

**APLIKASI PENGELOLAAN TRANSAKSI DAN MANAJEMEN MENU  
DI CAFÉ BLACK TASTE BERBASIS WEB DAN MOBILE**

**(Skripsi)**

**Oleh**

**A Fathurrahman Z**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS LAMPUNG**

**BANDAR LAMPUNG**

**2019**

## **ABSTRACT**

### **APPLICATIONS FOR MANAGEMENT OF TRANSACTIONS AND MENU MANAGEMENT IN BLACK TASTE CAFE WEB AND MOBILE BASED**

**BY**

**A FATHURRAHMAN Z**

Black Taste Cafe is a restaurant that provides many variants of food and drinks. This cafe has many customers every day, they need to update their services to increase service in the cafe. Menu order that are still manual and sometimes there is a discrepancy between income and availability of menu items and ingredients are a problem at this time. So, to resolving the problem, an application was designed specifically for cafe workers. This application can be used to manage menu ordering processes from customers to the kitchen, regulate the availability ingredients, process payment transactions, calculate total revenue from all transactions, and make a shopping list of materials that was empty. The results of this study are: (1) The resulting application can manage the availability of menus with ingredients and make payment transactions as expected. (2) Functional testing using the Black Box Testing method and the Equivalence Partitioning (EP) technique, assessment using questionnaires with the calculation of questionnaire results using the Likert Scale method which obtained an average value of 94%, this value is included in the category of "Very Good". From the assessment results obtained, this application runs very well and can be used at Black Taste Cafe.

**Keywords:** Black Box Testing, Equivalence Partitioning, Likert Scale.

## **ABSTRAK**

### **APLIKASI PENGELOLAAN TRANSAKSI DAN MANAJEMEN MENU DI CAFÉ BLACK TASTE BERBASIS WEB DAN MOBILE**

**OLEH**

**A FATHURRAHMAN Z**

Cafe Black Taste adalah tempat makan yang menyediakan berbagai makanan dan minuman dengan berbagai varian dan harga. Kafe ini menerima banyak kunjungan pelanggan setiap harinya dan perlu adanya pembaharuan pelayanan untuk meningkatkan pelayanan di kafe tersebut. Pemesanan menu yang masih manual dan sering adanya ketidaksesuaian antara pendapatan dengan ketersediaan stok menu dan bahan menjadi masalah pada saat ini. Maka, untuk mengatasi masalah itu dibangun sebuah aplikasi yang diperuntukkan khusus bagi pekerja kafe. Aplikasi ini dapat digunakan untuk mengatur proses pemesanan menu dari pelanggan sampai ke dapur, mengatur ketersediaan stok bahan makanan dan minuman, melakukan proses transaksi pembayaran, menghitung total pendapatan dari semua transaksi, dan membuat daftar belanja dari bahan yang sudah habis. Hasil dari penelitian ini adalah: (1) Aplikasi yang dihasilkan dapat mengelola ketersediaan menu dengan bahan serta melakukan pembayaran transaksi sesuai dengan yang diharapkan. (2) Pengujian fungsional menggunakan metode Black Box Testing dan teknik Equivalence Partitioning (EP), penilaian menggunakan kuisioner dengan perhitungan hasil kuisioner menggunakan metode Skala Likert yang memperoleh nilai rata-rata sebesar 94%, nilai ini termasuk dalam kategori "Sangat Baik". Dari hasil penilaian yang didapat maka aplikasi ini berjalan sangat baik dan dapat digunakan di Cafe Black Taste.

**Kata Kunci:** *Black Box Testing, Equivalence Partitioning, Skala Likert.*

**APLIKASI PENGELOLAAN TRANSAKSI DAN MANAJEMEN MENU  
DI CAFÉ BLACK TASTE BERBASIS WEB DAN MOBILE**

**Oleh**

**A Fathurrahman Z**

**Skripsi**

Sebagai Salah Satu Syarat untuk Mencapai Gelar  
**SARJANA KOMPUTER**

pada

Jurusan Ilmu Komputer  
Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG**

**2019**

**Judul Skripsi : APLIKASI PENGELOLAAN TRANSAKSI  
DAN MANAJEMEN MENU DI CAFÉ  
BLACK TASTE BERBASIS WEB DAN  
MOBILE**

**Nama Mahasiswa : A Fathurrahman Z**

**Nomor Pokok Mahasiswa : 1317051001**

**Jurusan : Ilmu Komputer**

**Fakultas : Matematika dan Ilmu Pengetahuan Alam**

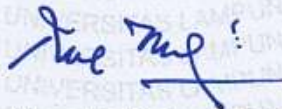
**MENYETUJUI**

**1. Komisi Pembimbing**



**Febi Eka Febriansyah, M.T.**  
NIP 19800219 200604 1 001

**2. Ketua Jurusan Ilmu Komputer**



**Dr. Ir. Kurnia Muludi, M.S.Sc.**  
NIP 19640616 198902 1 001



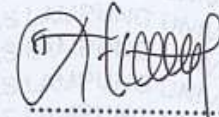
**MENGESAHKAN**

**1. Tim Penguji**

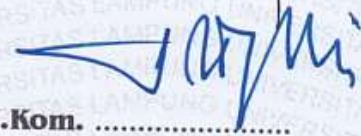
**Ketua : Febi Eka Febriansyah, M.T.**



**Penguji  
Bukan Pembimbing : Tristiyanto, S.Kom., M.I.S., Ph.D.**



**Penguji  
Bukan Pembimbing : Drs. Rd. Irwan Adi Pribadi, M.Kom.**



**2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam**



**Drs. Suratman, M.Sc.**  
NIP.19640604 199003 1 002

**Tanggal Lulus Ujian Skripsi : 1 Agustus 2019**

## PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya yang berjudul “Aplikasi Pengelolaan Transaksi Dan Manajemen Menu di Café Black Taste Berbasis Web Dan Mobile” merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila dikemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang saya terima.

Bandar Lampung, 12 Agustus 2019



  
**A. FATHURRAHMAN Z.**  
NPM. 1317051001

## **RIWAYAT HIDUP**



Penulis dilahirkan pada tanggal 03 Maret 1995 di Bukit Kemuning, Lampung sebagai anak kedua dari dua bersaudara dengan Ayah yang bernama Drs. Cacang Sutarsa (alm) dan Ibu bernama Sukarsih. Penulis menyelesaikan pendidikan dasar di SD Islam Ibnu Rusyd

Kotabumi selesai pada tahun 2007. Kemudian penulis melanjutkan ke pendidikan menengah pertama di SMPN 7 Kotabumi dan diselesaikan penulis pada tahun 2010, kemudian melanjutkan ke pendidikan menengah atas di SMAN 1 Bukit Kemuning kemudian pada kelas XI pindah ke SMAN 1 Martapura dan diselesaikan pada tahun 2013.

Pada tahun 2013 penulis terdaftar sebagai mahasiswa di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung. Selama dalam masa perkuliahan, penulis mengikuti organisasi internal Jurusan yaitu Himakom (Himpunan Mahasiswa Jurusan Ilmu Komputer) periode 2013/2014 hingga periode 2015/2016. Selama menjadi mahasiswa beberapa kegiatan yang dilakukan penulis antara lain pada bulan Februari 2016 penulis melaksanakan Kerja Praktik (KP) di PT PLN Cabang Tanjung Karang, dan pada bulan Juli 2016 Penulis melaksanakan Kuliah Kerja Nyata (KKN) di Desa Tanjung Ratu Kecamatan Selagai Lingga, Lampung Tengah.



## **PERSEMBAHAN**

*Segala puji Syukur atas berkah dan rahmat dari Allah Subhanallah 'Wata'ala,  
Kupersembahkan Skripsi Ini Untuk Orang-Orang Yang Selalu Kuharapkan  
Cinta dan Kasih Sayangnya...*

*Teruntuk Ayah yang tidak dapat menyaksikan anakmu ini menjadi sarjana,  
selalu ku ingat semua pesanmu dan do'aku tak pernah putus agar engkau  
tenang di alam sana...*

*Teruntuk Bapak dan Ibu yang tidak pernah memutus doa untuk anak-anaknya,  
terimakasih untuk dukungan dan tuntunan kalian, semoga kelak aku menjadi  
anak yang berguna dan membuat bangga kalian...*

*Teruntuk Kakakku, terima kasih atas teguran, motivasi dan dukungan dari  
engkau...*

*Teruntuk teman-temanku tercinta, terimakasih untuk selalu ada dan  
mendukung disaat senang dan sulit...*

*Almamater Tercinta,*

**UNIVERSITAS LAMPUNG**

## **MOTTO**

“Karena sesungguhnya sesudah kesulitan itu ada kemudahan”  
(Al-Insyirah 94:5)

“Waktu berlalu, Tanpa pernah menunggu, gunakan dia dengan bijaksana ”  
(Endang Soekamti)

“Aku tak gagal dalam ujian. Aku hanya telah menemukan 100 cara yang salah.  
(**Benjamin Franklin**)

“Hormati dan sayangi Ibumu, Maka Allah akan mudahkan rejekimu”  
(Penulis)

## SANWACANA

Assallamu'alaikum Warahmatullahi Wabarakatu..

Puji syukur kehadiran Allah Subhanallahu Wata'ala karena atas berkah dan rahmat-Nya penulis dapat menyelesaikan skripsi ini yang berjudul “Aplikasi Pengelolaan Transaksi Dan Manajemen Menu di Café Black Taste Berbasis Web Dan Mobile”. Skripsi ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Komputer di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

Dalam pelaksanaan dan penyusunan skripsi ini Penulis sangat berterima kasih dan memberikan penghargaan yang sedalam-dalamnya kepada seluruh pihak yang membantu menyelesaikan skripsi ini. Penulis ingin mengucapkan terima kasih dengan setulus hati terutama kepada:

1. Kedua Orangtua tercinta, Bapak, Ibu, dan Kakakku (Imam Rizki Fauzi Ananda Amd. Kep.) serta saudara-saudaraku yang selalu kusayangi dan ku kasihi yang selalu memberikan dukungan, masukan, motivasi, dan do'anya yang tak terhingga.
2. Bapak Drs. Suratman, M.Sc., selaku Dekan FMIPA Universitas Lampung.
3. Bapak Dr. Ir. Kurnia Muludi, M.S.Sc. selaku ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.

4. Febi Eka Febriansyah, M.T sebagai Pembimbing skripsi atas kesediaannya, kesabaran dan keikhlasannya untuk memberikan dukungan, bimbingan, nasihat, saran, dan kritik dalam proses penyelesaian skripsi ini.
5. Bapak Tristiyanto, S.Kom., M.I.S., dan Bapak Drs. Rd. Irwan Adi Pribadi, M.Kom selaku Dosen Pembahas skripsi, yang telah memberikan saran dan masukan guna penyempurnaan dalam penulisan skripsi ini.
6. Bapak dan Ibu Dosen Jurusan Ilmu Komputer yang telah memberikan ilmu dan pengalaman hidup selama penulis menjadi mahasiswa.
7. Ibu Ade Nora Maela, Mas Nofal, Mas Zai, dan Mbak Lia yang telah membantu memudahkan segala urusan administrasi penulis di Jurusan Ilmu Komputer.
8. Seluruh keluarga dan saudaraku yang telah memberikan dukungan selama proses perkuliahan yang tidak bisa disebutkan satu persatu.
9. Sahabat-sahabatku yang sudah seperti keluarga, Tegar, Dika, Hafis, Abdi, Radit, Maulidi, Fina, dan Yanak terimakasih banyak untuk selalu ada baik dalam keadaan suka maupun duka.
10. Teman teman yang telah membantu Angga, Dwi, dan Bintang yang telah membantu dan menyelesaikan masalah dalam pengkodean program terimakasih untuk semua bantuan dan masukannya.

11. Rekan-rekan Ilmu Komputer 2013 yang tidak bisa disebutkan satu persatu, terima kasih untuk segala dukungan, bantuan, serta kebersamaannya selama ini .

12. Semua pihak yang secara langsung maupun tidak langsung yang telah membantu dalam penyelesaian skripsi ini.

Penulis telah berusaha semaksimal mungkin dalam penulisan skripsi ini untuk mencapai suatu kelengkapan dan kesempurnaan. Penulis juga mengharapkan saran dan kritik yang bersifat membangun dari semua pihak. Akhirnya dengan segala kerendahan hati penulis berharap skripsi ini dapat memberikan manfaat, baik kepada penulis khususnya maupun kepada pembaca pada umumnya.

Bandar Lampung, 12 Agustus 2019

Penulis,

A Fathurrahman Zaqiyuddin



## DAFTAR ISI

	Halaman
<b>DAFTAR ISI.....</b>	<b>xiv</b>
<b>DAFTAR TABEL .....</b>	<b>xvii</b>
<b>DAFTAR GAMBAR.....</b>	<b>xix</b>
<b>DAFTAR KODE PROGRAM .....</b>	<b>xxiii</b>
<b>I. PENDAHULUAN.....</b>	<b>1</b>
5.1 Latar Belakang.....	1
5.2 Rumusan Masalah.....	2
5.3 Batasan Masalah.....	2
5.4 Tujuan Penelitian.....	3
5.5 Manfaat Penelitian.....	3
<b>II. TINJAUAN PUSTAKA.....</b>	<b>4</b>
5.1 Kualitas Pelayanan.....	4
5.2 Pengertian Aplikasi.....	5
5.3 Pengertian Aplikasi <i>Mobile</i> .....	6
5.4 Pengertian Android .....	7
5.5 Pengertian Java .....	8

5.6 Android Studio.....	10
5.7 Extensible Markup Language (XML).....	10
5.8 Application Programming Interface (API).....	11
5.9 Java Script Object Notation (JSON).....	11
5.10 Software Development Kit (SDK).....	11
5.11 Metode Waterfall.....	11
5.12 PHP (Hypertext Preprocessor).....	13
5.13 Database dan MySQL.....	14
5.14 HTML.....	15
5.15 Cascading Style Sheet (CSS).....	16
5.16 Unified Modelling Language (UML) .....	16
5.17 Metode Pengujian Sistem .....	20

### **III. METODOLOGI PENELITIAN .....23**

3.1. Waktu dan Tempat Penelitian.....	23
3.2. Alat Pendukung Penelitian.....	24
3.3. Metodologi Penelitian .....	25
3.3.1. Diagram Alir Penelitian .....	25
3.3.2. Metode Pengembangan Aplikasi .....	28
A. Analisa Kebutuhan .....	28
B. Desain .....	29
C. Pengkodean Aplikasi .....	54
D. Pengujian Sistem.....	54

<b>IV. HASIL DAN PEMBAHASAN .....</b>	<b>59</b>
4.1. Analisa Kebutuhan .....	59
4.2. Implementasi Sistem .....	60
4.2.1. Halaman <i>Login</i> Android.....	60
4.2.2. Halaman Utama.....	61
4.2.3. Halaman <i>Input</i> Pesanan.....	62
4.2.4. Halaman Daftar Pesanan .....	73
4.2.5. Halaman <i>Login</i> Web .....	84
4.2.6. Halaman Beranda .....	85
4.2.7. Halaman Daftar Pesanan .....	86
4.2.8. Halaman Daftar Belanja.....	98
4.2.9. Halaman Data Menu .....	100
4.2.10. Halaman Data Transaksi .....	103
4.2.11. Halaman Data Bahan.....	107
4.2.12. Halaman Data Meja.....	109
4.2.13. Halaman Data Karyawan .....	111
4.2.14. Halaman Pengaturan .....	113
4.3. Pengujian Sistem.....	116
4.3.1 Pengujian Internal .....	116
4.3.1.1. Pengujian Fungsional.....	116
A. Pengujian Versi Android.....	116
B. Pengujian Ukuran Layar .....	117
C. Pengujian <i>User Interface</i> .....	118
D. Pengujian Menu Aplikasi.....	119

4.3.2. Pengujian Eksternal .....	121
4.3.2.1 Hasil Analisa Kuisisioner .....	122
<b>V. PENUTUP</b> .....	<b>124</b>
5.1 Kesimpulan .....	124
5.2 Saran .....	124
<b>DAFTAR PUSTAKA</b> .....	<b>126</b>

## DAFTAR TABEL

Tabel		Halaman
Tabel 3.1.	Waktu Kegiatan Peneletian .....	23
Tabel 3.2.	Daftar Pengujian <i>Equivalence Partititoning</i> .....	56
Tabel 4.1.	Pengujian Versi Android .....	117
Tabel 4.2.	Pengujian Ukuran Layar .....	117
Tabel 4.3.	Pengujian User Interface .....	118
Tabel 4.4.	Pengujian Fungsi dari Menu Aplikasi Android.....	119
Tabel 4.5.	Pengujian Fungsi dari Menu Aplikasi Web.....	120
Tabel 4.6.	Kriteria penilaian dan bobot nilai .....	121
Tabel 4.7.	Hasil Penilaian Kuisisioner .....	121
Tabel 4.8.	Kriteria <i>index</i> penilaian hasil kuesioner .....	123



