PEMBANGUNAN SISTEM PENGELOLAAN RESTORAN TERINTEGRASI DENGAN PENGUJIAN KUALITAS BERBASIS ISO 9126

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan

mencapai gelar Sarjana Komputer

Disusun Oleh:

ANHAR TRIBOWO

NIM. 115060807111113

PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2015

repository.ub.ac.id

UNIVERSITAS BRAWIJAYA

LEMBAR PERSETUJUAN

PEMBANGUNAN SISTEM PENGELOLAAN RESTORAN TERINTEGRASI DENGAN PENGUJIAN KUALITAS BERBASIS ISO 9126

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan

mencapai gelar Sarjana Komputer

Disusun Oleh:

ANHAR TRIBOWO

NIM. 115060807111113

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Denny Sagita R., S.Kom., M.Kom.	**Ir. Heru Nurwarsito, M.Kom.**
NIK. 851124 06 1 1 0250	**NIP. 19650402 199002 1 001**

LEMBAR PENGESAHAN

PEMBANGUNAN SISTEM PENGELOLAAN RESTORAN TERINTEGRASI DENGAN PENGUJIAN KUALITAS BERBASIS ISO 9126

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
mencapai gelar Sarjana Komputer

Disusun Oleh:
ANHAR TRIBOWO
NIM. 115060807111113

Skripsi ini telah diuji dan dinyatakan lulus pada
15 Juni 2015

Penguji I

Tri Astoto Kurniawan, S.T., M.T., Ph.D.
NIP. 19710518 200312 1 001

Penguji II

Fajar Pradana, S.ST., M.Eng.
NIK. 871121 16 1 1 0371

Penguji III

Wibisono Sukmo Wardhono, S.T., M.T.
NIK. 820404 06 1 1 0091

Mengetahui,
Ketua Program Studi Informatika/Ilmu Komputer

Drs. Marji, M.T.
NIP. 19670801 199203 1 001

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 pustaka.

Apabila ternyata di dalam SKRIPSI ini dapat dibuktikan terdapat unsur-unsur 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, 18 Juni 2015
Mahasiswa

Anhar Tribowo
NIM. 115060807111113

ABSTRAK

Restoran merupakan salah satu jenis usaha yang mementingkan kualitas pelayanannya. Namun dalam kegiatan operasionalnya, masalah keterlambatan pelayanan sering muncul karena kurangnya koordinasi yang baik antar bagian. Salah satu faktornya adalah jarak antara dapur dan ruang makan. Sistem pengelolaan restoran terintegrasi dapat dibangun untuk mengurangi masalah tersebut. Namun pada penggunaannya, sistem harus memiliki kualitas yang baik agar dapat mendukung jalannya kegiatan operasional restoran. Sebab kualitas yang buruk pada sistem dapat berpotensi mengakibatkan *error* sehingga pihak yang menggunakannya dapat mengalami kerugian finansial. Penelitian ini membahas tentang pembangunan sistem pengelolaan restoran terintegrasi yang memenuhi standar karakteristik utama model kualitas ISO 9126. ISO 9126 memiliki enam karakteristik utama yaitu *functionality, reliability, usability, efficiency, maintainability,* dan *portability*. *Functionality* diuji dengan menggunakan pengujian validasi terhadap seluruh kebutuhan fungsional, *reliability* diuji dengan menggunakan *tool* WAPT (*Web Application Performance Tools*) yang fokus pada ketersediaan *server* dengan *sessions, pages,* dan *hits* sebagai parameternya, *usability* diuji dengan menggunakan model kuesioner SUS (*System Usability Scale*) yang disebar kepada lima belas responden, *efficiency* juga diuji dengan menggunakan *tool* WAPT namun fokus pada waktu respon *server*, *maintainability* diuji dengan menggunakan persamaan MI (*Maintainability Index*), dan *portability* diuji dengan menggunakan *tool* BrowseEmAll yang fokus pada kemampuan sistem untuk dapat dioperasikan di *web browser* yang berbeda. Berdasarkan hasil pengujian, sistem telah memenuhi standar karakteristik utama ISO 9126 yaitu dengan *functionality* sebesar 1, *reliability* sebesar 100%, *usability* sebesar 68,5, *efficiency* sebesar 0,78 detik, *maintainability* sebesar 66,62597, dan hasil *portability* menunjukkan tidak ditemukannya *error* maupun *warning* di tujuh jenis *desktop browser* dan lima jenis *mobile browser*. Selain itu, hasil pengujian integrasi menunjukkan bahwa seluruh proses yang melibatkan interaksi antar kelas telah terintegrasi dengan baik.

Kata Kunci: Sistem Pengelolaan Restoran Terintegrasi, Pelayanan Restoran, Model Kualitas, ISO 9126

repository.ub.ac.id

ABSTRACT

Restaurant is one of the types of businesses concerned with the quality of service. But in its operational activities, service delay problems often arise because of the lack of good coordination between sections. One of the factors is the distance between the kitchen and dining room. The integrated restaurant management system can be developed to reduce the related problem. However, the system must have a good quality, so it can support the restaurant operational activities. The poor system quality can potentially result an error, so that who use it may get financial losses. This study discusses the development of the integrated restaurant management system that is meet the standards of the main characteristics of the ISO 9126 quality model. ISO 9126 has six main characteristics namely functionality, reliability, usability, efficiency, maintainability, and portability. Functionality was tested using validation testing to all functional requirements, reliability was tested using WAPT (Web Application Performance Tools) which focuses on server availability with sessions, pages, and hits as the parameter, usability was tested using SUS (System Usability Scale) questionnaire model that distributed to fifteen respondents, efficiency also tested by using WAPT but focus on server response time, maintainability was tested using the MI (Maintainability Index) equation, and portability was tested using BrowseEmAll that focuses on the system ability to be operated in different web browser. Based on the testing result, system has met the standard of all main characteristics of ISO 9126 namely with the functionality worth 1, reliability is 100%, usability worth 68,5, efficiency is 0,78 seconds, maintainability worth 66,62597, and the portability result showed that there were no errors or warnings in seven types of desktop browser and five types of mobile browser. In addition, the integration testing results indicate that the integration of the entire process that involves interaction between the classes has been well integrated.

