, October). Formalizing the UML class diagram using Object-Z. In International Conference on the Unified Modeling Language (pp.83-98). Springer, Berlin, Heidelberg.
- Laudon, K.C. & Laudon, J.P., 2014. Management Information Systems : Managing the Digital Firm Thirteenth. Harlow: Pearson Education Limited
- Nurwanto. 2019. Penerapan Progressive Web Application (PWA) pada E-Commerce, [e-journal]. Tersedia melalui: Preferensian Universitas Muhammadiyah Ponorogo <https://www.researchgate.net/publication/335504148_Penerapan_Progressive_Web_Application_PWA_pada_E-Commerce> [diakses: 4 Februari 2021]

O'Brien, J. & Marakas, 2010. Introduction to Information Systems. 15th Edition. 15 ed. New York: Paul Ducham.

Object Management Group. 2013. Business Process Model and Notation (BPMN) [e-book]. Tersedia melalui: object management group <<https://www.omg.org/spec/BPMN>>. [diakses 26 April 2021]

Oliver, Andre. 2020. 6 Prinsip Penting untuk Merancang Desain *User Interface* yang baik. [Online] tersedia di <<https://glints.com/id/lowongan/prinsip-desain-user-interface/>> [diakses 1 Maret 2021]

Osmani, Addy. 2015. Getting Started with Progressive Web Apps. [Online] Tersedia di <<https://developers.google.com/web/updates/2015/12/getting-started-PWA>> [diakses 3 Februari 2021]

Pressman, R.S., 2010. Software engineering : a practitioner's approach. 7th ed. New York: McGraw-Hill

Rahmalia, Nadiyah. 2021. Yuk, Kenalan dengan Wireframing untuk Desain UI/UX. [Online] Tersedia di <<https://glints.com/id/lowongan/wireframe-adalah/#:~:text=Secara%20sederhana%2C%20wireframe%20adalah,yang%20harus%20dipahami%20dengan%20baik>> [diakses 27 February 2021]

React. 2021. React Library JavaScript untuk membangun antarmuka pengguna. [online] Tersedia di <<https://id.reactjs.org/>> [diakses 29 Januari 2021]

Richard, Sam. 2020. What are Progressive Web Apps?. [online] Tersedi di <<https://web.dev/what-are-pwas/>> [diakses 14 Juli 2021]

Rosa, A.S, Shalahuddin, M., 2018. Rekyasa Perangkat Lunak Terstruktur dan Berorientasi Objek (Edisi Revisi). Bandung: Informatika

Sommerville, I., 2011. Software engineering. 9th ed. Boston: Pearson Education, Inc.

Usability gov. 2021. System Usability Scale (SUS). [online] Tersedia di <<https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>> [diakses 27 February 2021]

Visual Paradigm. 2021. *What is Entity Relationship Diagram (ERD)*. [online] Tersedia di <<https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/>> [diakses 1 Maret 2021]

Whitten, J.L. & Bentley, L.D., 2007. System Analysis and Design Methods. McGraw Hil.

LAMPIRAN HASIL WAWANCARA

Notulen Wawancara

Narasumber : Abdul Latif

Jabatan : CEO Senjani Kitchen

Tanggal Wawancara : 31 Januari 2021

Dokumentasi :



Pertanyaan & Jawaban :

1. Apa itu Senjani Kitchen?
Senjani Kitchen itu layanan catering makanan, untuk saat ini yang paling diminati ada 2 yaitu event catering dan daily catering. Kalau yang event catering itu catering makanan dengan jumlah pesanan porsi banyak, biasanya dipakai saat ada acara. Sedangkan daily catering itu, pemesanan perorangan atau keluarga, seperti go-food dan grab-food, cuman produksi dan pengiriman dari Senjani Kitchen.
2. Apakah ada permasalahan yang sedang dihadapi?
Ada masalah, cuman lebi th tepatnya pingin lebih efisien aja. Layanan yang paling sibuk itu yang Daily Catering karena setiap hari pasti ada. Selama pelanggan melakukan pemesanan, nge cek jadwal, dan lain-lain itu semua lewat WA dan Instagram. Jadi pelanggan itu harus selalu nunggu balasan untuk melakukan pemesanan.
3. Bagaimana proses bisnis Daily Catering?
Pertama pelanggan membeli jumlah dan jenis kupon melalui Whatsapp. Lalu menukarkan kupon dengan makanan sesuai dengan jadwal makanan yang ada. Selanjutnya pada hari pemesanan, kurir mengantarkan makanan ke rumah/ kos/ kantor sesuai dengan pesan an
4. Bagaimana cara mengakses Jadwal Makanan?
Biasanya lewat personal chat WA, instagram story dan Whatsapp story. Namun untuk pelanggan yang masih mempunyai kupon akan dikirimkan lewat personal chat WA oleh admin
5. Siapa saja konsumen dari Daily Catering?

Mahasiswa, ibu rumah tangga, dan karyawan

6. Siapa konsumen paling besar Daily Catering?
Paling banyak itu mahasiswa perempuan

7. Apa saja jenis kupon menu yang disediakan oleh Daily Catering?

Basic Catering, Reusable Catering, Deluxe Catering, Family Catering, dan Ala carte.