## DAFTAR GAMBAR

Gambar	Halaman
Gambar 2.1. <i>Waterfall Model</i> .....	12
Gambar 2.2. <i>Usecase</i> .....	17
Gambar 2.3. Aktor .....	18
Gambar 2.4. Asosiasi .....	18
Gambar 3.1. Diagram Alir Penelitian .....	26
Gambar 3.2. <i>Use Case Diagram</i> .....	30
Gambar 3.3. <i>Class Diagram</i> Sistem .....	31
Gambar 3.4. <i>Activity Diagram</i> Pesan Menu .....	32
Gambar 3.5. Rancangan <i>Interface</i> Halaman Utama, Daftar Menu, Data Pembeli, Konfirmasi, dan Rincian Pembelian .....	32
Gambar 3.6. <i>Sequence Diagram</i> Pesan Menu .....	33
Gambar 3.7. <i>Activity Diagram</i> Melihat Daftar Pesanan .....	34
Gambar 3.8. Rancangan <i>Interface</i> Halaman Utama, Daftar Pesanan, dan Rincian Pesanan .....	35
Gambar 3.9. <i>Sequence Diagram</i> Melihat Daftar Pesanan .....	36
Gambar 3.10. <i>Activity Diagram</i> Transaksi Pembayaran .....	37
Gambar 3.11. Rancangan <i>Interface</i> Cetak Pembayaran .....	38

Gambar 3.12. <i>Sequence Diagram</i> Transaksi Pembayaran .....	39
Gambar 3.13. <i>Activity Diagram</i> Data Transaksi .....	39
Gambar 3.14. Rancangan <i>Interface</i> Data Transaksi.....	39
Gambar 3.15. <i>Sequence Diagram</i> Riwayat Pembelian .....	40
Gambar 3.16. <i>Activity Diagram</i> Tambah Data <i>Login User</i> .....	40
Gambar 3.18. <i>Sequence Diagram</i> Tambah <i>Login User</i> .....	41
Gambar 3.17. Rancangan <i>Interface</i> Tambah Data <i>User</i> dan Form Tambah Data <i>User</i> .....	41
Gambar 3.19. <i>Activity Diagram</i> Edit data <i>Login</i> .....	42
Gambar 3.20. Rancangan <i>Interface</i> Edit Data <i>User</i> .....	43
Gambar 3.21. <i>Sequence Diagram</i> Edit data <i>Login</i> .....	43
Gambar 3.22. <i>Activity Diagram</i> Hapus data <i>Login User</i> .....	44
Gambar 3.23. Rancangan <i>Interface</i> Hapus Data <i>User Login User</i> .....	45
Gambar 3.24. <i>Sequence Diagram</i> Hapus data <i>Login User</i> .....	45
Gambar 3.25. <i>Activity Diagram</i> Tambah Menu .....	46
Gambar 3.26. Perancangan <i>Interface</i> Data Menu dan Tambah Data Menu.....	47
Gambar 3.27. <i>Sequence Diagram</i> Tambah Data Menu.....	47
Gambar 3.28. <i>Activity Diagram</i> Edit Menu .....	48
Gambar 3.29. Perancangan <i>Interface</i> Edit Menu .....	48
Gambar 3.30. <i>Sequence Diagram</i> Edit Menu.....	49
Gambar 3.31. <i>Activity Diagram</i> Hapus Menu .....	50
Gambar 3.32. Perancangan <i>Interface</i> Hapus Menu.....	50
Gambar 3.33. <i>Sequence Diagram</i> Hapus Menu .....	50
Gambar 3.34. <i>Activity Diagram</i> Data Bahan.....	51

Gambar 3.35. <i>Rancangan Interface</i> Data Bahan.....	51
Gambar 3.36. <i>Sequence Diagram</i> Daftar Belanja .....	52
Gambar 3.37. <i>Activity Diagram</i> Daftar Belanja .....	53
Gambar 3.38. Perancangan <i>Interface</i> Daftar Belanja.....	53
Gambar 3.39. <i>Sequence Diagram</i> Daftar Belanja .....	53
Gambar 4.1. <i>Halaman Login Android</i> .....	61
Gambar 4.2. Halaman Utama .....	62
Gambar 4.3. Halaman Input Nomor Meja.....	63
Gambar 4.4. Tampilan Halaman Menu Makanan Dan Minuman.....	66
Gambar 4.5. Tampilan <i>Popup</i> Dialog Pesanan .....	69
Gambar 4.6. Tampilan Halaman Rincian Pesanan.....	70
Gambar 4.7. Tampilan Halaman Daftar Pesanan dan ubah Nomor Meja .....	74
Gambar 4.8. Tampilan Rincian Daftar Pesanan .....	78
Gambar 4.9. Tampilan Halaman Ubah dan Hapus Menu Pesanan .....	84
Gambar 4.10. Tampilan Halaman <i>Login Web</i> .....	85
Gambar 4.11. Tampilan Halaman Beranda <i>Ownwer</i> dan Manajer .....	86
Gambar 4.12. Tampilan Halaman Beranda <i>Waiter</i> .....	86
Gambar 4.13. Tampilan Halaman Daftar Pesanan .....	87
Gambar 4.14.. Tampilan Halaman Tambah Menu.....	90
Gambar 4.15.. Tampilan Halaman Rincian Pesanan.....	92
Gambar 4.16.. Tampilan Halaman Cetak Struk .....	95
Gambar 4.17. Tampilan Halaman Daftar Belanja.....	98
Gambar 4.18. Tampilan Halaman Data Menu .....	101
Gambar 4.19. Tampilan Halaman Data Transaksi .....	104

Gambar 4.20. Tampilan Halaman Data Bahan.....	107
Gambar 4.21. Tampilan Halaman Data Meja.....	110
Gambar 4.22. Tampilan Halaman Data Karyawan .....	112
Gambar 4.23. Tampilan Halaman Pengaturan .....	114

## DAFTAR KODE PROGRAM

Kode Program	Halaman
Kode Program 4.1 fungsi untuk mengambil data meja .....	63
Kode Program 4.2 tampilan data meja.....	64
Kode Program 4.3 fungsi untuk mengambil data menu .....	66
Kode Program 4.4 tampilan data menu .....	68
Kode Program 4.5 fungsi untuk mengambil data rincian pesanan .....	70
Kode Program 4.6 tampilan rincian pesanan .....	72
Kode Program 4.7 fungsi untuk mengambil data daftar pesanan .....	74
Kode Program 4.8 tampilan daftar pesanan.....	76
Kode Program 4.9 fungsi untuk mengambil data rincian daftar pesanan.....	79
Kode Program 4.10 tampilan rincian daftar pesanan.....	81
Kode Program 4.11 fungsi pengurangan data bahan .....	82
Kode Program 4.12 <i>controllers</i> untuk mengambil data daftar pesanan.....	87
Kode Program 4.13 <i>views</i> untuk menampilkan daftar pesanan .....	88
Kode Program 4.14 <i>controllers</i> untuk menambahkan menu daftar pesanan ....	90
Kode Program 4.15 <i>views</i> untuk menampilkan tambah menu daftar pesanan .....	91
Kode Program 4.16 <i>controllers</i> untuk mengambil data rincian	



daftar pesanan .....	92
Kode Program 4.17 <i>views</i> untuk menampilkan rincian daftar pesanan .....	93
Kode Program 4.18 <i>controllers</i> untuk mengambil data transaksi untuk cetak struk daftar pesanan .....	96
Kode Program 4.19 <i>views</i> untuk menampilkan cetak struk daftar pesanan.....	97
Kode Program 4.20 <i>controllers</i> untuk mengambil data daftar belanja.....	99
Kode Program 4.21 <i>views</i> untuk menampilkan daftar belanja .....	99
Kode Program 4.22 <i>controllers</i> untuk mengambil data menu.....	101
Kode Program 4.23 <i>views</i> untuk menampilkan data menu.....	102
Kode Program 4.24 <i>controllers</i> untuk mengambil data transaksi.....	104
Kode Program 4.25 <i>views</i> untuk menampilkan data transaksi .....	105
Kode Program 4.26 <i>controllers</i> untuk mengambil data bahan .....	108
Kode Program 4.27 <i>views</i> untuk menampilkan data bahan .....	108
Kode Program 4.28 <i>controllers</i> untuk mengambil data meja.....	110
Kode Program 4.29 <i>views</i> untuk menampilkan data meja.....	111
Kode Program 4.30 <i>controllers</i> untuk mengambil data karyawan .....	112
Kode Program 4.31 <i>views</i> untuk menampilkan data karyawan .....	113
Kode Program 4.32 <i>controllers</i> untuk mengambil data pengaturan.....	114
Kode Program 4.33 <i>views</i> untuk menampilkan data pengaturan.....	115

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Bisnis makanan atau kuliner adalah salah satu bisnis yang cukup banyak dan berkembang dengan cepat. Selain karena cukup menguntungkan, makanan adalah kebutuhan pokok manusia dan banyak orang yang ingin mencari suasana lain sehingga mereka terkadang memilih makan di luar rumah seperti makan di *cafe* atau restoran.

Kualitas rasa makanan adalah faktor utama yang menentukan banyaknya pembeli di suatu *cafe* atau restoran. Namun, pelayanan terhadap konsumen harus lebih baik dan diprioritaskan juga. Kualitas pelayanan khususnya pada *cafe* atau restoran sangat berpengaruh terhadap kepuasan pelanggan, sehingga pelanggan tidak akan kecewa atau berpindah ke restoran lain.

Pemilik *cafe* atau restoran perlu memanfaatkan teknologi *smartphone* Android agar pelayanan lebih efektif dan efisien dengan membuat sebuah aplikasi yang dapat mempermudah pemesanan menu di *cafe* atau restoran. Aplikasi ini diharapkan dapat membantu dan mempermudah pelayan mencatat menu pesanan dan mengirimkan data ke koki yang berada di dapur, mempermudah perhitungan pembayaran pada kasir.

Terkadang terdapat ketidaksesuaian antara pendapatan yang masuk dengan persediaan stok bahan yang tersedia. Seringkali pendapatan yang masuk besar dan stok bahan yang tersedia masih cukup, tetapi dalam satu waktu stok bahan habis namun pendapatannya kecil. Hal ini dapat terjadi karena tidak adanya kontrol persediaan stok bahan untuk satu menu. Untuk itulah perlu adanya aplikasi untuk mengontrol stok bahan makanan agar bahan-bahan yang digunakan dapat stabil dan seimbang dengan pendapatan yang ada.

Hal inilah yang menjadi latar belakang untuk dikembangkannya Aplikasi Pengelolaan Transaksi dan Manajemen Menu dengan mengambil studi kasus di *Cafe Black Taste*. Sistem ini berbasis web dan mobile.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang tersebut, maka dapat dirumuskan suatu permasalahan dalam penelitian ini adalah bagaimana mengembangkan suatu aplikasi untuk pemesanan makanan dan minuman pada *cafe* atau restoran yang berbasis Android dan *web server* yang dapat meningkatkan dan membantu pelayanan agar lebih efektif dan efisien.

## **1.3 Batasan Masalah**

Adapun batasan masalah dalam penelitian ini adalah:

1. Aplikasi yang akan dikembangkan berbasis mobile dan web.
2. Aplikasi ini hanya mengelola pemesanan makanan dan minuman di *Cafe Black Taste*.

3. Aplikasi ini hanya dapat digunakan oleh karyawan atau pegawai *Cafe Black Taste*, bukan digunakan untuk pelanggan.
4. Aplikasi ini mengelola pelayanan pemesanan makan, pembayaran, ketersediaan menu, memuat daftar bahan dan membuat daftar belanja untuk ketersediaan bahan yang habis.

#### **1.4 Tujuan Penelitian**

Adapun tujuan dari penelitian ini adalah membangun sebuah aplikasi yang dapat membantu pelayanan transaksi pada *cafe* atau restoran sehingga pelayanan lebih efektif dan efisien.

#### **1.5 Manfaat Penelitian**

Adapun manfaat dari penelitian ini adalah:

1. Aplikasi yang dibuat diharapkan mampu meningkatkan pelayanan dan membantu pelayan agar lebih efektif dan efisien.
2. Inovasi baru dalam pelayanan *cafe* atau restoran yang diharapkan mampu menjadi daya tarik pelanggan.
3. Aplikasi diharapkan mampu meningkatkan manajemen *cafe* agar lebih baik dan terstruktur.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Kualitas Pelayanan**

Kualitas suatu produk atau pelayanan perlu mendapat perhatian besar dari manajer, karena kualitas mempunyai hubungan langsung dengan kemampuan bersaing dan tingkat keuntungan yang diperoleh perusahaan.

Ada 5 dimensi utama kualitas pelayanan sebagai berikut:

- A.** Realibilitas (*Reliability*), berkaitan dengan kemampuan perusahaan untuk memberikan layanan yang akurat sejak pertama kali tanpa membuat kesalahan apapun dan menyampaikan jasanya sesuai dengan waktu yang disepakati.
- B.** Daya Tanggap (*Responsiveness*), berkenaan dengan kesediaan dan kemampuan para karyawan untuk membantu para pelanggan dan merespon permintaan mereka, serta menginformasikan kapan jasa akan diberikan dan kemudian memberikan jasa secara cepat.
- C.** Jaminan (*Assurance*), yakni perilaku para karyawan mampu menumbuhkan kepercayaan pelanggan terhadap perusahaan dan perusahaan bisa menciptakan rasa aman bagi para pelanggannya. Jaminan juga berarti bahwa para karyawan selalu bersikap sopan dan



menguasai pengetahuan dan ketrampilan yang dibutuhkan untuk menangani setiap pertanyaan atau masalah pelanggan.

- D. Empati (*Emphaty*), berarti bahwa perusahaan memahami masalah para pelanggannya dan bertindak demi kepentingan pelanggan, serta memberikan perhatian personal kepada para pelanggan dan memiliki jam operasi yang nyaman.
- E. Bukti Fisik (*Tangibles*), berkenaan dengan daya tarik fasilitas fisik, perlengkapan, dan material yang digunakan perusahaan, serta penampilan karyawan (Tjiptono dan Chandra, 2005).

## **2.2 Pengertian Aplikasi**

Menurut Buyens (2001), aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas. Misalnya termasuk perangkat lunak perusahaan, software akuntansi, perkantoran, grafis perangkat lunak, dan pemutar media. Menurut Kamus Besar Bahasa Indonesia (KBBI) Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu.

Dari beberapa pendapat di atas dapat disimpulkan bahwa aplikasi merupakan software yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data. Program aplikasi merupakan program siap pakai. Program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Contoh aplikasi ialah program pemroses kata dan Web Browser. Aplikasi akan

menggunakan sistem operasi (OS) komputer dan aplikasi yang lainnya yang mendukung.

### **2.3 Pengertian Aplikasi *Mobile***

Menurut Buyens (2001) aplikasi *mobile* berasal dari kata *application* dan *mobile*. *Application* yang artinya penerapan, lamaran, dan penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju sedangkan *mobile* dapat diartikan sebagai perpindahan dari suatu tempat ke tempat yang lain. Kata *mobile* mempunyai arti bergerak atau berpindah, sehingga aplikasi *mobile* menurut Purnama (2010) adalah sebutan untuk aplikasi yang berjalan di *mobile device* .

Dengan menggunakan aplikasi *mobile*, dapat dengan mudah melakukan berbagai macam aktifitas mulai dari hiburan, berjualan, belajar, mengerjakan pekerjaan kantor, *browsing*, dan lain sebagainya. Pemanfaatan aplikasi *mobile* untuk hiburan paling banyak digemai oleh pengguna telepon seluler, karena dengan memanfaatkan adanya *fitur game*, *music player*, sampai *video player* membuat kita menjadi semakin mudah menikmati hiburan kapan saja dan dimanapun. Perangkat *mobile* memiliki banyak jenis dalam hal ukuran, *design layout*, tetapi mereka memiliki kesamaan karakteristik yang sangat berbeda dari *desktop system*. Perangkat *mobile* memiliki memori yang kecil.

## 2.4 Pengertian Android

Android menurut Nazaruddin (2012) merupakan sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Android umum digunakan di smartphone dan juga tablet PC. Fungsinya sama seperti sistem operasi Symbian di Nokia, iOS di Apple dan BlackBerry OS. Android tidak terikat ke satu merek Handphone saja, beberapa vendor terkenal yang sudah memakai Android antara lain Samsung, Sony Ericsson, HTC, Nexus, Motorola, dan lain-lain.

Android sudah bekerjasama dengan beberapa *smart mobile* seperti LG, Samsung, Sony Ericsson, dan lainnya. Sekitar September 2007 sebuah studi melaporkan bahwa Google mengajukan hak paten aplikasi telepon seluler, dan akhirnya Google mengenalkan Nexus One, salah satu jenis telepon pintar GSM yang menggunakan Android pada sistem operasinya. Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010). Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka, Android, perangkat bergerak (Mobile) yang merupakan modifikasi kernel Linux 2.6.

Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru. Banyak smartphone dan PC Tablet menggunakan sistem operasi dengan versi yang berbeda. Semakin tinggi versi, fiturnya semakin canggih dan banyak. Telepon pertama yang memakai sistem operasi Android adalah HTC Dream yang dirilis pada tanggal 22 Oktober 2008.

## 2.5 Pengertian Java

Menurut definisi Sun Microsystem, dalam buku Shalahuddin dan Rosa (2010) Java adalah nama sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer yang berdiri sendiri (*standalone*) ataupun pada lingkungan jaringan.

Sedangkan menurut Budi Raharjo (2010) Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal.

Aplikasi-aplikasi berbasis Java umumnya dikompilasi ke dalam p-code (*bytecode*) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang

memungkinkan aplikasi Java mampu berjalan di beberapa platform sistem operasi yang berbeda, Java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini Java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web. Adapun kelebihan dari Java yaitu:

1. *Multi-platform*
2. OOP (*Object Oriented Programming* - Pemrogram Berorientasi Objek)
3. Perpustakaan Kelas Yang Lengkap
4. Bergaya C++, memiliki sintaks seperti bahasa pemrograman C++ sehingga menarik banyak pemrogram C++ untuk pindah ke Java.
5. Pengumpulan sampah otomatis, sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas)

Sedangkan kekurangan Java yaitu:

1. Masih ada beberapa hal yang tidak kompatibel antara platform satu dengan *platform* lain.
2. Mudah didekompilasi. Dekompilasi adalah proses membalikkan dari kode jadi menjadi kode sumber.
3. Penggunaan memori yang banyak.

## 2.6 Android Studio

Android Studio adalah sebuah *Integrated Development Environment* (IDE) untuk mengembangkan aplikasi pada *platform* Android. Android Studio diumumkan pada 16 Mei 2013 pada konferensi Google I/O oleh *Product Manager Google*, Ellie Powers. Android Studio tersedia secara bebas di bawah Apache License 2.0. Android Studio berada di awal tahap akses preview mulai dari versi 0.1 pada Mei 2013, kemudian memasuki tahap beta mulai dari versi 0.8 yang dirilis pada Bulan Juni 2014. Android Studio dengan kemampuan yang stabil dirilis pada Bulan Desember 2014, mulai dari versi 1.0. Android Studio tersedia untuk diunduh pada Windows, Mac OS X dan Linux.

## 2.7 Extensible Markup Language (XML)

XML merupakan dasar terbentuknya *web service* yang digunakan untuk mendeskripsikan data. Pada level paling detail *web service* secara keseluruhan dibentuk dari XML. Fungsi utama dari XML adalah komunikasi antar aplikasi, integrasi data, dan komunikasi aplikasi eksternal dengan partner luaran. Dengan standarisasi XML, aplikasi-aplikasi yang berbeda dapat dengan mudah berkomunikasi antar satu dengan yang lain (Deviana, 2011).

## **2.8 *Application Programming Interface (API)***

API (*Application Programming Interface*) atau Antarmuka Pemrograman Aplikasi adalah sekumpulan perintah, fungsi, dan protokol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk sistem operasi tertentu (Ichwan dan Fifin, 2011)

## **2.9 *Java Script Object Notation (JSON)***

JSON (*Java Script Object Notation*) adalah format pertukaran data yang bersifat ringan, disusun oleh Douglas Crockford. Fokus JSON adalah pada representasi data di *website*. JSON dirancang untuk memudahkan pertukaran data pada situs dan merupakan perluasan dari fungsi-fungsi *javascript* (Chasseur dan Patel, 2013).

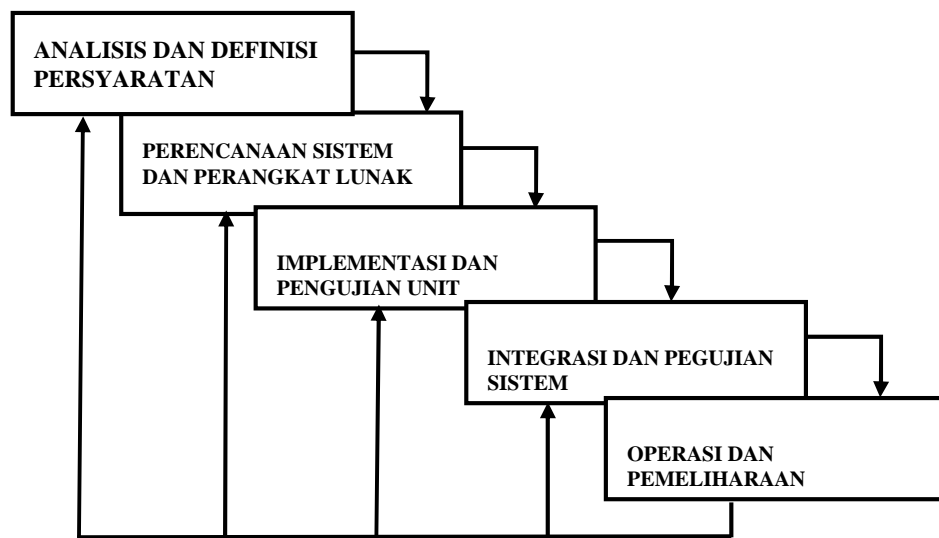
## **2.10 *Software Development Kit (SDK)***

Android SDK adalah *tools* API yang diperlukan untuk mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Sebagai *platform* aplikasi netral, Android memberi kesempatan bagi semua orang untuk membuat aplikasi yang dibutuhkan, yang bukan merupakan aplikasi bawaan *Handphone/Smartphone* (Safaat, 2011).

## **2.11 *Metode Waterfall***

Menurut Sommerville (2003) *Waterfall* Model merupakan salah satu model proses perangkat lunak yang mengambil kegiatan proses dasar seperti spesifikasi, pengembangan, validasi, dan evolusi dengan

mempresentasikannya sebagai fase-fase proses yang berbeda seperti analisis dan definisi persyaratan, perancangan perangkat lunak, implementasi dan pengujian unit, integrasi dan pengujian sistem, operasi, dan pemeliharaan. Pada Gambar 2.1, Sommerville (2003) menjelaskan bahwa tahap-tahap utama dari *Waterfall Model* adalah memetakan kegiatan kegiatan pengembangan dasar.



Gambar 2.1. *Waterfall Model* (Sommerville, 2003)

a. Analisis dan Definisi Persyaratan

Proses mengumpulkan informasi kebutuhan sistem/perangkat lunak melalui konsultasi dengan *user system*. Proses ini mendefinisikan secara rinci mengenai fungsi-fungsi, batasan dan tujuan dari perangkat lunak sebagai spesifikasi sistem yang akan dibuat.

b. Perancangan Sistem dan Perangkat Lunak

Proses perancangan sistem difokuskan pada empat atribut, yaitu struktur data, arsitektur perangkat lunak, representasi antar muka, dan detail (*algoritma*)



prosedural. Yang dimaksud struktur data adalah representasi dari hubungan logis antara elemen-elemen data individual.

#### c. Implementasi dan Pengujian Unit

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai program atau unit program. Kemudian pengujian unit melibatkan verifikasi bahwa setiap unit program telah memenuhi spesifikasinya.

#### d. Integrasi dan Pengujian Sistem

Unit program/program individual diintegrasikan menjadi sebuah kesatuan sistem dan kemudian dilakukan pengujian. Dengan kata lain, pengujian ini ditujukan untuk menguji keterhubungan dari tiap-tiap fungsi perangkat lunak untuk menjamin bahwa persyaratan sistem telah terpenuhi.

#### e. Operasi dan Pemeliharaan

Tahap ini biasanya memerlukan waktu yang paling lama. Sistem diterapkan (*di-install*) dan dipakai. Pemeliharaan mencakup koreksi dari beberapa kesalahan yang tidak ditemukan pada tahapan sebelumnya.

### **2.12 PHP (*Hypertext Preprocessor*)**

PHP merupakan akronim dari *Hypertext Preprocessor*. PHP merupakan bahasa pemrograman berbasis web yang sudah sering digunakan. Terdapat perbedaan antara *web* yang menggunakan PHP dan *web* yang hanya sekedar menggunakan HTML saja. Hal tersebut dapat dilihat pada proses saat *webserver* memenuhi permintaan *client* untuk menampilkan halaman *web*.

Pada halaman *web* yang hanya menggunakan HTML, server langsung mengirimkan halaman yang diminta oleh *client* dalam bentuk *script* HTML. Sedangkan pada *web* yang menggunakan PHP sebelum server mengirimkan *script* HTML kepada *client*, server membaca terlebih dahulu *script* PHP yang ada pada server tersebut kemudian mengirimkan hasil dari *script* PHP tersebut kepada *client* berupa HTML (Kadir, 2007).

### 2.13 Database dan MySQL

*Database* adalah kumpulan data yang tersusun secara sistematis sehingga dapat memudahkan pengguna dalam mengakses dan mengelola serta untuk mendapatkan informasi yang efektif dan efisien. *Database* dapat berdiri sendiri dan dapat juga digabung menjadi satu kesatuan, penggabungan antar *database* disebut dengan relasi, sebagai contoh data *costumer* digabung dengan data *order*. Untuk menggabungkan data harus ada penghubung yang disebut dengan *index*. Syarat dari *index* adalah tidak boleh ada data yang sama dan data yang digabung masing-masing harus mempunyai *index* yang sama.

DBMS (*Database Management System*) merupakan software yang banyak digunakan dalam pengolahan data. Contoh manajemen *database* adalah sistem manajemen *database* relasi. Komponen dalam sistem manajemen *database* relasi adalah:

1. *Database* sebagai tempat untuk menyimpan data yang terstruktur, data tersimpan dalam tabel, tabel tersebut terdiri dari baris dan kolom yang didalamnya terdapat *field* dan *record*.
2. *System Management* adalah *software* yang digunakan untuk mengelola database tersebut.
3. Relasi adalah hubungan antara tabel - tabel yang ada di dalam *database*.

Dari sekian banyak *database* yang ada, MySQL adalah *database* yang paling sering digunakan. Selain bersifat *open source* dan menggunakan *SQL*, MySQL bisa dijalankan diberbagai *platform* misalnya *Windows*, *Linux*, dan lain sebagainya. MySQL terkenal dengan pengolahan data yang cepat (Puspitosari, 2011).

## 2.14 HTML

Dokumen html adalah file teks murni yang dapat dibuat dengan editor teks sembarang. Dokumen html dikenal sebagai *web page*. Dokumen ini merupakan dokumen yang disajikan dalam *browser web surfer*. Dokumen ini umumnya berisi informasi ataupun *interface* aplikasi di dalam internet (Sidik, 2005). Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (*Standard Generalized Markup Language*), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium* (W3C). HTML dibuat

oleh kolaborasi Caillau TIM dengan Berners-lee Robert ketika mereka bekerja di CERN pada tahun 1989 (CERN adalah lembaga penelitian fisika energi tinggi di Jenewa).

## 2.15 CSS

*Cascading Style Sheet* (CSS) merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah web sehingga lebih terstruktur dan seragam. CSS bukan merupakan bahasa pemrograman. Pada umumnya CSS dipakai untuk memformat tampilan halaman web. CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran *border*, warna *border*, warna *hyperlink*, warna *mouse over*, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya. CSS adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan *perogrammer* untuk menampilkan halaman yang sama dengan format yang berbeda.

## 2.16 Unified Modelling Language (UML)

*Unified Modelling Language* (UML) menurut Martin Fowler (2005 : 1) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO). UML merupakan standar yang relatif terbuka yang dikontrol oleh *Object*

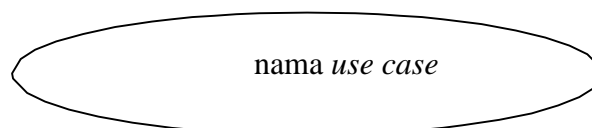
*Management Company* (OMG), sebuah konsorsium terbuka yang terdiri dari banyak perusahaan.

## 1. Use Case Diagram

*Use Case* menurut Martin Fowler (2005) adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use Case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan. *Use Case Diagram* menampilkan aktor mana yang menggunakan *use case* mana, *uses case* mana yang memasukkan *use case* lain dan hubungan antara aktor dan *use case*.

Simbol-simbol pada *use case diagram* adalah sebagai berikut (Shalahuddin dan Rosa, 2008).

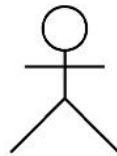
1. *Use case* adalah fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama *use case*. *Use case* dapat dilihat pada Gambar 2.2.



**Gambar 2.2** *Use case* (Shalahuddin dan Rosa, 2008).

## 2. Aktor

Aktor adalah orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor. Simbol actor pada *use case diagram* dapat dilihat pada Gambar 2.3.



**Gambar 2.3** Aktor (Shalahuddin dan Rosa, 2008).

## 3. Asosiasi atau *Association*

4. Asosiasi adalah komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor. Simbol asosiasi dapat dilihat pada Gambar 2.4.

---

**Gambar 2.4** Asosiasi (Shalahuddin dan Rosa, 2008)

## 5. Ekstensi atau *Extend*

Ekstensi adalah relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu; mirip dengan prinsip *inheritance* pada pemrograman berorientasi objek; biasanya *use case* tambahan

memiliki nama depan yang sama dengan *use case* yang ditambahkan (Shalahuddin dan Rosa, 2008).

#### 6. Generalisasi atau *Generalization*

Generalisasi adalah Hubungan generalisasi dan spesialisasi (umum - khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya (Shalahuddin dan Rosa, 2008).

#### 7. *Include*

*Include* adalah *use case* yang tambahan akan selalu melakukan pengecekan apakah *use case* yang ditambahkan telah dijalankan sebelum *use case* tambahan dijalankan (Shalahuddin dan Rosa, 2008).

## 2. Activity Diagram

*Activity diagram* menurut Martin Fowler (2005) adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja. Dalam beberapa hal, *activity diagram* memainkan peran mirip diagram alir, tetapi perbedaan prinsip antara notasi diagram alir adalah *activity diagram* mendukung *behavior paralel*. *Node* pada sebuah *activity diagram* disebut sebagai *action*, sehingga diagram tersebut menampilkan sebuah *activity* yang tersusun dari *action*.

## 3. Sequence Diagram

*Sequence diagram* menurut Munawar (2005) adalah grafik dua dimensi dimana obyek ditunjukkan dalam dimensi horizontal, sedangkan *lifeline* ditunjukkan dalam dimensi vertikal.

#### 4. Class Diagram

*Class diagram* menurut Munawar (2005) merupakan himpunan dari objek-objek yang sejenis. Sebuah objek memiliki keadaan sesaat (*state*) dan perilaku (*behavior*). *State* sebuah objek adalah kondisi objek tersebut yang dinyatakan dalam *attribute/properties*. Sedangkan perilaku suatu objek mendefinisikan bagaimana sebuah objek bertindak/beraksi dan memberikan reaksi.

#### 2.17 Metode Pengujian Sistem

Pengujian *Blackbox* merupakan pengujian yang berdasar kepada spesifikasi kebutuhan sistem dan tidak perlu memahami struktur pemrograman. Pengujian ini murni dilakukan dengan sudut pandang penguji yaitu *end user*. Pengujian *blackbox* memainkan peran penting dalam pengujian perangkat lunak dalam membantu proses validasi seluruh fungsi dari sistem yang dibangun. Pengujian ini dianggap selesai apabila semua kebutuhan *user* atau pengguna telah dipenuhi. Walaupun jika nantinya ada kebutuhan *user* yang belum lengkap atau muncul kebutuhan yang tak terduga, maka hal tersebut dapat didefinisikan ulang kemudian. Pengujian ini hanya berdasar pada perspektif *user*. Hal yang paling penting dalam pengujian *blackbox* ini adalah pengujian ini menangani *input* yang valid maupun tidak sesuai perspektif *user* (Nidhra dan Jagurthi, 2012).

Pengujian *Blackbox* memperlakukan perangkat lunak yang akan diuji sebagai sebuah “Kotak Hitam”, tanpa diketahui tentang struktur kerja internal dan pengujian ini hanya memeriksa aspek fundamental dari sebuah



sistem. Saat melakukan pengujian *blackbox*, penguji harus mengetahui arsitektur sistem namun tidak harus memiliki akses untuk memperhatikan *source code* sistem. Beberapa keunggulan dan kerugian menggunakan pengujian *blackbox* adalah sebagai berikut (Khan dan Farmeena, 2012).

#### A. Keuntungan

1. Sangat efisien bagi aplikasi yang memiliki *source code* yang banyak.
2. Persepsi penguji sangat sederhana.
3. Perbedaan perspektif pengguna dan pengembang sangat mudah dilihat (penguji dan pengembang tidak saling berhubungan).
4. Uji coba pengembangan menjadi lebih cepat.

#### B. Kerugian

1. Hanya sebagian dari skenario tes yang dapat dijalankan, selebihnya hanya dapat mengulas sistem secara terbatas.
2. Sulit untuk menggambarkan proses uji coba tanpa melihat spesifikasi yang jelas (Khan dan Farmeena, 2012).