Keywords: Integrated Restaurant Management System, Restaurant Service, Quality Model, ISO 9126

UNIVERSITAS BRAWIJAYA

KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa yang telah melimpahkan rahmat, taufik, dan hidayah-Nya sehingga laporan skripsi dengan judul "Pembangunan Sistem Pengelolaan Restoran Terintegrasi dengan Pengujian Kualitas Berbasis ISO 9126" ini dapat diselesaikan.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Denny Sagita Rusdianto, S.Kom., M.Kom. selaku dosen pembimbing I yang telah memberikan ilmu dan saran selama penyusunan skripsi ini.
2. Bapak Ir. Heru Nurwarsito, M.Kom. selaku dosen pembimbing II yang juga memberikan ilmu dan saran selama penyusunan skripsi ini.
3. Orang tua penulis yaitu Bapak Arief Martono dan Ibu Lasmini beserta seluruh keluarga besar penulis atas segala nasihat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini.
4. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D., Bapak Fajar Pradana, S.ST., M.Eng., dan Bapak Wibisono Sukmo Wardhono, S.T., M.T. selaku dosen penguji skripsi yang telah memberikan saran perbaikan dalam skripsi ini.
5. Bapak Drs. Marji, M.T. selaku ketua Program Studi Informatika.
6. Bapak Barlian Henryranu Prasetio, S.T., M.T. selaku dosen penasihat akademik yang selalu memberikan nasihat kepada penulis selama menempuh masa studi.
7. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada penulis selama menempuh pendidikan di Informatika Universitas Brawijaya.
8. Seluruh mahasiswa Informatika Universitas Brawijaya 2011 khususnya teman-teman seperjuangan skripsi di Laboratorium Rekayasa Perangkat Lunak yang telah memotivasi dan mendukung terealisasinya skripsi ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 18 Juni 2015

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN ..... **ii**
LEMBAR PENGESAHAN ..... **iii**
PERNYATAAN ORISINALITAS ..... **iv**
ABSTRAK ..... **v**
KATA PENGANTAR ..... **vii**
DAFTAR ISI ..... **viii**
DAFTAR TABEL ..... **xii**
DAFTAR GAMBAR ..... **xiv**
DAFTAR LAMPIRAN ..... **xvii**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Batasan Masalah 3
1.4 Tujuan 3
1.5 Manfaat 3
1.6 Sistematika Penulisan 3
BAB II KAJIAN PUSTAKA DAN DASAR TEORI **4**
2.1 Kajian Pustaka 4
2.2 Restoran 5
2.3 Rekayasa Perangkat Lunak 7
2.4 Model *Waterfall* 7
2.5 UML (*Unified Modeling Language*) 8
2.5.1 Diagram *Use Case* 9
2.5.2 Diagram Kelas 10
2.5.3 Diagram Aktivitas 11
2.5.4 Diagram Sekuen 11
2.6 Pola Arsitektur MVC (*Model, View, Controller*) 13
2.7 CodeIgniter 14
2.8 Pengujian Integrasi 15
2.9 ISO 9126 15

2.9.1 *Functionality* 16
2.9.2 *Reliability* 17
2.9.3 *Usability* 17
2.9.4 *Efficiency* 18
2.9.5 *Maintainability* 18
2.9.6 *Portability* 19
BAB III METODOLOGI PENELITIAN 20
3.1 Studi Literatur 20
3.2 Analisis Kebutuhan 22
3.3 Perancangan Sistem 24
3.4 Implementasi 24
3.5 Pengujian dan Analisis 25
3.5.1 Pengujian Unit 25
3.5.2 Pengujian Integrasi 25
3.5.3 Pengujian Validasi 26
3.5.4 Pengujian Berbasis ISO 9126 26
3.6 Pengambilan Kesimpulan dan Saran 29
BAB IV PERANCANGAN DAN IMPLEMENTASI 30
4.1 Gambaran Umum Sistem 30
4.2 Analisis Kebutuhan 31
4.2.1 Elisitasi Kebutuhan 31
4.2.2 Identifikasi Aktor 33
4.2.3 Spesifikasi Kebutuhan 33
4.2.4 Diagram *Use Case* 35
4.3 Perancangan Sistem 49
4.3.1 Perancangan Arsitektural 50
4.3.2 Perancangan Diagram Kelas 51
4.3.3 Perancangan Basis Data 54
4.3.4 Perancangan Diagram Aktivitas 55
4.3.5 Perancangan Diagram Sekuen 66
4.3.6 Perancangan Antarmuka 77

4.4 Implementasi 80
4.4.1 Spesifikasi Lingkungan Implementasi 80
4.4.2 Batasan Implementasi 81
4.4.3 Implementasi Basis Data 81
4.4.4 Implementasi Kelas 82
4.4.5 Implementasi Algoritma 83
4.4.6 Implementasi Antarmuka 87
BAB V PENGUJIAN DAN ANALISIS 90
5.1 Pengujian Unit 90
5.1.1 Tujuan 90
5.1.2 Mekanisme 90
5.1.3 Hasil Pengujian 90
5.1.4 Analisis 99
5.2 Pengujian Integrasi 99
5.2.1 Tujuan 99
5.2.2 Mekanisme 99
5.2.3 Hasil Pengujian 104
5.2.4 Analisis 106
5.3 Pengujian Validasi 106
5.3.1 Tujuan 106
5.3.2 Mekanisme 106
5.3.3 Hasil Pengujian 114
5.3.4 Analisis 116
5.4 Pengujian Berbasis ISO 9126 116
5.4.1 *Functionality* 116
5.4.2 *Reliability* 117
5.4.3 *Usability* 119
5.4.4 *Efficiency* 120
5.4.5 *Maintainability* 122
5.4.6 *Portability* 124
BAB VI PENUTUP 126

6.1 Kesimpulan 126
6.2 Saran 126
DAFTAR PUSTAKA 128
LAMPIRAN 131

DAFTAR TABEL

Tabel 2.1 Area Fungsional pada Sistem Pengelolaan Restoran 4
Tabel 2.2 Simbol-Simbol Diagram *Use Case* 9
Tabel 2.3 Simbol-Simbol Diagram Kelas 10
Tabel 2.4 Simbol-Simbol Diagram Aktivitas 11
Tabel 2.5 Simbol-Simbol Diagram Sekuen 12
Tabel 2.6 Standar Jakob Nielsen 18
Tabel 2.7 Standar *Maintainability Index* (MI) 19
Tabel 4.1 Identifikasi Aktor 33
Tabel 4.2 Spesifikasi Kebutuhan Fungsional 34
Tabel 4.3 Spesifikasi Kebutuhan Non Fungsional 35
Tabel 4.4 Skenario Membuat Pemesanan Menu 36
Tabel 4.5 Skenario Melihat Status Pemesanan Menu 37
Tabel 4.6 Skenario Membuat Tagihan Pembayaran 38
Tabel 4.7 Skenario Melihat Tagihan Pembayaran 39
Tabel 4.8 Skenario Melihat Daftar Pesanan 40
Tabel 4.9 Skenario Notifikasi Pesanan 41
Tabel 4.10 Skenario Mencetak Tagihan Pembayaran 41
Tabel 4.11 Skenario Mencari Pesanan 42
Tabel 4.12 Skenario Memvalidasi Pesanan 43
Tabel 4.13 Skenario Mencetak Pesanan 44
Tabel 4.14 Skenario Kelola Menu Restoran 45
Tabel 4.15 Skenario Kelola Pengguna 46
Tabel 4.16 Skenario Mengubah Informasi Restoran 47
Tabel 4.17 Skenario Melihat Histori Transaksi 48
Tabel 4.18 Skenario Login 49
Tabel 4.19 Spesifikasi Perangkat Keras 81
Tabel 4.20 Spesifikasi Perangkat Lunak 81
Tabel 4.21 Implementasi Kelas 82
Tabel 5.1 Kasus Uji Algoritma melihatStatusPemesananMenu 92
Tabel 5.2 Kasus Uji Algoritma memberikanNotifikasiPesanan 95
Tabel 5.3 Kasus Uji Algoritma melihatDaftarPesanan 98
Tabel 5.4 Kasus Uji Integrasi Unit PengelolaanPemesanan dan Pemesanan 99
Tabel 5.5 Kasus Uji Integrasi Unit PengelolaanPemesanan dan ModAutentikasi 100
Tabel 5.6 Kasus Uji Integrasi Unit PengelolaanPemesanan dan MenuRestoran 100
Tabel 5.7 Kasus Uji Integrasi Unit PengelolaanPemesanan dan Restoran 101
Tabel 5.8 Kasus Uji Integrasi Unit Pemesanan dan DetailPesan 101
Tabel 5.9 Kasus Uji Integrasi Unit PengelolaanPembayaran dan Pembayaran .. 102