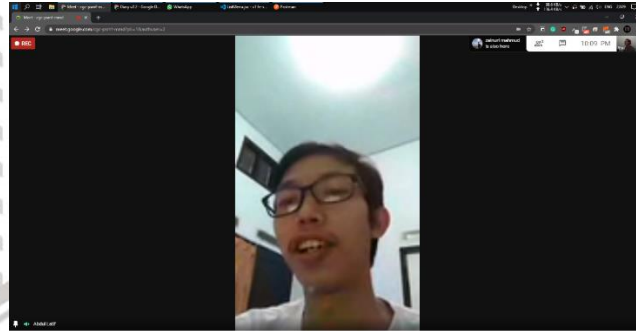
Notulen Wawancara

Narasumber : Abdul Latif

Jabatan : CEO Senjani Kitchen

Tanggal Wawancara : 24 Februari 2021

Dokumentasi :

**Pertanyaan & Jawaban :**

1. Bagaimana proses pemesanan Daily Catering saat ini?

Pelanggan *chat* Whatsapp admin Senjani Kitchen, lalu admin mengirimkan *Form Online Link* sesuai dengan jenis paket kupon apa yang diinginkan oleh pelanggan.

Setelah itu, pelanggan mengisi *form* nama, nomor telepon, dan identitas lainnya, di dalam *form* itu juga pelanggan memilih paket kupon yang sudah dikategorikan berdasarkan harga, jumlah kupon yang dapat ditukarkan, dan jenis nasi. Setelah pelanggan mengisi dan memilih di *form* tersebut pelanggan melakukan pembayaran dan mengirimkan bukti pembayaran ke admin Senjani Kitchen melalui Whatsapp. Setelah data pelanggan lengkap, admin akan mengirim data tersebut ke staf dapur dan kurir untuk diproses sesuai dengan jadwal pemesanan.

2. Apa saja fitur-fitur yang diharapkan ada dalam aplikasi pemesanan Daily Catering yang akan dibuat?

Yang pastinya pelanggan dapat memesan Daily Catering, bisa memilih lebih dari 1 alamat, mempunyai fitur kode unik untuk mempermudah verifikasi pembayaran, dapat memesan makanan dan jadwal lebih fleksibel, bisa mempunyai 2 paket, mengetahui jumlah kupon yang dimiliki dan catatan pemesanan dan makanan.

3. Berapa jumlah pelanggan saat ini?

Untuk total pelanggan yang terdaftar hingga saat ini hingga 1.500 pelanggan, namun untuk pelanggan aktif Daily Catering hingga lebih dari 100 pelanggan.

4. Media sistem informasi apa yang cocok untuk pelanggan?

Kalau bisa web yang dapat dibuka di *smartphone* saja karena untuk proses bisnis cara pemesanan saat ini pelanggan menggunakan aplikasi *chat* Whatsapp, sehingga pelanggan tidak harus melakukan pemasangan pada *smartphone* mereka. Pelanggan mempunyai spesifikasi yang bermacam-macam dikhawatirkan *internal storage* tidak mencukupi.

5. Selain Daily Catering, ada layanan lain yaitu Event Catering. Apakah perlu ada alur pemesanan khusus untuk Event Catering?

Untuk saat ini Event Catering belum mempunyai proses bisnis yang baku. Oleh karena itu, lebih baik untuk Event Catering pada aplikasi *web* nanti diarahkan ke Whatsapp saja.



Notulen Wawancara

Narasumber : Abdul Latif

Jabatan : CEO Senjani Kitchen

Tanggal Wawancara : 2 Maret 2021

Dokumentasi :



Pertanyaan & Jawaban :

1. Kenapa sistem informasi ini diperlukan?
Paling sering kejadian kesalahan admin dalam pencatatan pemesanan
2. Kesalahan apa yang terjadi?
Kesalahan dalam pencatatan. Contoh kesalahan kecil itu seperti pengguna memesan tanpa sambal namun tercatat menggunakan sambal. Sedangkan kesalahan yang cukup fatal yaitu minta tidak dikirim namun tercatat dikirim dan contoh lainnya minta dikirim ke kantor namun tercatat dikirim ke rumah
3. Seberapa sering kesalahan tersebut terjadi?
Seminggu sekali pasti ada, walaupun ini hanya hal remeh seperti sambal
4. Fitur yang diharapkan dari admin apa saja?
Verifikasi upload screenshot pembayaran dan mengelola jadwal menu
5. Untuk pelanggan harus nyaman dibuka di smartphone, sedangkan untuk admin bagaimana?
Smartphone juga agar bisa dibuka nyaman dimana saja
6. Berapa perhari rata-rata pengguna yang pakai layanan Daily Catering?
Sekitar 60 Orang
7. Ada gak hari atau tanggal tertentu pemesanan tiba-tiba banyak?

Dari pengalaman 2 tahun ini, biasanya hari senin paling banyak.

Sepertinya karena hari senin itu memulai pekan, kalau sabtu dan minggu itu biasanya mereka libur ya jadi tidak perlu daily catering.

8. Kira-kira ada maksimal pemesanan dalam sehari?

Sampai saat ini belum pernah benar-benar kewalahan, masih bisa diatasi.

Untungnya sistem pre order karena kami sudah tahu kapan pesanan akan

banyak, sebelum hari tersebut kami sudah siap-siap menambah pekerja sementara.

9. Kalau dari sisi pelanggan, kenapa sistem informasi ini harus ada?

Agar memberikan kebebasan ke pelanggan biar bisa daftar daily catering

kapan saja, tanpa harus nunggu balasan dari admin. Karena untuk

sekarang admin masih merangkap jadi tim packaging