*Equivalence partitioning* didasarkan pada asumsi bahwa input dan output domain program ini dapat dibagi menjadi beberapa kelas (valid dan tidak valid) sehingga semua kasus dalam kasus uji fungsi partisi tunggal akan sesuai atau menunjukkan perilaku yang sesuai. Partisi dilakukan sehingga program menjalankan fungsi yang sama pada setiap nilai masukan yang tergabung ke dalam kelas kesetaraan. Uji kasus dirancang untuk menguji partisi masukan atau domain output. Kelas kesetaraan ditentukan dengan memeriksa dan menganalisa berbagai masukan data. Hanya satu kasus uji dari setiap partisi yang diperlukan untuk mengurangi jumlah kasus uji yang

diperlukan untuk mencapai cakupan fungsional. Keberhasilan pendekatan ini tergantung pada tester yang mampu mengidentifikasi partisi dari input dan output (Nidhra dan Jagurthi, 2012).



Tabel 3.1 Waktu Kegiatan Penelitian (lanjutan)

No	Kegiatan	Waktu Pelaksanaan																	
		Tahun 2018										Tahun 2019							
		Bulan																	
		3	4	5	6	7	8	9	10	11	12	1	2	3	4	5			
7	<i>Coding</i>																		
8	<i>Testing</i>																		
9	<i>Implementasi dan maintenance</i>																		

### 3.2 Alat Pendukung Penelitian

Penelitian ini dilakukan dengan menggunakan beberapa alat untuk mendukung dan menunjang pelaksanaan penelitian. Alat yang digunakan merupakan perangkat keras (*hardware*) dan perangkat lunak (*software*). Alat tersebut di antaranya adalah sebagai berikut.

- a. *Notebook* dengan spesifikasi.
  - Prosesor Intel Core i3-3217U 1,80 Ghz
  - RAM 4 GB
  - *Harddisk* 500 GB
  - *Display* 14`` HD LED
- b. *Smartphone Android* dengan spesifikasi.
  - Android OS 6.0 (Marshmallow)
  - CPU Qualcomm Snapdragon 800, Quad-core 2.5 GHz
  - RAM 2GB
  - Kapasitas penyimpanan internal 16GB
- c. Perangkat Lunak (*Software*) yang digunakan.

- Sistem Operasi Windows 10 (64Bit)
- Bahasa Pemrograman:PHP, HTML, Java dan XML
- Android Studio
- Notepad++
- Baslamiq Mockup 3
- Star UML
- Microsoft Office 2013

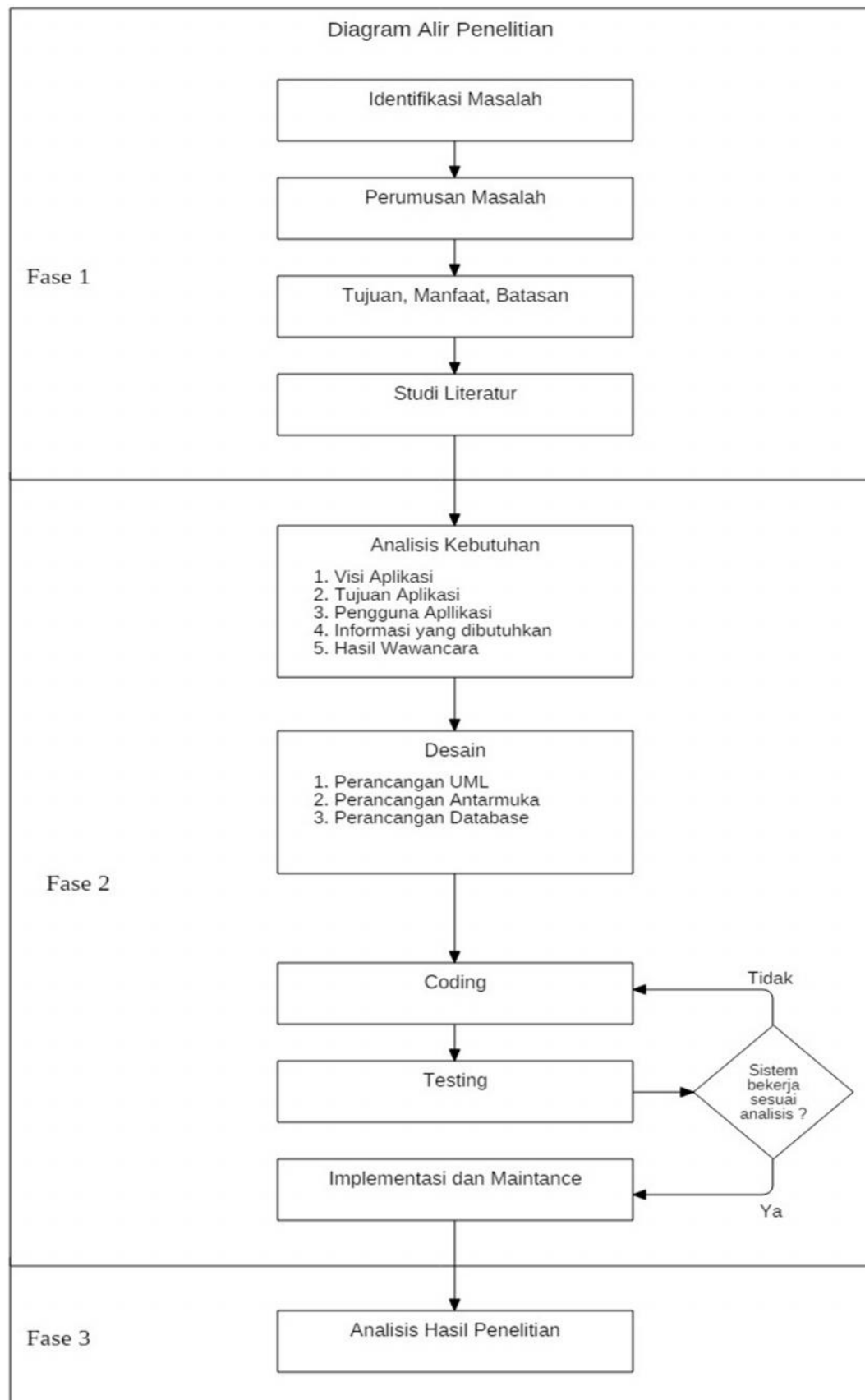
### **3.3 Metodologi Penelitian**

#### **3.3.1 Diagram Alir Metodologi Penelitian**

Penelitian dilakukan berdasarkan diagram alir metodologi penelitian yang terdapat pada Gambar 3.1. Penjelasan dari diagram alir penelitian metodologi penelitian pada Gambar 3.1 adalah sebagai berikut.

##### **a. Fase 1**

1. Tahap pertama pada penelitian ini adalah mengidentifikasi masalah, yaitu menentukan masalah yang diselesaikan dalam penelitian.
2. Tahap kedua adalah perumusan masalah, yaitu pertanyaan penelitian yang membutuhkan jawaban melalui pengumpulan data yang umumnya disusun dalam bentuk kalimat tanya.
3. Tahap ketiga adalah menentukan tujuan, manfaat, dan batasan masalah, yaitu penentuan tujuan dan manfaat dari penelitian yang penulis lakukan. Serta batasan digunakan untuk membatasi pembahasan dan ruang lingkup penelitian.



**Gambar 3. 1** Diagram Alir Penelitian

4. Tahap keempat adalah studi literature, yaitu memperoleh teori-teori dan hasil-hasil penelitian terlebih dahulu yang dijadikan sebagai landasan penelitian.

**b. Fase 2**

5. Tahap kelima dalam fase kedua yaitu pengembangan sistem. Pada penelitian ini, metode yang digunakan adalah *Waterfall*. Tahap pertama yang dilakukan dalam pengembangan sistem ini adalah analisis kebutuhan yang diperlukan pada sistem.
6. Tahap keenam adalah design, yaitu perancangan sistem dan perancangan antarmuka yang menggunakan *Unified Modelling Language* (UML) serta perancangan basis data dari sistem yang dibangun. Pada tahap ini diketahui semua entitas luar, input dan output yang terlibat dalam sistem serta *usecase*, *class* diagram, *activity* diagram, *sequence* diagram yang digunakan dalam analisis sistem. Selanjutnya dilakukan perancangan antarmuka (*interface*), perancangan antarmuka ini dilakukan untuk merancang tata letak sistem sesuai dengan analisis kebutuhan sistem.
7. Tahap ke-tujuh adalah *coding*, yaitu tahap pembuatan program aplikasi.
8. Tahap kedelapan adalah *testing*, yaitu tahap pengujian program. Program yang telah selesai dibuat kemudian diuji dengan metode pengujian perangkat lunak *black box*. Apabila program aplikasi tidak berjalan sesuai dengan analisis maka kembali ke tahap *coding*. Namun, apabila telah berjalan maka ke tahap selanjutnya
9. Tahap kesembilan adalah penerapan aplikasi dan *maintenance*, yaitu penyerahan aplikasi kepada pengguna dan pemeliharaan sistem.

### c. Fase 3

10. Tahap kesepuluh adalah analisis hasil penelitian, yaitu penjelasan apakah suatu penelitian telah berhasil memecahkan permasalahan yang telah dideskripsikan di awal.

### 3.3.2 Metode Pengembangan Aplikasi

Penelitian dilakukan berdasarkan metode pengembangan sistem yang dipilih yaitu *Waterfall* yang terdiri dari lima tahapan yang dikerjakan secara berurutan dan sistematis. Tahapan-tahapan terdiri dari Analisis Kebutuhan (*Requirements Developments*), Desain (*Software Design*), Penerapan desain dan penulisan kode program (*Implementation and Coding*), Penerapan aplikasi dan pemeliharaan (*Operational and Maintenance*).

#### A. Analisis Kebutuhan (*Requirements Developments*)

Analisis kebutuhan merupakan tahap awal penelitian. Tahap ini dimulai dengan mengumpulkan data-data yang mendukung dalam penelitian. Dalam mengembangkan Aplikasi Pengelolaan Transaksi dan Manajemen Menu di Café Black Taste Berbasis Web Dan Mobile telah didapatkan informasi sebagai berikut :

1. Visi Aplikasi adalah memajemen seluruh proses pekerjaan yang ada di cafe agar semuanya lebih terstruktur dan berjalan dengan baik.



2. Tujuan Aplikasi adalah mengatur proses pemesanan menu, mengatur proses transaksi pembayaran, dan mengatur menu dan bahan.
3. Pengguna dari aplikasi adalah pemilik kafe dan pegawai kafe.
4. Informasi yang dibutuhkan dalam aplikasi ini adalah informasi data bahan makanan dan minuman, data menu makanan dan minuman, data nomor meja dan data karyawan.
5. Hasil wawancara yang didapat dari pemilik kafe, sering adanya kesalahan penghitungan harga, sering adanya nota pembayaran yang hilang, pemesanan menu menjadi rumit ketika lokasi ramai pengunjung, terkadang adanya ketidaksesuaian antara uang pendapatan dengan stok bahan yang tersisa. Maka dari itu pemilik menginginkan adanya sistem yang dapat memanajemen hal-hal tersebut.

Dengan analisis dan data yang telah didapatkan ini dapat diketahui *user requirment* yang dibutuhkan dalam pengembangan aplikasi.

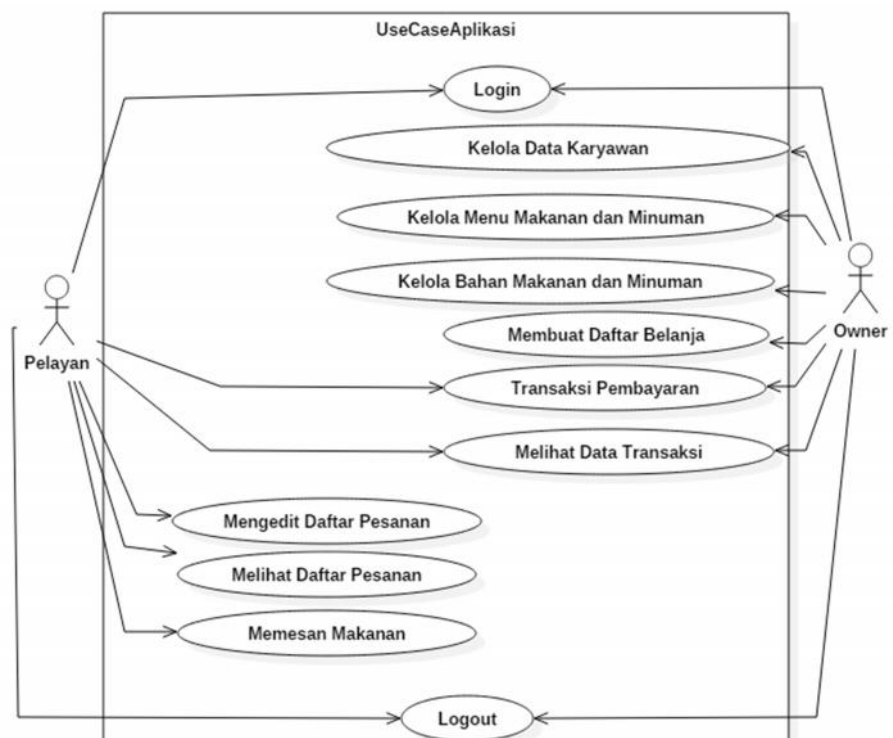
## **B. Desain (*Software Design*)**

Tahapan ini merupakan tahapan merencanakan rancangan *interface* dan fungsional aplikasi yang akan digunakan didalam sistem. Perancangan sistem yang digunakan adalah *Unified Modeling Language* (UML). *Unified Modeling Language* (UML) menggunakan notasi standar untuk menjelaskan secara visual mengenai elemen-elemen permodelan. Tahapan rancangan untuk aplikasi ini dibuat ke dalam bentuk desain yang terdiri dari

perancangan desain *Use Case Diagram*, desain *Activity Diagram*, desain *Sequence Diagram*, *Desain Class Diagram*, dan desain *User Interface*.

### 1. *Use case Diagram*

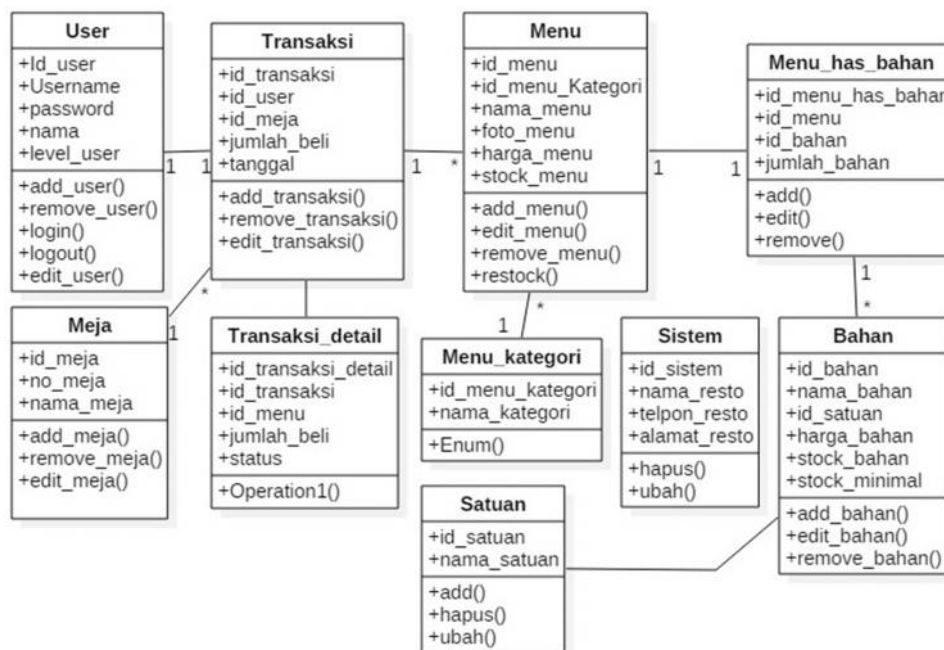
*Use case Diagram* di bawah ini menggambarkan sistem dari sudut pandang Pengguna dan Admin. *Usecase diagram* ini menjelaskan bagaimana Aplikasi Pengelolaan Transaksi dan Manajemen Menu di Café Black Taste berbasis web dan mobile ini berjalan. Pada aplikasi ini memiliki 8 interaksi, 3 interaksi pelayan, 4 interaksi kasir, dan 7 interaksi untuk admin. Usecase aplikasi Aplikasi Pengelolaan Transaksi dan Manajemen Menu di Café Black Taste berbasis web dan mobile dapat dilihat pada Gambar 3.2 .



**Gambar 3.2** *Usecase Diagram*

### 3. Class Diagram

*Class diagram* adalah sebuah *class* yang menggambarkan struktur dan penjelasan *class*, paket, dan objek serta hubungan satu sama lain. Pada pembuatan sistem ini *class diagram* dapat dilihat pada Gambar 3.3.