Tabel 5.10 Kasus Uji Integrasi Unit PengelolaanPembayaran dan ModAutentikasi 102
Tabel 5.11 Kasus Uji Integrasi Unit PengelolaanPembayaran dan Pembayaran 103
Tabel 5.12 Kasus Uji Integrasi Unit Pembayaran dan DetailBayar..... 104
Tabel 5.13 Hasil Pengujian Integrasi 104
Tabel 5.14 Kasus Uji Validasi Membuat Pemesanan Menu..... 106
Tabel 5.15 Kasus Uji Validasi Melihat Status Pemesanan Menu 107
Tabel 5.16 Kasus Uji Validasi Membuat Tagihan Pembayaran..... 107
Tabel 5.17 Kasus Uji Validasi Melihat Tagihan Pembayaran 107
Tabel 5.18 Kasus Uji Validasi Melihat Daftar Pesanan..... 108
Tabel 5.19 Kasus Uji Validasi Notifikasi Pesanan 108
Tabel 5.20 Kasus Uji Validasi Mencetak Tagihan Pembayaran..... 108
Tabel 5.21 Kasus Uji Validasi Mencari Pesanan 109
Tabel 5.22 Kasus Uji Validasi Memvalidasi Pesanan..... 109
Tabel 5.23 Kasus Uji Validasi Mencetak Pesanan 110
Tabel 5.24 Kasus Uji Validasi Kelola Menu Restoran 110
Tabel 5.25 Kasus Uji Validasi Kelola Pengguna 111
Tabel 5.26 Kasus Uji Validasi Mengubah Informasi Restoran..... 111
Tabel 5.27 Kasus Uji Validasi Melihat Histori Transaksi..... 111
Tabel 5.28 Kasus Uji Validasi Login 112
Tabel 5.29 Kasus Uji Validasi *Reliability* 112
Tabel 5.30 Kasus Uji Validasi *Usability* 112
Tabel 5.31 Kasus Uji Validasi *Efficiency*..... 113
Tabel 5.32 Kasus Uji Validasi *Maintainability* 113
Tabel 5.33 Kasus Uji Validasi *Portability* 113
Tabel 5.34 Hasil Pengujian Validasi..... 114
Tabel 5.35 Ringkasan Hasil Pengujian *Reliability* 118
Tabel 5.36 Hasil Pengujian *Usability* 120
Tabel 5.37 Ringkasan Hasil Pengujian *Efficiency*..... 121
Tabel 5.38 Ringkasan Hasil Pengujian *Maintainability* 123
Tabel 5.39 Ringkasan Hasil Pengujian *Portability* 124

DAFTAR GAMBAR

Gambar 2.1 Prosedur Pelayanan Restoran dengan Cara Manual 6
Gambar 2.2 Tahapan Umum Rekayasa Perangkat Lunak 7
Gambar 2.3 Ilustrasi Model *Waterfall* 7
Gambar 2.4 Diagram UML (*Unified Modeling Language*) 8
Gambar 2.5 Pola Arsitektur MVC (*Model, View, Controller*) 13
Gambar 2.6 Struktur Folder CodeIgniter 14
Gambar 2.7 Karakteristik Utama Model Kualitas ISO 9126 16
Gambar 3.1 Diagram Alir Tahapan Penelitian 20
Gambar 3.2 Diagram Alir Pengembangan Sistem dengan Model *Waterfall* 21
Gambar 3.3 Tahapan Pengujian Unit 25
Gambar 3.4 Tahapan Pengujian Integrasi 26
Gambar 3.5 Tahapan Pengujian Validasi 26
Gambar 3.6 Tahapan Uji Karakteristik *Functionality* 27
Gambar 3.7 Tahapan Uji Karakteristik *Reliability* 27
Gambar 3.8 Tahapan Uji Karakteristik *Usability* 27
Gambar 3.9 Tahapan Uji Karakteristik *Efficiency* 28
Gambar 3.10 Tahapan Uji Karakteristik *Maintainability* 28
Gambar 3.11 Tahapan Uji Karakteristik *Portability* 29
Gambar 4.1 Logo OtseResto 30
Gambar 4.2 Diagram Alir Analisis Kebutuhan 31
Gambar 4.3 Proses Bisnis Utama Restoran 33
Gambar 4.4 Diagram *Use Case* OtseResto 36
Gambar 4.5 Diagram Alir Perancangan Sistem 50
Gambar 4.6 Arsitektur OtseResto 50
Gambar 4.7 Diagram Kelas OtseResto Level Analisis 51
Gambar 4.8 Diagram Kelas *Controller* OtseResto Level Implementasi 52
Gambar 4.9 Diagram Kelas *Model* OtseResto Level Implementasi 53
Gambar 4.10 *Entity Relationship Diagram* (ERD) OtseResto 54
Gambar 4.11 Diagram Aktivitas Membuat Pemesanan Menu 55
Gambar 4.12 Diagram Aktivitas Melihat Status Pemesanan Menu 56
Gambar 4.13 Diagram Aktivitas Membuat Tagihan Pembayaran 56
Gambar 4.14 Diagram Aktivitas Melihat Tagihan Pembayaran 57
Gambar 4.15 Diagram Aktivitas Mencetak Tagihan Pembayaran 57
Gambar 4.16 Diagram Aktivitas Melihat Daftar Pesanan 58
Gambar 4.17 Diagram Aktivitas Notifikasi Pesanan 58
Gambar 4.18 Diagram Aktivitas Memvalidasi Pesanan 59
Gambar 4.19 Diagram Aktivitas Mencari Pesanan 59
Gambar 4.20 Diagram Aktivitas Mencetak Pesanan 60
Gambar 4.21 Diagram Aktivitas Membuat Menu Restoran 61