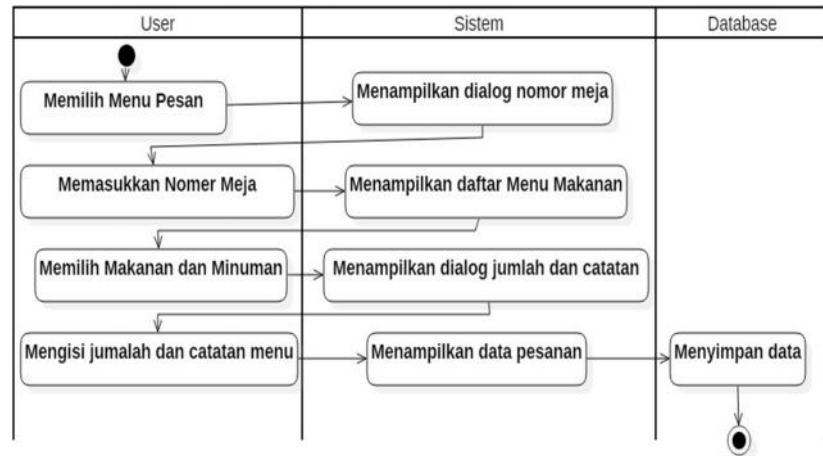
Gambar 3.3 Class Diagram Sistem

### 2. Activity Diagram, Sequence Diagram, dan Perancangan Tampilan

*Activity Diagram* merupakan diagram yang menggambarkan berbagai aliran aktivitas yang terjadi pada sistem, yang meliputi bagaimana sebuah proses dapat terjadi, kemungkinan-kemungkinan yang bisa terjadi, serta bagaimana akhir dari proses tersebut. *Sequence Diagram* merupakan suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan

komunikasi di antara obyek-obyek tersebut. Berikut ini adalah *Activity Diagram*, *Sequence Diagram*, dan perancangan interface dari Aplikasi Pengelolaan Transaksi dan Manajemen Menu di Café Black Taste berbasis web dan mobile.

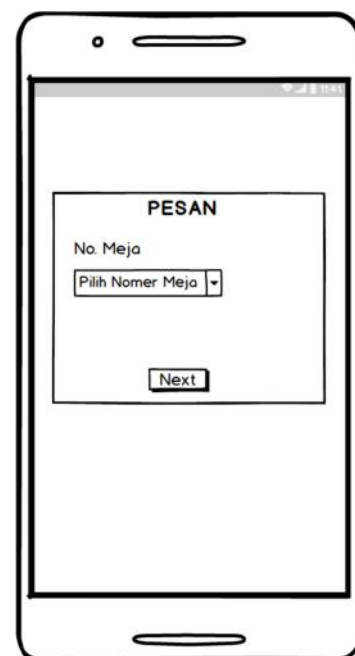
#### A. Pesan Menu



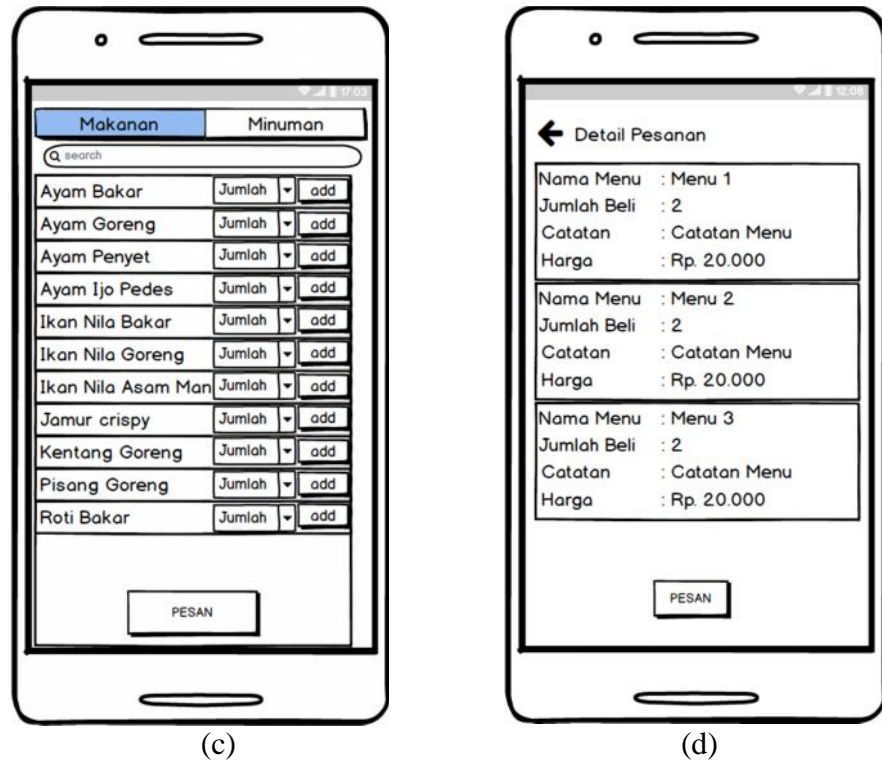
**Gambar 3.4** *Activity Diagram* Pesan Menu



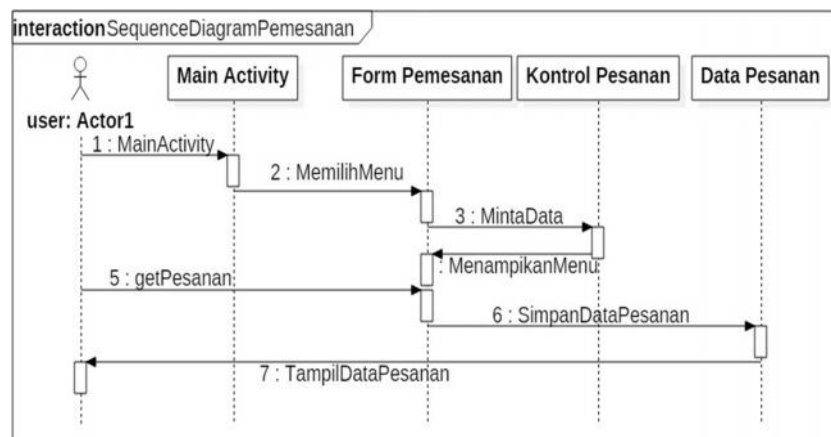
(a)



(b)



**Gambar. 3.5** Rancangan *Interface* (a) Halaman Utama, (b) Nomor Meja, (c) Daftar Menu, (d) Konfirmasi dan Rincian Pembelian

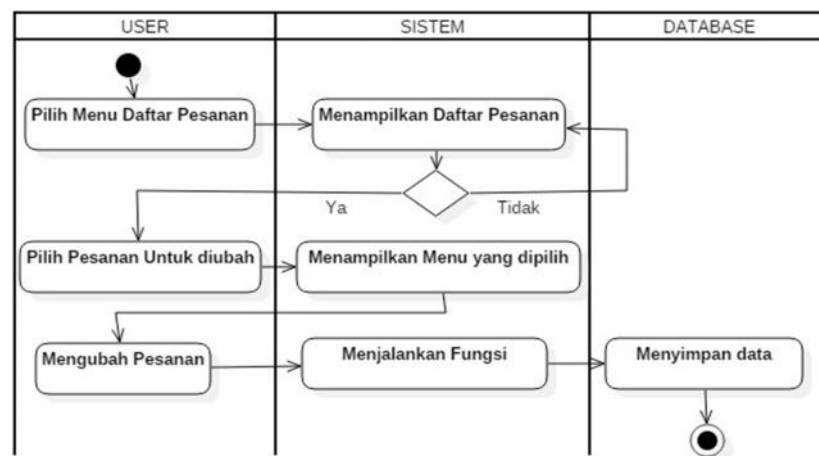


**Gambar 3.6** *Sequence Diagram* Pesan Menu

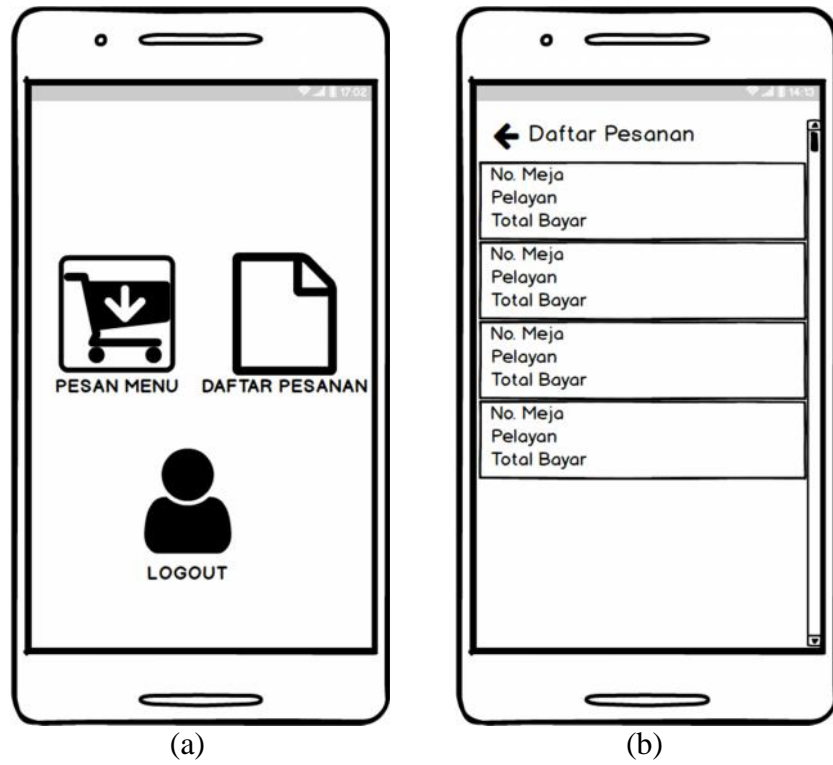
*Activity* dan *sequence diagram* pada Gambar 3.4 dan Gambar 3.6 menggambarkan proses alur untuk memesan menu melalui aplikasi android yang dilakukan oleh pelayan. Pertama

pengguna memilih menu pesan pada halaman utama aplikasi yang terlihat pada Gambar 3.5 (a), lalu sistem akan menampilkan *form* untuk memasukkan nomor meja yang terlihat pada Gambar 3.5 (b) lalu sistem menampilkan daftar makanan dan minuman yang terlihat pada Gambar 3.5 (c), pengguna memilih makanan dan minuman yang akan dipesan sesuai dengan pesanan konsumen dan selanjutnya memilih tombol pesan, selanjutnya sistem menampilkan seluruh rincian pesanan yang terlihat pada Gambar 3.5 (d), data pesanan selesai dan tersimpan di *database*.

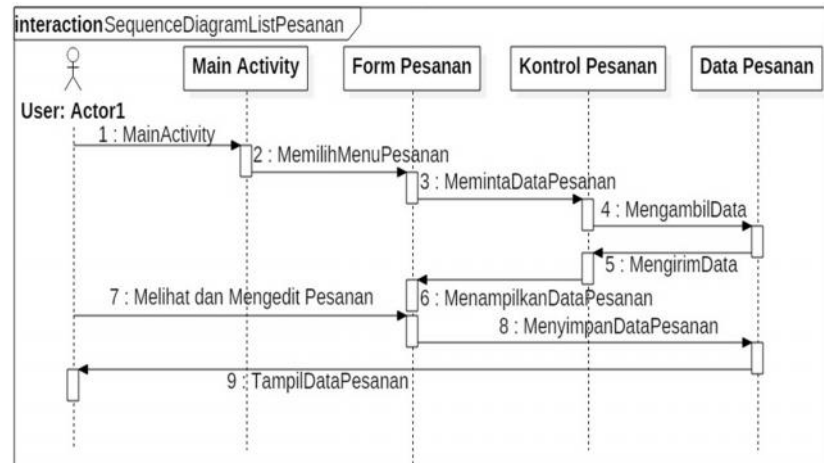
#### B. Melihat dan Mengubah Daftar Pesanan



**Gambar 3.7** Activity Diagram Melihat Daftar Pesanan



**Gambar 3.8** Rancangan *Interface* (a) Halaman Utama, (b) Daftar Pesanan, (c) Rincian Pesanan.

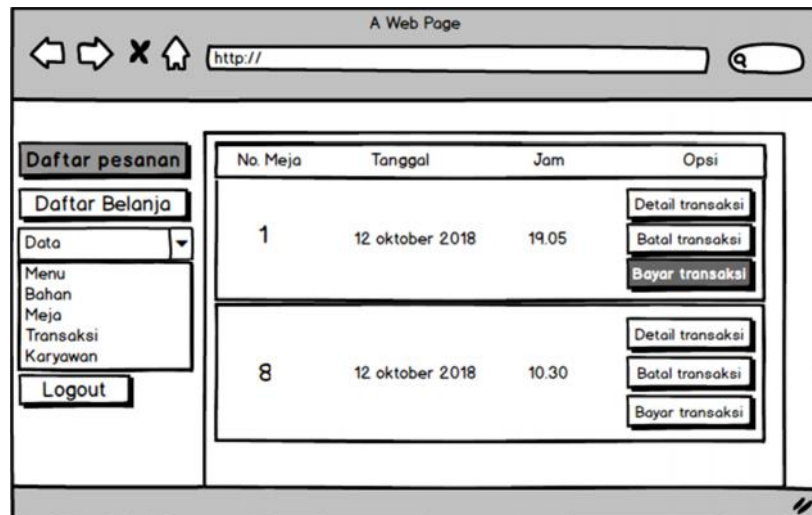


**Gambar 3.9** *Sequence diagram* melihat daftar pesanan

*Activity* dan *sequence diagram* pada Gambar 3.7 dan Gambar 3.9 menggambarkan proses alur untuk melihat daftar pesanan dan mengedit daftar pesanan seperti melakukan tambah dan hapus makanan atau minuman pada suatu pesanan melalui aplikasi android yang dilakukan oleh pelayan. Pertama Pengguna memilih menu daftar Pesanan pada halaman utama aplikasi yang terlihat pada Gambar 3.8 (a), lalu sistem menampilkan daftar pesanan yang terlihat pada Gambar 3.8 (b), pada setiap pesanan dapat memiliki fungsi, jika pesanan dipilih maka sistem akan menampilkan rincian pesanan yang terlihat pada Gambar 3.8 (c). Pada tampilan rincian pesanan ada 3 buah fragmen halaman yaitu fragmen disiapkan, dimasak dan disajikan. Pada halaman disiapkan pengguna dapat menambahkan makanan dan minuman atau menghapus makanan dan minuman pada salah satu pesanan. Pesanan yang sedang dalam proses dimasak, pengguna dapat mengganti

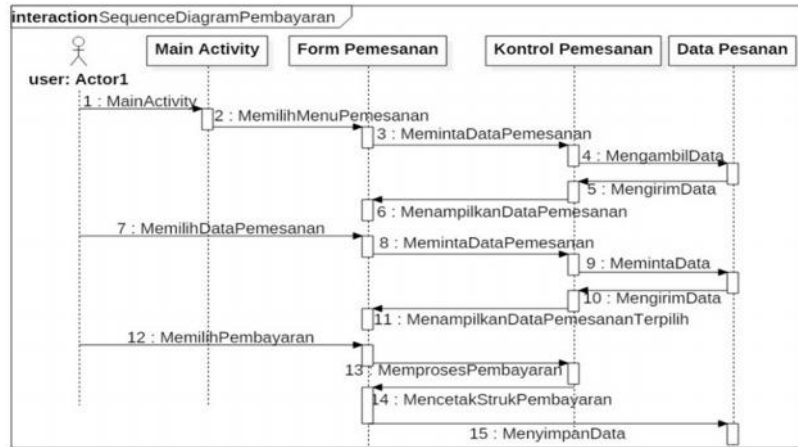






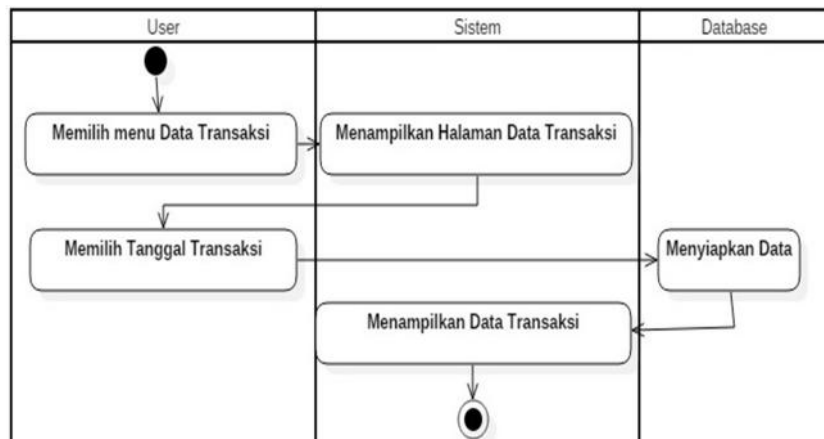
**Gambar 3.11** Rancangan *Interface* Cetak Pembayaran

*Activity* dan *sequence diagram* pada Gambar 3.10 dan Gambar 3.12 menggambarkan proses alur untuk melakukan transaksi pembayaran yang dapat dilakukan oleh kasir dan admin melalui web. Pertama pengguna memilih menu daftar pesanan yang ada pada halaman utama web, kemudian sistem akan menampilkan daftar pesanan yang belum dibayar dapat dilihat pada Gambar 3.11. Pilih salah satu pesanan yang akan dibayar, lalu sistem akan menampilkan rincian pesanan yang akan dibayar oleh pelanggan.