Gambar 4.22 Diagram Aktivitas Melihat Menu Restoran 61
Gambar 4.23 Diagram Aktivitas Mengubah Menu Restoran 62
Gambar 4.24 Diagram Aktivitas Menghapus Menu Restoran 62
Gambar 4.25 Diagram Aktivitas Membuat Pengguna 63
Gambar 4.26 Diagram Aktivitas Melihat Pengguna 63
Gambar 4.27 Diagram Aktivitas Mengubah Pengguna 64
Gambar 4.28 Diagram Aktivitas Menghapus Pengguna 64
Gambar 4.29 Diagram Aktivitas Mengubah Informasi Restoran 65
Gambar 4.30 Diagram Aktivitas Melihat Histori Transaksi 65
Gambar 4.31 Diagram Aktivitas Login 66
Gambar 4.32 Diagram Sekuen Membuat Pemesanan Menu 67
Gambar 4.33 Diagram Sekuen Melihat Status Pemesanan Menu 67
Gambar 4.34 Diagram Sekuen Membuat Tagihan Pembayaran 68
Gambar 4.35 Diagram Sekuen Melihat Tagihan Pembayaran 68
Gambar 4.36 Diagram Sekuen Mencetak Tagihan Pembayaran 69
Gambar 4.37 Diagram Sekuen Melihat Daftar Pesanan 69
Gambar 4.38 Diagram Sekuen Notifikasi Pesanan 70
Gambar 4.39 Diagram Sekuen Memvalidasi Pesanan 70
Gambar 4.40 Diagram Sekuen Mencari Pesanan 71
Gambar 4.41 Diagram Sekuen Mencetak Pesanan 71
Gambar 4.42 Diagram Sekuen Membuat Menu Restoran 72
Gambar 4.43 Diagram Sekuen Melihat Menu Restoran 72
Gambar 4.44 Diagram Sekuen Mengubah Menu Restoran 73
Gambar 4.45 Diagram Sekuen Menghapus Menu Restoran 73
Gambar 4.46 Diagram Sekuen Membuat Pengguna 74
Gambar 4.47 Diagram Sekuen Melihat Pengguna 74
Gambar 4.48 Diagram Sekuen Mengubah Pengguna 75
Gambar 4.49 Diagram Sekuen Menghapus Pengguna 75
Gambar 4.50 Diagram Sekuen Mengubah Informasi Restoran 76
Gambar 4.51 Diagram Sekuen Melihat Histori Transaksi 76
Gambar 4.52 Diagram Sekuen Login 77
Gambar 4.53 Perancangan Antarmuka *Login* 77
Gambar 4.54 Perancangan Antarmuka Pelayan 78
Gambar 4.55 Perancangan Antarmuka Kasir 78
Gambar 4.56 Perancangan Antarmuka Koki 79
Gambar 4.57 Perancangan Antarmuka Manajer 79
Gambar 4.58 Diagram Alir Implementasi 80
Gambar 4.59 Skema Basis Data OtseResto 82
Gambar 4.60 Algoritma Membuat Pemesanan Menu 84
Gambar 4.61 Algoritma Melihat Status Pemesanan Menu 85
Gambar 4.62 Algoritma Membuat Tagihan Pembayaran 85

Gambar 4.63 Algoritma Notifikasi Pesanan 86
Gambar 4.64 Algoritma Melihat Daftar Pesanan..... 87
Gambar 4.65 Implementasi Antarmuka *Login* 87
Gambar 4.66 Implementasi Antarmuka Pelayan 87
Gambar 4.67 Implementasi Antarmuka Kasir 88
Gambar 4.68 Implementasi Antarmuka Koki..... 88
Gambar 4.69 Implementasi Antarmuka Manajer 89
Gambar 5.1 Pembentukan *Node* Algoritma melihatStatusPemesananMenu..... 91
Gambar 5.2 *Flowgraph* Algoritma melihatStatusPemesananMenu..... 91
Gambar 5.3 Pembentukan *Node* Algoritma memberikanNotifikasiPesanan 94
Gambar 5.4 *Flowgraph* Algoritma memberikanNotifikasiPesanan 94
Gambar 5.5 Pembentukan *Node* Algoritma melihatDaftarPesanan 97
Gambar 5.6 *Flowgraph* Algoritma melihatDaftarPesanan 97

DAFTAR LAMPIRAN

Lampiran 1 Lembar Pesanan Restoran Mie Jogging Cabang Soekarno Hatta 131
Lampiran 2 Struk Pembayaran Restoran Mie Jogging Cabang Soekarno Hatta . 131
Lampiran 3 Lembar Pesanan Restoran Mie Jogging Cabang Tlogomas..... 131
Lampiran 4 Struk Pembayaran Restoran Mie Jogging Cabang Tlogomas..... 132
Lampiran 5 Struk Pembayaran Restoran Mie Setan 132
Lampiran 6 Struk Pembayaran Restoran Noodle Inc 132
Lampiran 7 Struk Pembayaran Restoran Griya Bebek dan Ayam Nelongso..... 133
Lampiran 8 Struk Pembayaran Restoran Bebek Sinjay 133
Lampiran 9 Model Kuesioner SUS untuk Pengujian *Usability* 134
Lampiran 10 Hasil Pengujian *Reliability* dengan WAPT 135
Lampiran 11 Hasil Pengujian *Efficiency* dengan WAPT 140
Lampiran 12 Hasil Pengujian *Maintainability* dengan DMSSearchEngine..... 141
Lampiran 13 Hasil Pengujian *Portability* Halaman Login 142
Lampiran 14 Hasil Pengujian *Portability* Halaman Utama Pelayan 142
Lampiran 15 Hasil Pengujian *Portability* Halaman Buat Pemesanan..... 142
Lampiran 16 Hasil Pengujian *Portability* Lihat Status Pemesanan..... 142
Lampiran 17 Hasil Pengujian *Portability* Halaman Utama Kasir 143
Lampiran 18 Hasil Pengujian *Portability* Halaman Buat Tagihan..... 143
Lampiran 19 Hasil Pengujian *Portability* Halaman Lihat Tagihan 143
Lampiran 20 Hasil Pengujian *Portability* Halaman Utama Koki 143
Lampiran 21 Hasil Pengujian *Portability* Halaman Lihat Daftar Pesanan 144
Lampiran 22 Hasil Pengujian *Portability* Halaman Cari Pesanan 144
Lampiran 23 Hasil Pengujian *Portability* Halaman Utama Manajer 144
Lampiran 24 Hasil Pengujian *Portability* Halaman Tambah Menu Restoran..... 144
Lampiran 25 Hasil Pengujian *Portability* Halaman Lihat Menu Restoran 145
Lampiran 26 Hasil Pengujian *Portability* Halaman Ubah Menu Restoran 145
Lampiran 27 Hasil Pengujian *Portability* Halaman Tambah Pengguna..... 145
Lampiran 28 Hasil Pengujian *Portability* Halaman Lihat Pengguna 145
Lampiran 29 Hasil Pengujian *Portability* Halaman Ubah Pengguna 146
Lampiran 30 Hasil Pengujian *Portability* Halaman Ubah Informasi Restoran 146
Lampiran 31 Hasil Pengujian *Portability* Halaman Lihat Histori Transaksi 146