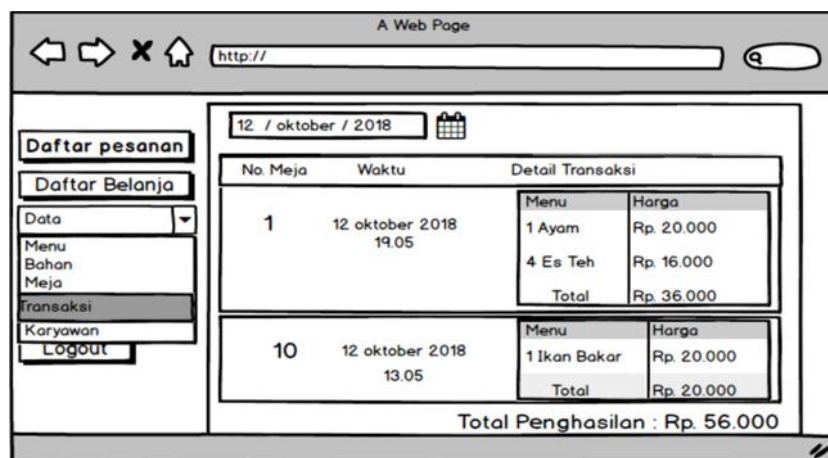


Gambar 3.12 Sequence Diagram Transaksi Pembayaran

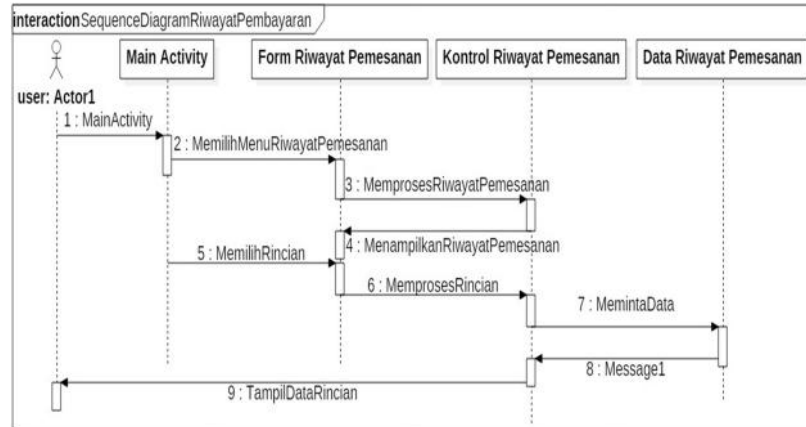
D. Data Transaksi



Gambar 3.13 Activity Diagram Data Transaksi



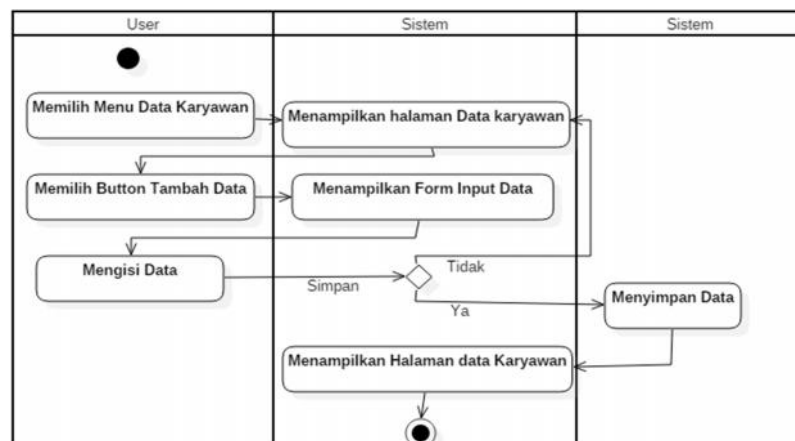
Gambar 3.14 Rancangan Interface Data Transaksi



**Gambar 3.15** *Sequence Diagram* Riwayat Pembelian

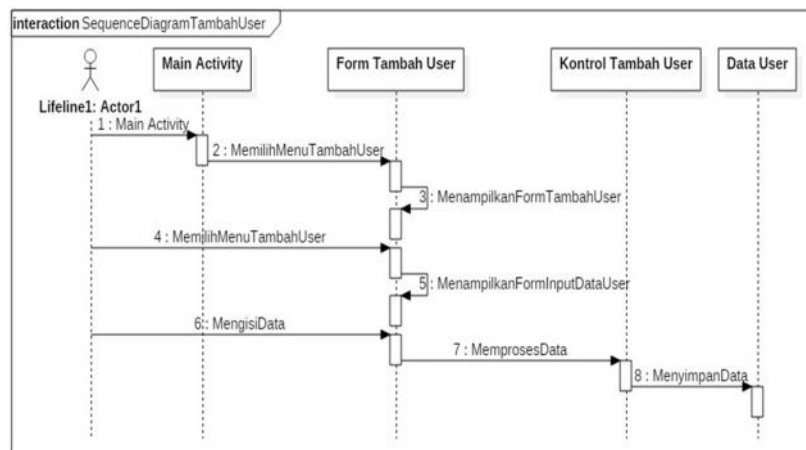
*Activity* dan *sequence diagram* pada Gambar 3.13 dan Gambar 3.15 menggambarkan proses alur untuk melihat data transaksi yang dapat dilakukan oleh kasir dan admin melalui web. Pertama pengguna memilih menu data transaksi, lalu sistem akan menampilkan tabel data transaksi. Pengguna dapat mengelompokkan data sesuai dengan tanggal seperti pada Gambar 3.14.

#### E. Menambahkan Data *Login User*

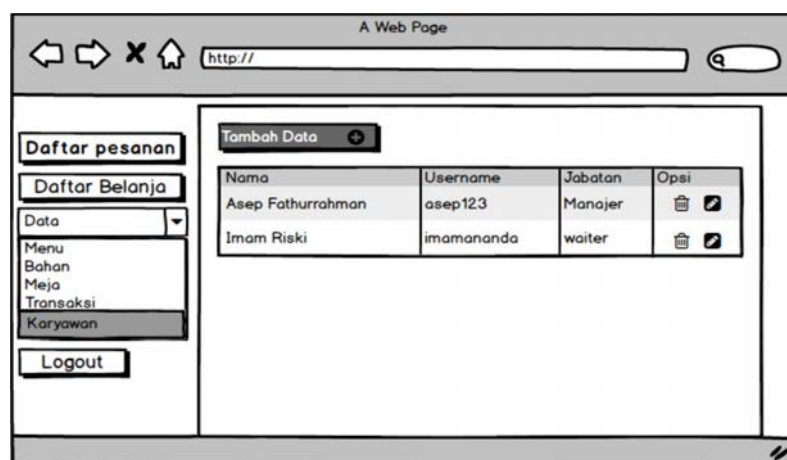


**Gambar 3.16** *Activity Diagram* Tambah Data *Login User*

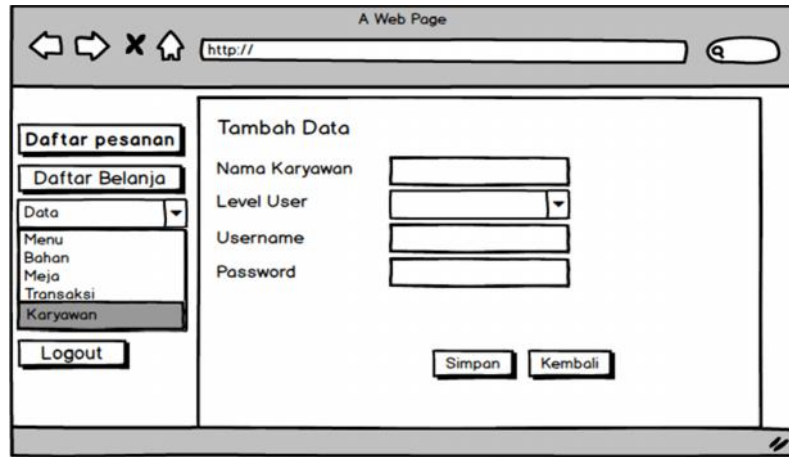
*Activity* dan *sequence diagram* pada Gambar 3.16 dan Gambar 3.17 menggambarkan proses alur untuk menambahkan *login* user/karyawan ke aplikasi yang dapat dilakukan oleh admin melalui web. Pertama pengguna memilih menu data karyawan pada halaman utama, lalu sistem akan menampilkan halaman seperti pada Gambar 3.18 (a). Tahap selanjutnya pengguna memilih *button* tambah data lalu sistem akan menampilkan *form input* seperti pada Gambar 3.18 (b).



**Gambar 3.17** *Sequence Diagram Tambah Login User*



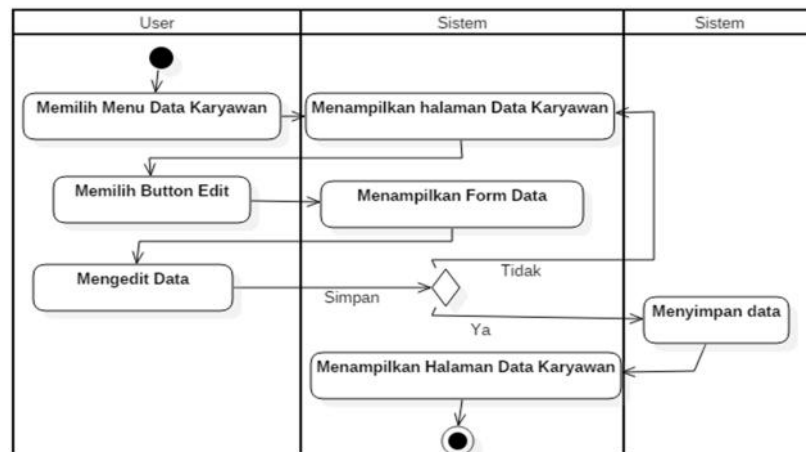
(a)



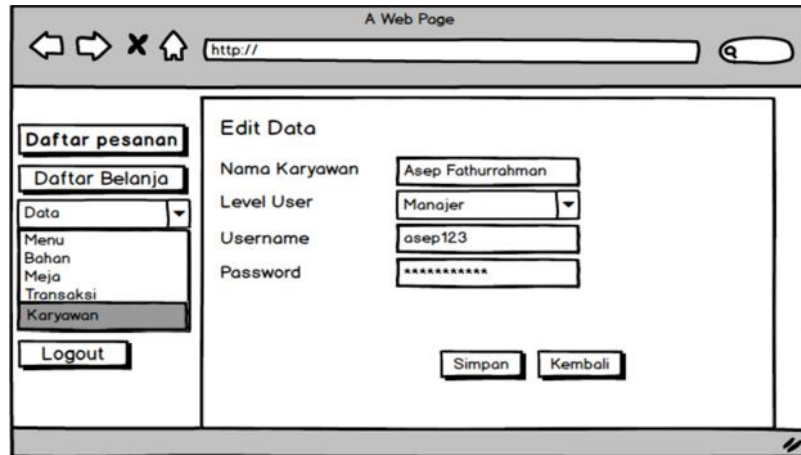
(b)

**Gambar 3.18** Rancangan *Interface* (a) Tambah Data *User*, (b) Form Tambah Data *User*

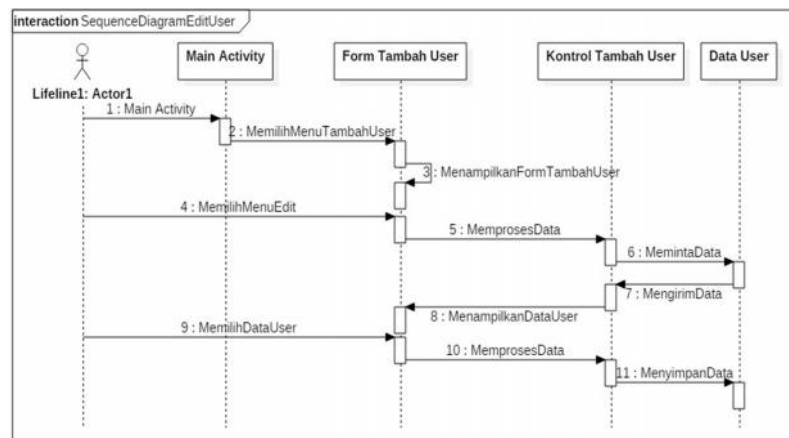
#### F. Mengubah Data *Login User*



**Gambar 3.19** Activity Diagram Edit data Login



**Gambar 3.20** Rancangan *Interface* Edit Data User

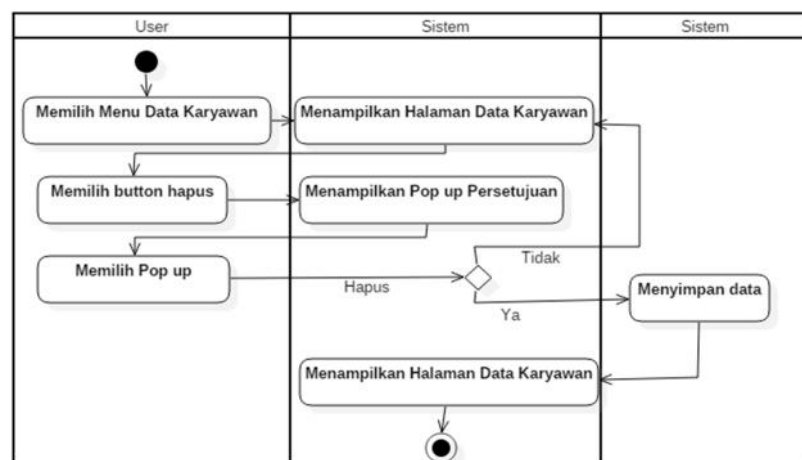


**Gambar 3.21** *Sequence Diagram* Edit data Login

*Activity* dan *sequence diagram* pada Gambar 3.19 dan 3.21 menggambarkan proses alur untuk mengubah *login user/karyawan* ke aplikasi yang dapat dilakukan oleh admin melalui web. Pertama pengguna memilih menu data karyawan pada halaman utama, lalu sistem akan menampilkan seperti pada Gambar 3.20. Tahap selanjutnya pengguna memilih tombol Edit, lalu sistem akan menampilkan *form* data yang akan diubah oleh pengguna yang terlihat pada Gambar 3.20.

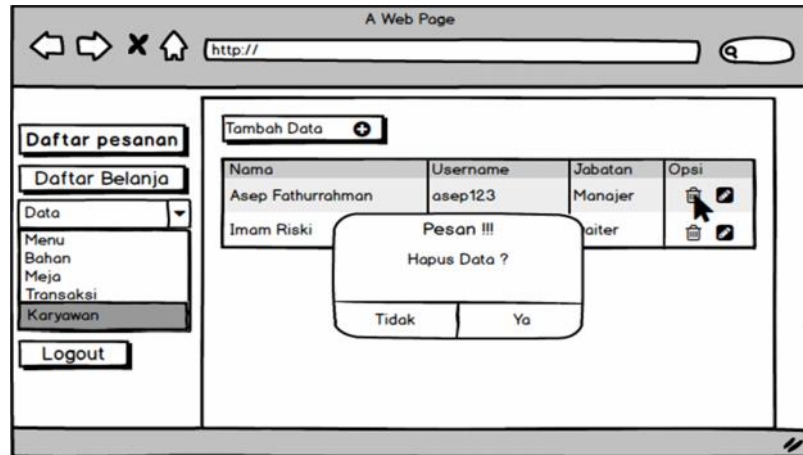
### G. Menghapus Data *Login User*

*Activity* dan *sequence diagram* pada Gambar 3.22 dan Gambar 3.24 menggambarkan proses alur untuk menghapus *login user*/karyawan ke aplikasi yang dapat dilakukan oleh admin melalui web. Pertama pengguna memilih menu data karyawan pada halaman utama, lalu sistem akan menampilkan halaman seperti yang terlihat pada Gambar 3.23. Tahap selanjutnya pengguna memilih tombol Hapus selanjutnya akan muncul *popup* untuk menyakinkan pengguna akan menghapus data atau tidak seperti pada Gambar 3.23.