BAB I
PENDAHULUAN

1.1 Latar Belakang

Restoran merupakan salah satu jenis usaha yang mementingkan kualitas pelayanannya. Sehingga, pelaku bisnis ini harus memiliki pengawasan pelayanan yang baik terhadap kegiatan operasional restoran (Saputra & Harjanti, 2013). Dalam kegiatan operasionalnya, setiap bagian yang ada pada restoran harus dapat berkoordinasi dengan cepat guna meningkatkan kualitas pelayanan restoran. Namun kenyataannya, sering terjadi keterlambatan pelayanan yang disebabkan kurangnya koordinasi yang baik antar bagian dalam restoran (Atmodjo, 2005).

Jarak antara dapur dan ruang makan merupakan salah satu faktor kurangnya koordinasi yang baik. Menurut Atmodjo (2005), apabila jarak antar bagian tersebut cukup jauh, maka dapat dilakukan pembentukan kerja sama beberapa orang pelayan tambahan. Namun, hal tersebut dinilai kurang efisien dan efektif sebab membutuhkan tambahan sumber daya. Sehingga, pemilik restoran akan membutuhkan biaya yang tidak sedikit untuk menggaji beberapa orang pelayan tambahan tersebut.

Untuk mengurangi permasalahan tersebut, maka dibutuhkan sebuah sistem yang dapat mengurangi keterlambatan pelayanan dalam koordinasi antara dapur dan ruang makan. Pengelolaan antar bagian tersebut dapat dilakukan dengan adanya sebuah sistem informasi yang terintegrasi. Di mana sistem informasi merupakan sistem yang mengelola data pada sebuah bagian dalam organisasi untuk menjadi informasi yang bermanfaat bagi penggunanya. Sedangkan sistem informasi yang terintegrasi merupakan penyempurnaan dari sistem informasi konvensional di mana sistem ini memiliki integrasi data pada bagian yang terpisah dalam organisasi sehingga pengelolaan lebih efisien dan efektif (Perdananugraha, 2010). Maka dari itu, solusi yang akan dilakukan adalah membangun sistem pengelolaan restoran terintegrasi, yaitu sistem yang dapat mengelola integrasi data antar bagian yang terpisah sehingga pengelolaan data lebih efisien dan efektif serta mengurangi keterlambatan pelayanan dalam koordinasi antar bagian dalam restoran (Sigala, 2003). Sistem yang dibangun ini mencakup dua area fungsional utama yaitu operasional pada bagian dapur dan pemasaran pada bagian POS (*Point of Sales*) serta area pendukung manajerial untuk manajer restoran.

Tugas utama dari sistem tersebut adalah mendukung jalannya kegiatan operasional restoran. Sehingga, sistem dituntut memiliki kualitas yang baik agar dapat digunakan secara maksimal dan tidak menghambat kegiatan operasional di restoran. Sebab menurut Jamwal (2010), kualitas yang buruk pada sistem dapat berpotensi mengakibatkan *error* sehingga pihak yang menggunakannya dapat mengalami kerugian finansial.

Kualitas sistem tersebut dapat diuji melalui berbagai jenis pengujian perangkat lunak. Namun hanya dengan pengujian berbasis model kualitas, sistem dapat diuji dengan komprehensif sebab memiliki karakteristik-karakteristik untuk standar produk yang tidak ditemukan pada pengujian konvensional seperti *black-box* atau *white-box* (Jamwal, 2010). Terdapat dua sudut pandang dalam pengujian kualitas sebuah sistem yaitu sudut pandang produk dan proses. Untuk menguji kualitas sebuah sistem yang sudah diimplementasikan maka diperlukan sebuah model kualitas dari sudut pandang produk (Wahono, 2006).

Terdapat beberapa model kualitas dari sudut pandang produk yang populer yaitu McCall, Boehm, FURPS, dan ISO 9126. McCall adalah model kualitas yang muncul pertama, namun model tersebut tidak memiliki karakteristik *functionality*. Kemudian model kualitas Boehm memiliki keunggulan pada karakteristik yang melibatkan faktor perangkat keras namun kekurangannya *functionality* juga tidak ada dalam daftar karakteristiknya. Lalu FURPS memiliki pembagian karakteristik fungsional dan non fungsional namun tidak ada *portability* dalam daftar karakteristiknya. Berdasarkan perbedaan karakteristik tersebut, muncul ISO 9126 yang dibuat berdasarkan musyawarah internasional dan persetujuan dari semua anggota ISO sehingga memiliki karakteristik yang lebih lengkap dibanding model kualitas yang lain (Fahmy, et al., 2012) (Jamwal, 2010). Maka dari itu, ISO 9126 dapat digunakan untuk pengujian kualitas sistem pada penelitian ini.

Berdasarkan uraian sebelumnya, ISO 9126 dapat dipilih untuk pengujian kualitas dari sistem pengelolaan restoran terintegrasi ini sebab ISO 9126 memiliki karakteristik yang lengkap dan satu-satunya model kualitas yang dibuat berdasarkan musyawarah internasional dan persetujuan dari semua anggota organisasi ISO. Sistem pengelolaan restoran terintegrasi ini difokuskan untuk area fungsional pemasaran pada bagian POS (*Point of Sales*) dan area fungsional operasional pada bagian dapur sehingga diharapkan dapat mengurangi permasalahan koordinasi antar bagian tersebut. Selain itu, pada tahap akhir pembangunan akan dilakukan pengujian kualitas sistem menggunakan model kualitas ISO 9126 sebagai pengujian sebuah kualitas produk yang baik.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya maka didapatkan rumusan masalah sebagai berikut:

1. Bagaimana menganalisis dan merancang sistem pengelolaan restoran terintegrasi?
2. Bagaimana mengimplementasikan sistem pengelolaan restoran yang terintegrasi?
3. Bagaimana pengujian dengan karakteristik utama model kualitas ISO 9126 terhadap sistem yang dikembangkan?

1.3 Batasan Masalah

Penelitian ini dibatasi oleh hal-hal sebagai berikut:

1. Sistem yang dikembangkan berbasis web.
2. Sistem yang dikembangkan fokus pada dua area fungsional pada restoran yaitu operasional pada bagian dapur dan pemasaran pada bagian POS.
3. Karakteristik model kualitas ISO 9126 yang digunakan adalah *functionality, reliability, usability, efficiency, maintainability,* dan *portability*.

1.4 Tujuan

Tujuan yang ingin dicapai dari penelitian ini adalah mengembangkan sebuah sistem pengelolaan restoran terintegrasi yang memenuhi standar karakteristik utama model kualitas ISO 9126.