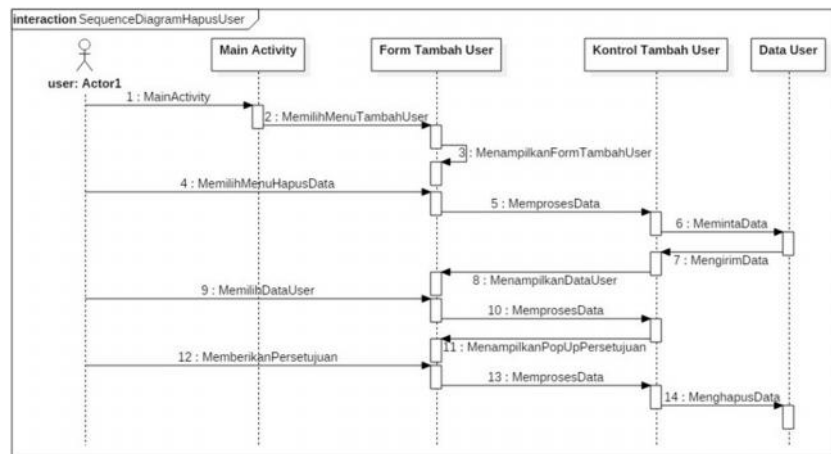


**Gambar 3.22** *Activity Diagram* Hapus data *Login User*





**Gambar 3.23** Rancangan *Interface* Hapus Data *User Login User*

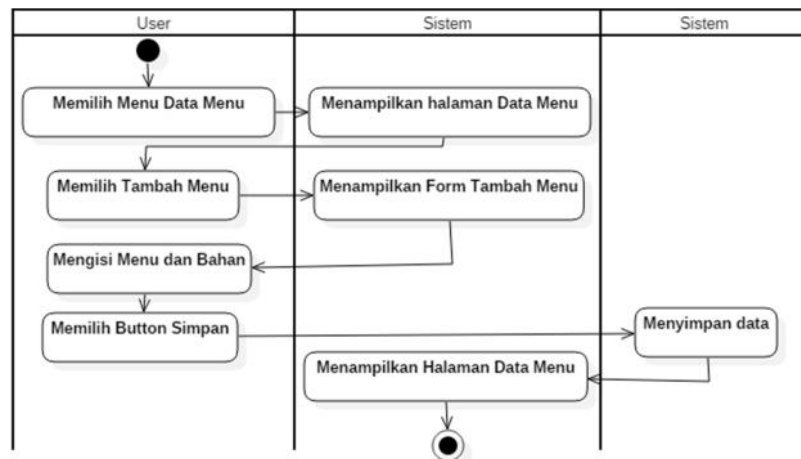


**Gambar 3.24** *Sequence Diagram* Hapus data *Login User*

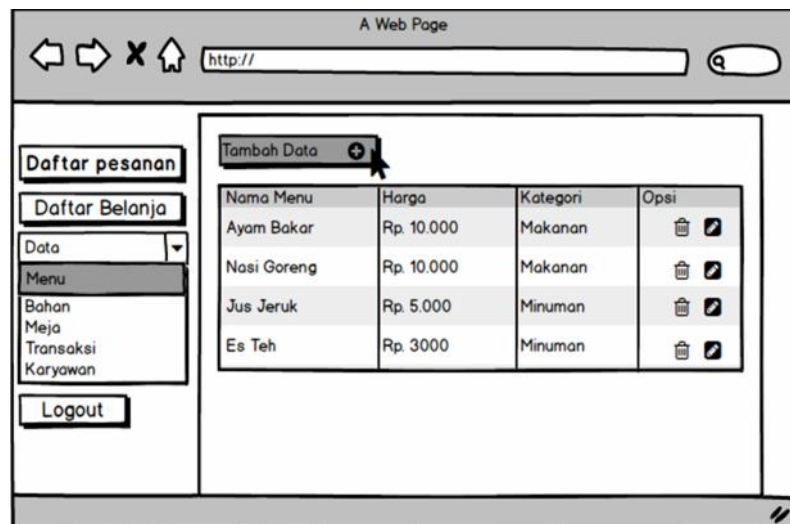
## H. Menambahkan Menu

*Activity* dan *sequence diagram* pada Gambar 3.25 dan Gambar 3.27 menggambarkan proses alur untuk menambahkan menu baru yang dapat dilakukan oleh admin melalui web. Pertama pengguna memilih menu Data Menu pada halaman utama, lalu sistem akan menampilkan halaman seperti yang terlihat pada Gambar 3.26 (a). Tahap selanjutnya pengguna memilih tombol Tambah Menu maka sistem akan menampilkan *form input*

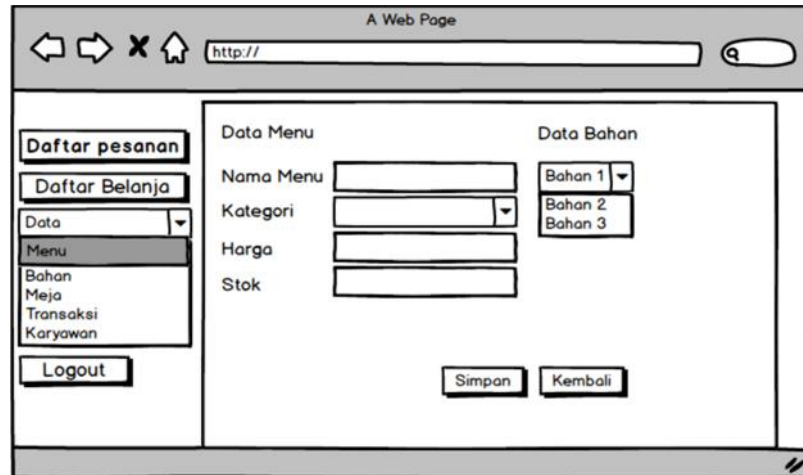
seperti pada Gambar 3.26 (b), *form input* berfungsi untuk menambahkan menu dan bahan yang baru yang harus di isi oleh pengguna. Setelah pengguna mengisi dan menyimpan data, menu baru akan tampil di halaman Data Menu.



**Gambar 3.25** Activity Diagram Tambah Menu

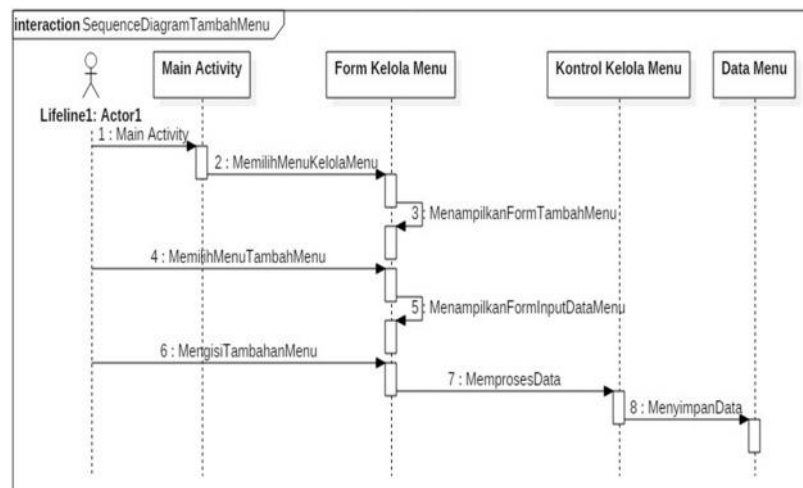


(a)



(b)

**Gambar 3.26** Perancangan *Interface* (a) Data Menu, (b) Tambah Data Menu

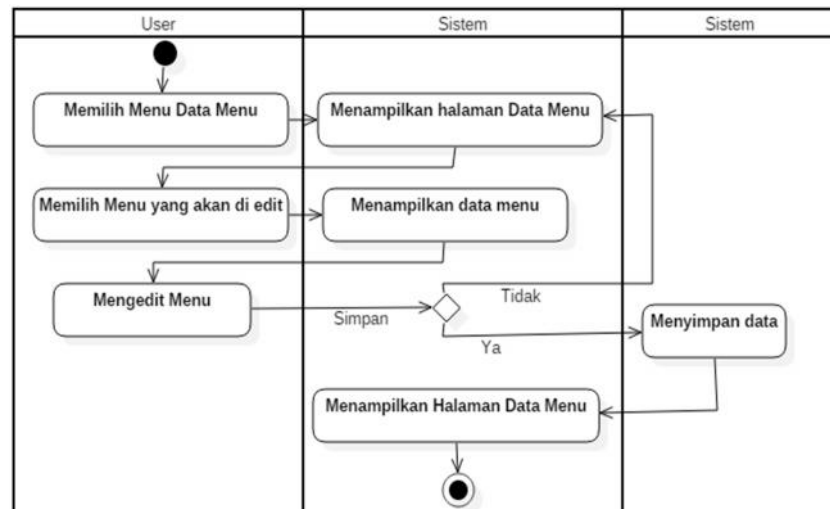


**Gambar 3.27** *Sequence Diagram* Tambah Data Menu

## I. Mengubah Menu

*Activity* dan *sequence diagram* pada Gambar 3.28 dan Gambar 3.30 menggambarkan proses alur untuk meng-*edit* menu yang telah ada yang dapat dilakukan oleh admin melalui web. Pertama pengguna memilih menu Data Menu pada halaman utama, lalu sistem akan menampilkan halaman Data Menu

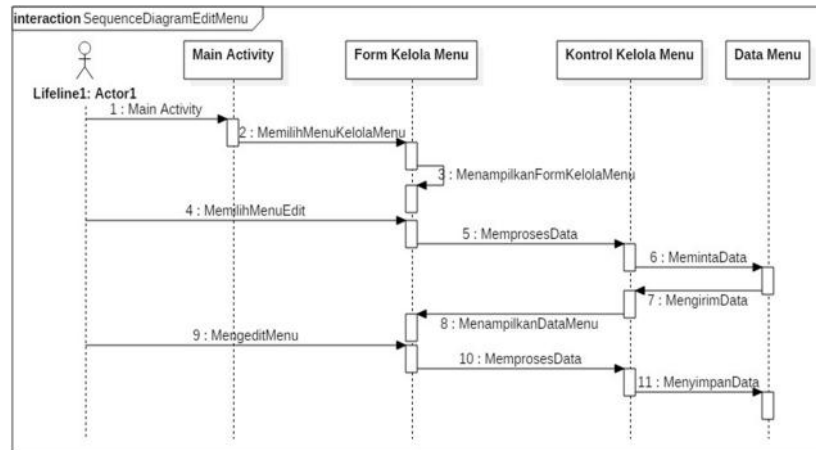
seperti pada Gambar 3.29. Tahap selanjutnya pengguna memilih tombol Edit, maka sistem akan menampilkan *form* data menu yang akan diubah oleh pengguna seperti yang terlihat pada Gambar 3.29. Selanjutnya pengguna mengubah data menu, dan sistem akan menyimpan data pada database.



**Gambar 3.28** Activity Diagram Edit Menu

The screenshot shows a web browser window titled 'A Web Page' with a search bar containing 'http://'. The main content area is divided into three sections. On the left is a sidebar with a 'Logout' button and a menu with options: 'Daftar pesanan', 'Daftar Belanja', 'Data', 'Menu', 'Bahan', 'Meja', 'Transaksi', and 'Karyawan'. The central section is titled 'Edit Data Menu' and contains four input fields: 'Nama Menu' (Ayam Bakar), 'Kategori' (Makanan), 'Harga' (20.000), and 'Stok' (10). The right section is titled 'Data Bahan' and contains three dropdown menus labeled 'Bahan 1', 'Bahan 2', and 'Bahan 3'. At the bottom right of the form are two buttons: 'Simpan' and 'Kembali'.

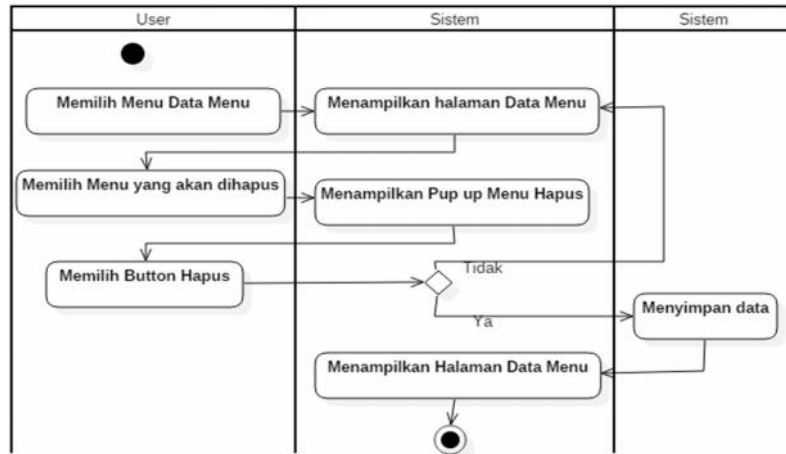
**Gambar 3.29** Perancangan Interface Edit Menu



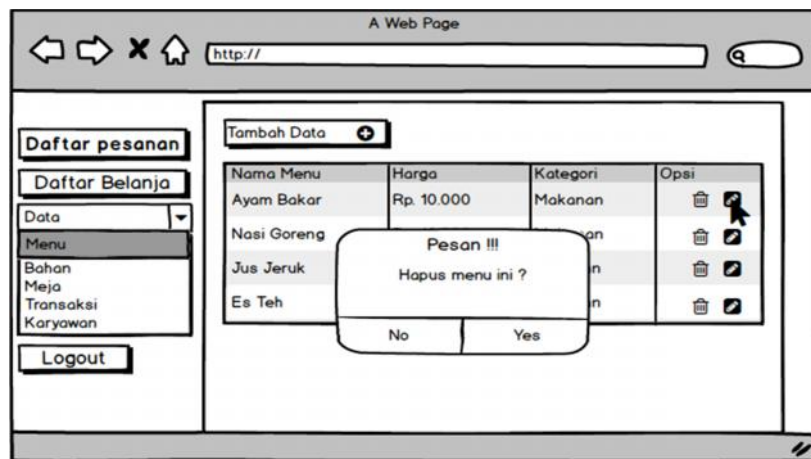
**Gambar 3.30** *Sequence Diagram Edit Menu*

#### J. Menghapus Menu

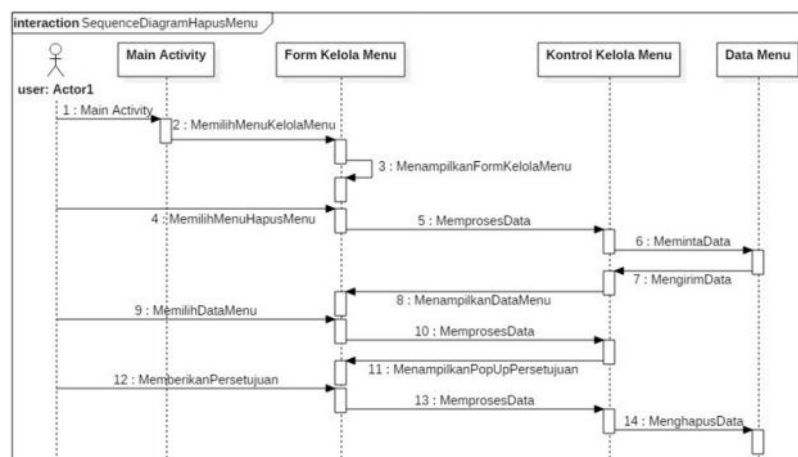
*Activity* dan *Sequence diagram* pada Gambar 3.31 dan Gambar 3.33 menggambarkan proses alur untuk menghapus sebuah menu yang dapat dilakukan oleh admin melalui web. Pertama pengguna memilih menu Data Menu pada halaman utama, lalu sistem akan menampilkan halaman seperti pada Gambar 3.32. Tahap selanjutnya pengguna memilih tombol Hapus, selanjutnya akan muncul *popup* untuk menyakinkan pengguna akan menghapus data menu atau tidak seperti pada Gambar 3.32.