1.5 Manfaat

Manfaat yang didapatkan dari penelitian ini adalah mempercepat pelayanan dalam koordinasi antar bagian pada restoran dan mengimplementasikan model kualitas ISO 9126.

1.6 Sistematika Penulisan

Sistematika penulisan ini ditujukan untuk memberikan gambaran dan uraian singkat terkait penulisan skripsi ini yang meliputi beberapa bab, yaitu:

1. **BAB 1 Pendahuluan**
   Bab ini memuat tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan.
2. **BAB 2 Kajian Pustaka dan Dasar Teori**
   Bab ini membahas kajian dari penelitian yang telah dilakukan sebelumnya serta dasar teori yang mendukung penyelesaian masalah dalam penelitian ini.
3. **BAB 3 Metodologi Penelitian**
   Bab ini memaparkan metode atau cara yang digunakan dalam penelitian ini secara umum.
4. **BAB 4 Perancangan dan Implementasi**
   Bab ini membahas tentang analisis kebutuhan, dan perancangan, serta implementasi dari Sistem Pengelolaan Restoran Terintegrasi.
5. **BAB 5 Pengujian dan Analisis**
   Pada bab ini akan dimuat hasil pengujian dan analisis terhadap sistem yang telah dikembangkan berbasis model kualitas ISO 9126.
6. **BAB 6 Penutup**
   Memuat tentang kesimpulan yang didapatkan dari pembangunan dan pengujian sistem dalam penelitian ini serta saran-saran untuk pengembangan lebih lanjut.

BAB II
KAJIAN PUSTAKA DAN DASAR TEORI

Pada bab ini akan dipaparkan tentang kajian pustaka dan dasar teori yang mendukung pembangunan sistem pengelolaan restoran terintegrasi.

2.1 Kajian Pustaka

Analisis kebutuhan pada penelitian ini dilakukan dengan mengkaji penelitian terdahulu. Kajian tersebut dilakukan terhadap penelitian yang relevan terkait pembangunan sistem pengelolaan restoran terintegrasi. Terdapat empat penelitian yang digunakan sebagai bahan untuk menganalisis kebutuhan sistem. Tabel 2.1 memodelkan area fungsional yang diimplementasikan pada masing-masing penelitian.

Tabel 2.1 Area Fungsional pada Sistem Pengelolaan Restoran

No.	Penelitian	Area Fungsional			Area Pendukung
		Pemasaran	Operasional		
		POS	Dapur	Gudang	Manajerial
1.	*Real-Time Process Management System in a Restaurant by Sharing Food Order Information* (Shimmura, et al., 2009).	✓	✓	×	✓
2.	*Intelligent Restaurant System Smartmenu* (Pieskä, et al., 2013).	✓	✓	×	✓
3.	*Development and Evaluation of an RFID-based e-Restaurant System for Customer-centric Service* (Tan & Chang, 2010).	✓	✓	✓	✓
4.	*Development of an RFID-based Sushi Management System* (Ngai, et al., 2008).	✓	✓	✓	✓

Penelitian yang dilakukan oleh Shimmura dengan judul "*Real-Time Process Management System in a Restaurant by Sharing Food Order Information*" membahas tentang sistem pengelolaan restoran yang *real-time* dengan menggunakan integrasi antara bagian POS (*Point of Sales*) dengan bagian dapur. Penekanan penelitiannya terletak pada tampilan pesanan (*order list*) di bagian

dapur yang merupakan pusat integrasi antar bagian POS. Pada akhir penelitiannya, disimpulkan bahwa implementasi sistem tersebut dapat membuat proses pelayanan pada restoran lebih efisien dan mengurangi keluhan dari tamu (Shimmura, et al., 2009).

Penelitian selanjutnya dilakukan oleh Pieskä, et al. dengan judul "*Intelligent Restaurant System Smartmenu*". Penggunaan bagian yang diimplementasikan pada penelitiannya hampir sama dengan penelitian Shimmura, et al. (2009). Perbedaan yang terlihat adalah keterlibatan tamu dalam sistem dan penggunaan terminal robot yang dapat digunakan tamu untuk memesan menu restoran. Sistem yang dibangun tersebut dapat mengotomatisasi proses pemesanan dari tamu atau pelayan menggunakan perangkat bergerak berbasis Android. Sedangkan terminal robot dapat digunakan sebagai cara alternatif tamu dalam memesan sebuah menu restoran (Pieskä, et al., 2013).

Penelitian ketiga dilakukan oleh Tan & Chang dengan judul "*Development and Evaluation of an RFID-based e-Restaurant System for Customer-centric Service*". Dalam penelitiannya, dibahas pembangunan dan evaluasi sistem pengelolaan restoran berbasis RFID (*Radio Frequency Identification*). Di mana sistem yang dibangun tersebut dapat mengenali identitas tamu melalui teknologi RFID sehingga profil mengenai menu favorit dan sejenisnya dapat diketahui. Bagian yang diimplementasikan pada sistem ini melibatkan tiga bagian pada restoran yaitu POS (kasir, pelayan, dan tamu), dapur, dan gudang. Pada akhir penelitiannya, disimpulkan bahwa sistem yang dibangun dapat meningkatkan efektivitas dalam penyediaan layanan restoran berbasis *customer-centric* (Tan & Chang, 2010).

Penelitian terakhir yang dikaji berjudul "*Development of an RFID-based Sushi Management System*" oleh Ngai, et al. Penelitian ini hampir serupa dengan penelitian yang dilakukan oleh Tan & Chang (2010) yang menggunakan RFID dalam implementasinya. Bagian yang diimplementasikannya juga serupa yaitu POS, dapur, dan gudang. Namun terdapat perbedaan yang mencolok yaitu pada penggunaan *conveyor belt*. Alat tersebut digunakan untuk mengotomatisasi beberapa bagian pada restoran sehingga dapat meningkatkan kualitas pelayanannya (Ngai, et al., 2008).

Berdasarkan kajian terhadap empat penelitian yang telah diuraikan sebelumnya, maka didapatkan bagian-bagian umum yang digunakan di sebagian besar restoran. Bagian-bagian tersebut adalah POS (*Point of Sale*), dapur, dan manajerial. Di mana bagian POS terdapat pada area fungsional pemasaran (*marketing*), bagian dapur terdapat pada area fungsional operasional (*operations*), dan bagian manajerial untuk manajer.

2.2 Restoran

Restoran merupakan suatu tempat yang dikelola secara komersial, yang menyelenggarakan pelayanan dengan baik kepada semua tamu berupa makan dan minum. Tujuan dari operasi restoran tersebut yaitu untuk mencari

keuntungan seperti yang didefinisikan oleh Prof. Vanco Christian dari *School Hotel Administration* di Cornell University. Selain bertujuan bisnis atau mencari keuntungan, membuat puas para tamu juga merupakan tujuan operasi restoran yang utama (Atmodjo, 2005).

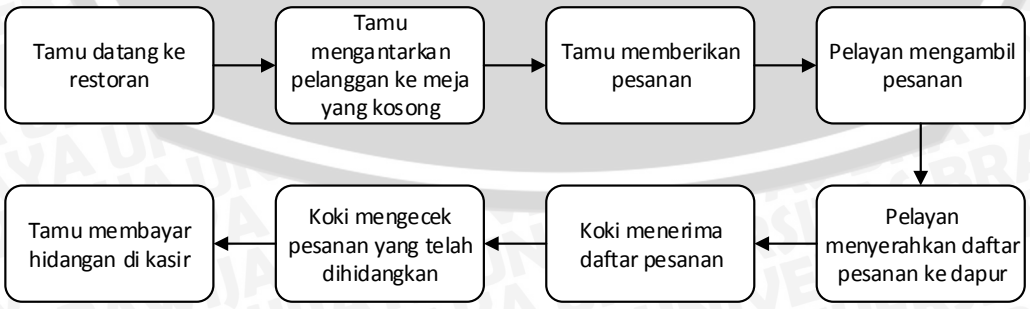
Dalam kegiatan operasionalnya, restoran memiliki empat tipe dasar pelayanan yang terkenal, yaitu:

1. *Table service*, merupakan sistem pelayanan restoran di mana penyajiannya dilakukan di atas meja makan oleh para pelayan (Atmodjo, 2005).
2. *Counter* service, merupakan sistem pelayanan restoran di mana penyajiannya dilakukan di atas meja *counter*/bar oleh para pelayan (Atmodjo, 2005).
3. *Self service,* disebut juga sebagai pelayanan bufet yaitu semua makanan secara lengkap disajikan di meja prasmanan (Atmodjo, 2005).
4. *Carry out service*, disebut juga sebagai *take out service* yaitu pelayanan yang hanya menyediakan makanan untuk dibungkus dalam boks/kotak makan (Atmodjo, 2005).

Setiap restoran memiliki tiga area fungsional utama yang dapat dijalankan untuk meraih kesuksesannya (Laube, 2015).

1. Operasional
   Area ini menangani seluruh kegiatan fungsional terkait dengan persiapan dan pelayanan terhadap tamu. Aktivitas ini terjadi di dapur dan bar (Laube, 2015).
2. Pemasaran
   Area ini menangani pemasaran sebuah restoran kepada publik. Aktivitas area ini seperti periklanan, promosi, dan penciptaan kesan yang baik (Laube, 2015).
3. Finansial
   Area ini menangani kegiatan fungsional yang terkait dengan akunting, pengelolaan kas, kontrol biaya selain itu juga pelaporan finansial dan operasional pada sebuah restoran (Laube, 2015).

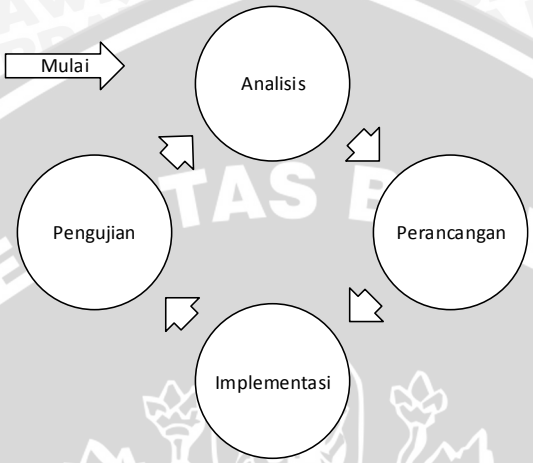
Prosedur pelayanan restoran yang masih memakai sistem yang manual atau tradisional dapat dilihat pada Gambar 2.1.



Gambar 2.1 Prosedur Pelayanan Restoran dengan Cara Manual
Sumber: (Tan & Chang, 2010)

2.3 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien dengan menggunakan mesin (Sukamto & Shalahuddin, 2013). Tahapan umum rekayasa tersebut dapat dilihat pada Gambar 2.2.

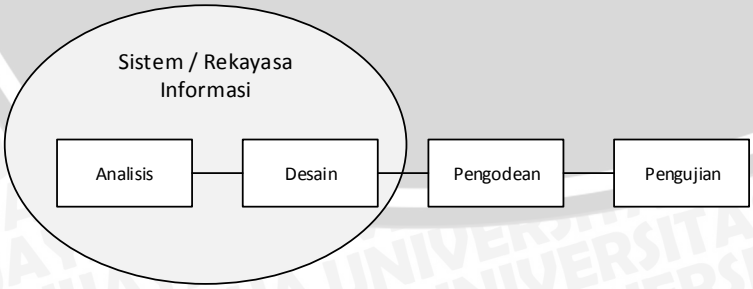


Gambar 2.2 Tahapan Umum Rekayasa Perangkat Lunak
Sumber: (Sukamto & Shalahuddin, 2013)

Proses perangkat lunak merupakan beberapa kumpulan aktivitas yang mempunyai tujuan untuk mengembangkan atau mengubah perangkat lunak. Secara umum proses perangkat lunak terdiri dari pengumpulan spesifikasi (*specification*), pengembangan (*development*), validasi (*validation*), dan evolusi (*evolution*) (Sukamto & Shalahuddin, 2013).

2.4 Model *Waterfall*

Salah satu model SDLC (*Software Development Lifecycle*) yang paling banyak digunakan adalah model *waterfall*. Model ini sering juga disebut sebagai model *sequential linier* atau alur hidup klasik (Sukamto & Shalahuddin, 2013). Gambar 2.3 mengilustrasikan tahapan pada model *waterfall*.



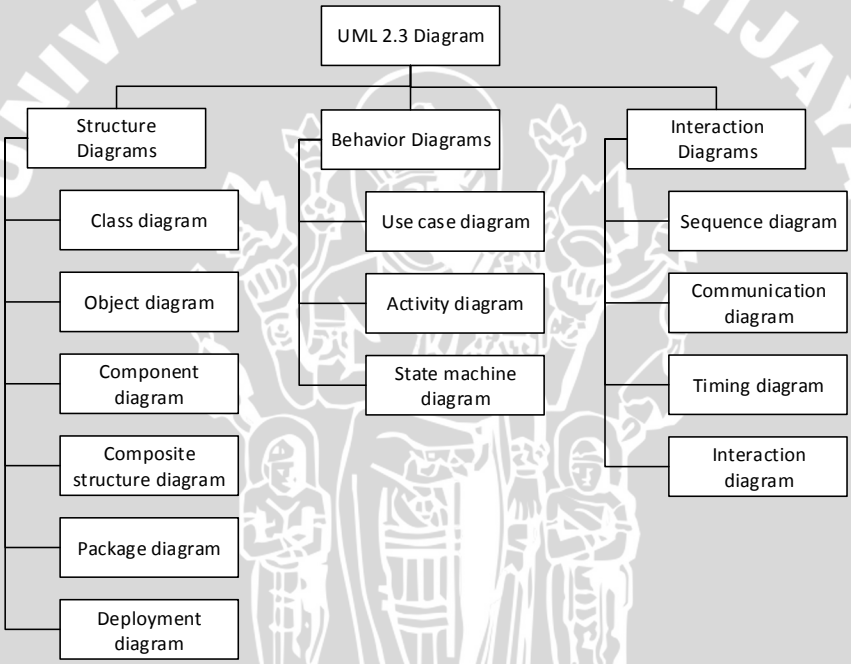
Gambar 2.3 Ilustrasi Model *Waterfall*
Sumber: (Sukamto & Shalahuddin, 2013)