Gambar 3.31 Activity Diagram Hapus Menu



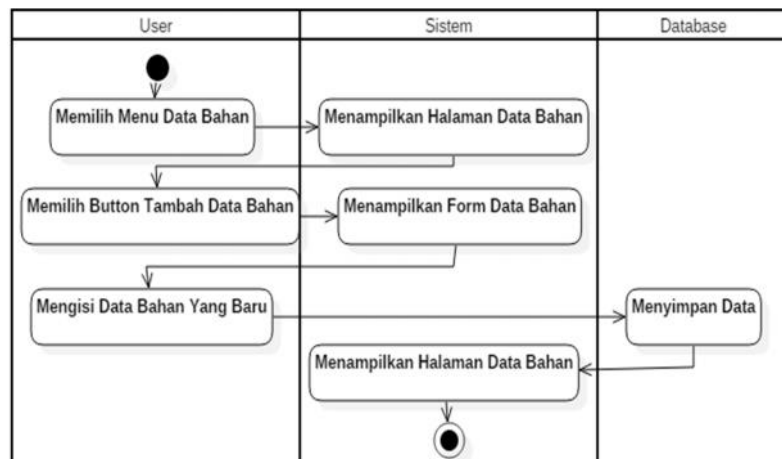
Gambar 3.32 Perancangan Interface Hapus Menu



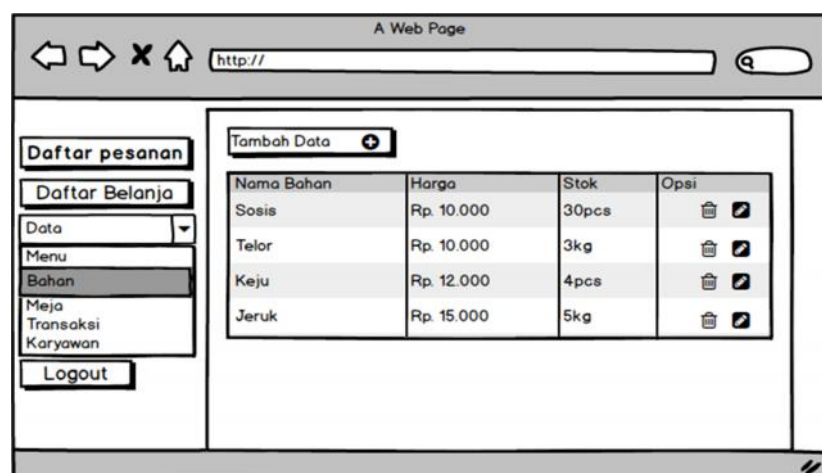
Gambar 3.33 Sequence Diagram Hapus Menu

## K. Bahan Menu Makanan dan Minuman

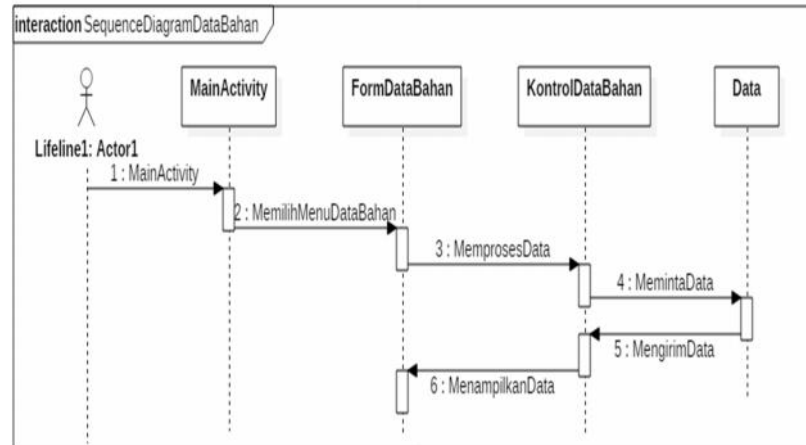
*Activity* dan *sequence diagram* pada Gambar 3.34 dan Gambar 3.36 menggambarkan proses alur untuk mengelola data bahan yang dapat dilakukan oleh admin melalui web. Pertama pengguna memilih menu Data Bahan pada halaman utama, lalu sistem akan menampilkan halaman seperti yang terlihat pada Gambar 3.35.



**Gambar 3.34** *Activity Diagram* Data Bahan



**Gambar 3.35** *Rancangan Interface* Data Bahan

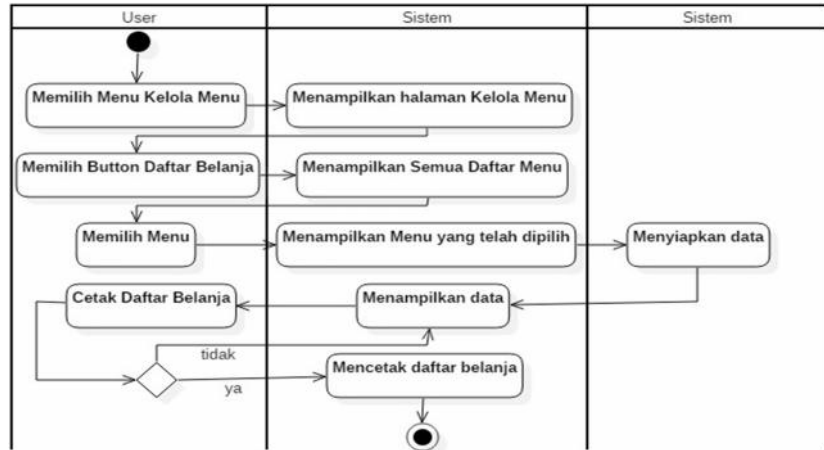


**Gambar 3.36** *Sequence Diagram* Data Bahan

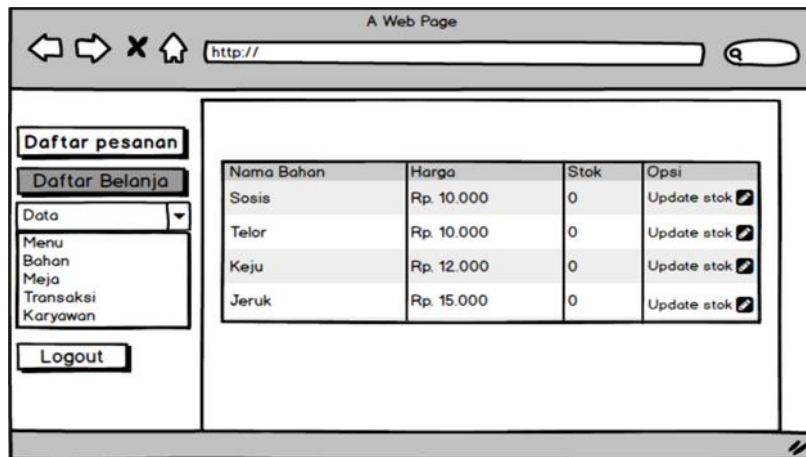
#### L. Daftar Belanja

*Activity* dan *Sequence diagram* pada Gambar 3.37 dan Gambar 3.39 menggambarkan proses alur untuk membuat daftar belanja yang dapat dilakukan oleh admin melalui web. Pertama pengguna memilih menu Daftar Belanja pada halaman utama, lalu sistem akan menampilkan halaman data daftar belanja seperti pada Gambar 2.38. Data yang ada dalam daftar belanja adalah data bahan yang stoknya sudah habis berada di bawah stok minimum yang ditetapkan.

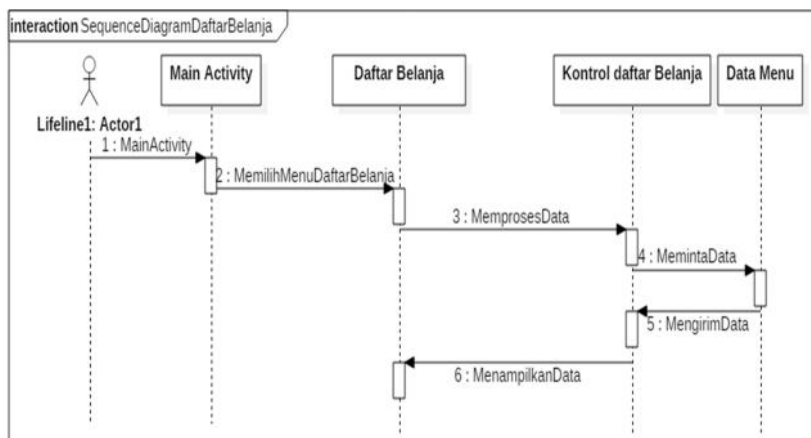




Gambar 3.37 Activity Diagram Daftar Belanja



Gambar 3.38 Perancangan Interface Daftar Belanja



Gambar 3.39 Sequence Diagram Daftar Belanja

### **C. Pengkodean Aplikasi**

Pengkodean merupakan proses menerjemahkan desain yang telah ditetapkan ke dalam bahasa pemrograman yang dapat dimengerti oleh komputer. Pada penelitian Aplikasi Pengelolaan Transaksi dan Manajemen Menu di Café Black Taste berbasis web dan mobile digunakan bahasa pemrograman Java dan sistem berbasis Web dengan bahasa pemrograman PHP.

### **D. Pengujian Aplikasi**

Proses pengujian dilakukan untuk memastikan perangkat lunak dapat bekerja sesuai apa yang telah direncanakan sebelumnya. Selain itu, pada proses ini berfungsi untuk menemukan kesalahan-kesalahan dan memastikan aplikasi ini akan memberikan hasil yang akurat. Pengujian aplikasi juga dilakukan untuk menjawab apakah tujuan dari pembuatan aplikasi tersebut sudah mencapai apa yang diharapkan.

#### **1. Pengujian Fungsional**

Proses pengujian fungsional dilakukan dengan *black-box testing*. *Black-box testing* dilakukan untuk menguji fitur-fitur dari aplikasi yang telah dibangun. Proses pengujian berfokus pada logika internal aplikasi, memastikan bahwa semua pernyataan dari fungsionalitas aplikasi telah diuji, yaitu melakukan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa input

yang dibatasi memberikan respon dan hasil yang aktual sesuai dengan hasil yang dibutuhkan. Semua fungsi-fungsi *software* diujicobakan, agar *software* bebas dari kesalahan, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

Pendekatan kasus uji dalam penelitian ini adalah pengujian *black-box* dengan metode *Equivalence Partitioning* (EP). Pengujian ini dilakukan dengan membagi domain masukan dari program ke dalam kelas-kelas sehingga *test case* dapat diperoleh. EP berdasarkan pada premis masukan dan keluaran dari suatu komponen yang dipartisi ke dalam kelas-kelas, menurut spesifikasi dari komponen tersebut, yang diperlakukan harus sama (ekuivalen) oleh komponen tersebut. Pada pengujian ini harus diyakinkan bahwa masukan yang sama akan menghasilkan respon yang sama pula. Alasan menggunakan metode EP pada pengujian aplikasi ini adalah karena metode ini dapat digunakan untuk mencari kesalahan pada fungsi, dapat mengetahui kesalahan pada *interface* dan kesalahan pada struktur data sehingga dapat mengurangi masalah terhadap nilai masukan. Rancangan daftar pengujian disajikan pada Tabel 3.2.

**Tabel 3.2** Daftar Pengujian *Equivalence Partitoning*

No	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil Yang Diharapkan
1	<i>Login</i>	Masukkan Data <i>Login</i>	Username: Yolanda Password: 12345678 9 Klik tombol Login	Menampilkan pesan kesalahan bahwa password yang dimasukkan salah
			Username: Yolanda Password: 12345678 99 Klik tombol Login	Menampilkan Halaman Home
2	Fungsi Menu Pesan	Pengujian Pemesanan makanan dan minuman	Klik Menu Pesan	Menampilkan daftar makanan dan minuman
			Memilih dan menyimpan makanan yang akan dipesan	Menyimpan pada <i>database</i>
3	<i>User Interface</i>	Pengujian pada tampilan halaman Menu Utama	Klik atau pilih menu Pesan	Menampilkan halaman menu pesan makanan dan minuman

**Tabel 3.2** Daftar Pengujian *Equivalence Partitioning* (lanjutan)

	<i>User Interface</i>	Pengujian pada tampilan halaman Menu Utama	Klik atau pilih menu Daftar Pesanan	Menampilkan halaman menu Daftar Pesanan
			Klik atau pilih menu Data Transaksi	Menampilkan halaman menu Data Transaksi
			Klik atau pilih menu Data Menu	Menampilkan halaman menu Data Menu
4	Fungsi Menu List Pesanan	Pengujian menambahkan pesanan	Klik Menu List Pesanan	Menampilkan daftar pesanan yang belum dibayar
			Memilih salah satu pesanan dan menambahkan pesanan ke pesanan	Menyimpan Perubahan pada <i>database</i>
5	Fungsi Menu Transaksi Pembayaran	Pengujian membayar makanan yang telah dipesan	Klik Menu List Pesanan	Menampilkan daftar pesanan yang belum dibayar
			Memilih tombol bayar pada pesanan yang akan dibayar	Menyimpan Perubahan pada <i>database</i>

## 2. Pengujian Non Fungsional

Pengujian non fungsional pada aplikasi ini akan dilakukan dengan menggunakan kuesioner. Proses pengujian dilakukan secara

objektif dimana program aplikasi diuji secara langsung ke pengguna dengan membuat dan menyebarkan kuesioner kepada responden mengenai kepuasan pengguna terhadap aplikasi dan kesesuaian aplikasi dengan kandungan dari tujuan pembuatan aplikasi yaitu aplikasi yang dapat memudahkan pengguna dalam melaksanakan kegiatan perkuliahan dan juga sebagai aplikasi yang *user friendly*.

Kuesioner dirancang untuk mengumpulkan informasi kinerja keseluruhan dari aplikasi yang diuji serta informasi mengenai komponen spesifik aplikasi. Kuesioner digunakan sebagai proses multi-tahap awal dengan definisi aspek pengujian dan berakhir dengan interpretasi hasil. Pertanyaan kuesioner dibuat pada bab IV bersamaan dengan interpretasi hasil dari kuesioner tersebut. Kuesioner terdiri dari beberapa pertanyaan dan disebar ke beberapa pengguna dari aplikasi ini. Topik dari pertanyaan tersebut di antaranya adalah sebagai berikut.

1. Tampilan/antarmuka.
2. Tingkat kemudahan pengguna.
3. Kelengkapan fitur.
4. Tingkat kepuasan pengguna dan Pencapaian tujuan dari pembuatan aplikasi.

## V. PENUTUP

### 5.1. Kesimpulan

Berdasarkan penelitian yang telah dilakukan dapat disimpulkan beberapa hal sebagai berikut:

1. Telah berhasil dibangun Aplikasi Pengelolaan Transaksi dan Manajemen Menu di Café Black Taste Berbasis Web Dan Mobile yang dapat membantu karyawan di Cafe Black Taste dalam proses pemesanan menu, menghitung pembayaran transaksi, dan membantu memeriksa ketersediaan stok menu dan bahan.
2. Berdasarkan pengujian dan implementasi yang telah dilakukan di Cafe Black Taste, dapat disimpulkan bahwa semua fungsi dan *interface* dari aplikasi ini berjalan sesuai dengan yang diharapkan.
3. Berdasarkan hasil kuisisioner yang telah dibagikan kepada 6 karyawan, dari hasil perhitungan kuisisioner tersebut dapat disimpulkan bahwa aplikasi ini mendapatkan rata-rata persentase penilaian sebesar 94%. Berdasarkan persentase tersebut, aplikasi ini termasuk dalam kategori “Sangat Baik”.

### 5.2. Saran

Berdasarkan penelitian yang dilakukan maka diperoleh beberapa saran untuk pengembangan aplikasi ini lebih lanjut sebagai berikut:

1. Aplikasi dapat dikembangkan dengan menambahkan fitur *Delivery Order*, sehingga aplikasi dapat dikomersilkan dan dipakai oleh umum.
2. Aplikasi dapat dikembangkan dengan menambahkan fungsi yang dapat menampilkan menu yang dipesan ditampilkan langsung di dapur.



## Daftar Pustaka

- Buyens, Jim. 2001. *Web Database Development*. Jakarta: Elex Media Komputindo.
- Chasseur, dan Patel, Jm. 2013. *Enabling JSON Document Stores in Relational Systems. Sixteenth International Workshop on the Web and Databases*. WebDB 2013.
- Deviana, H. 2011. *Penerapan XML Web service Pada Sistem Distribusi Barang*. Yogyakarta: Universitas Gadjah Mada.
- Fowler, Martin. 2004. *UML Distilled Panduan Singkat Bahasa pemodelan Objek Standar*. Edisi 3. Yogyakarta: Andi Publishing.
- Fandy Tjiptono & Gregorius Chandra. 2005. *Service, Quality & Satisfaction*, Edisi 2. Yogyakarta: Andi Offset.
- Haviluddin. 2011. *Memahami Penggunaan UML (Unified Modelling Language)*. Samarinda. Universitas Mulawarman.
- Ichwan, M, dan Fifin Hakiky. 2011. *Pengukuran Kinerja Goodreads Application Programming Interface (Api) Pada Aplikasi Mobile Android*. Bandung: Institut Teknologi Nasional Bandung.
- Kadir, Abdul. 2007. *From Zero to A Pro : Membuat Aplikasi Web dengan PHP dan Database MySQL*. Yogyakarta: Andi Offset.

- Khan, Mohd. Ehmer dan Farmeena Khan. 2012. *A Comparative Study of White Box, Black Box, and Grey Box Testing Techniques*. IJACSA. Vol.3 No.6.
- Nazruddin, Safaat H. 2012. *(Edisi Revisi) Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Informatika. Bandung.
- Nidhra, Srinivas dan Jagruthi Dondeti. 2012. *Black box and White box Testing Techniques – A Literature Review*. IJESA. Vol.2 No.2.
- Purnama, Rangsang.2010. *Mari Mengenal J2ME Java 2 Micro Edition*. Jakarta: Prestasi Pustaka.
- Safaat, N. 2011. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika.
- Shalahuddin, M. dan Rosa A. S. 2008. *Analisis dan Desain Sistem Informasi*. Bandung. Politeknik Telkom.
- Sidik,Bertha.2012. *Pemrograman Web dengan PHP*. Badung: Informatika.
- Sommerville, Ian. (2003), *Software Engineering (Rekayasa Perangkat Lunak)*/IanSommerville; alih bahasa, Dra.Yuhilza Hanum M.Eng. ; Hilarius Wibi Hardani. Ed.6. Jakarta: Erlangga.