Kelemahan yang ada pada model *waterfall* adalah ketika adanya perubahan spesifikasi. Perubahan tersebut sangat sulit dilakukan sebab tahapan pada proses

pengembangan ini bersifat statis. Oleh karena itu, model ini sangat cocok digunakan untuk pengembangan sistem dengan spesifikasi yang tidak berubah-ubah (Sukamto & Shalahuddin, 2013). Pada penelitian ini, sistem yang dikembangkan bersifat generik, di mana pada umumnya sistem tersebut memiliki spesifikasi yang sudah tetap di awal, sehingga model SDLC yang digunakan adalah model *waterfall*.

2.5 UML (*Unified Modeling Language*)

UML merupakan bahasa yang digunakan untuk pemodelan dan komunikasi mengenai sebuah sistem dengan diagram atau teks pendukung. UML 2.3 terdiri dari 13 jenis diagram yang dikelompokkan menjadi 3 kategori utama (Sukamto & Shalahuddin, 2013). Pembagian kategori dan jenis-jenis diagram tersebut dapat dilihat pada Gambar 2.4.



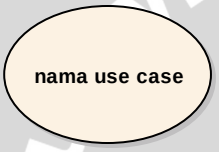
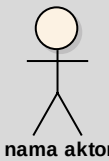
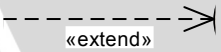
Gambar 2.4 Diagram UML (*Unified Modeling Language*)
Sumber: (Sukamto & Shalahuddin, 2013)

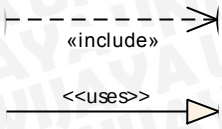
Gambar 2.4 menggambarkan struktur dari seluruh diagram UML, di mana masing-masing jenis diagram memiliki karakteristik tersendiri. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan struktur statis dari sistem. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem pada sebuah sistem. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem (Sukamto & Shalahuddin, 2013). Penelitian ini menggunakan empat jenis diagram dalam perancangan dan desain sistem yaitu diagram *use case,* diagram aktivitas, diagram kelas, dan diagram sekuen.

2.5.1 Diagram *Use Case*

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan sistem yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang berhak menggunakan fungsi itu. Dua hal yang harus diperhatikan dalam pembuatan *use case* yaitu pendefinisian aktor dan *use case*. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibuat di luar sistem yang akan dibuat itu sendiri. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor (Sukamto & Shalahuddin, 2013). Tabel 2.2 menggambarkan simbol-simbol yang ada pada diagram *use case*.

Tabel 2.2 Simbol-Simbol Diagram *Use Case*

Simbol	Deskripsi
Use case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama *use case.*
Aktor / *actor* 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
Asosiasi / *association* 	Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor.
Ekstensi / *extend* 	Relasi *use case* tambahan ke sebuah *use case* di mana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu. Biasanya *use case* tambahan memiliki nama depan yang sama dengan use case yang ditambahkan. Arah panah mengarah pada *use case* yang ditambahkan.
Generalisasi / *generalization* 	Hubungan generalisasi dan spesialisasi (umum - khusus) antara dua buah *use case* di mana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. Arah panah mengarah pada *use case* yang menjadi generalisasinya (umum).
Menggunakan / *include* / *uses*	Relasi *use case* tambahan ke sebuah *use case* di mana *use case* yang ditambahkan memerlukan *use*

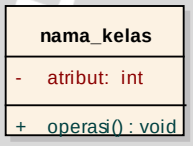
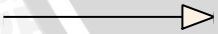
Simbol	Deskripsi
	*case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini.

Sumber: (Sukamto & Shalahuddin, 2013)

2.5.2 Diagram Kelas

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi adalah fungsi-fungsi yang dimiliki oleh suatu kelas (Sukamto & Shalahuddin, 2013). Tabel 2.3 menggambarkan simbol-simbol yang ada pada diagram kelas.

Tabel 2.3 Simbol-Simbol Diagram Kelas

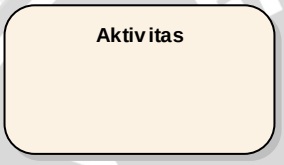
Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem.
Antarmuka / *interface* 	Sama seperti konsep *interface* dalam pemrograman berorientasi objek.
Asosiasi / *Association* 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*.
Asosiasi berarah / *directed association* 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*.
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
Kebergantungan / *dependency* 	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi / *aggregation* 	Relasi antar kelas dengan makna semua-bagian (*whole-part*).

Sumber: (Sukamto & Shalahuddin, 2013)

2.5.3 Diagram Aktivitas

Diagram aktivitas menggambarkan aktivitas dari sebuah sistem atau proses bisnis yang ada pada sistem. Dalam pemodelan diagram aktivitas yang perlu diperhatikan bahwa diagram ini menggambarkan aktivitas sistem bukan apa yang dilakukan aktor (Sukamto & Shalahuddin, 2013). Tabel 2.4 menggambarkan simbol-simbol yang ada pada diagram aktivitas atau *activity diagram*.

Tabel 2.4 Simbol-Simbol Diagram Aktivitas

Simbol	Deskripsi
Status awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / *decision*	Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu.
Penggabungan / *join*	Asosiasi penggabungan di mana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

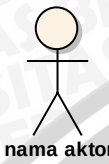
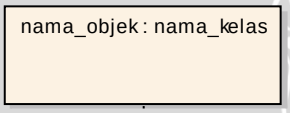
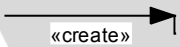
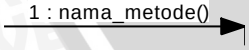
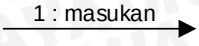
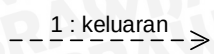
Sumber: (Sukamto & Shalahuddin, 2013)

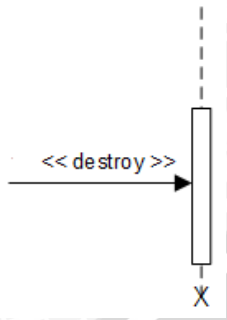
2.5.4 Diagram Sekuen

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri. Semua metode di dalam kelas harus ada di dalam diagram sekuen, jika tidak ada berarti perancangan metode di dalam kelas itu kurang baik. Hal ini dikarenakan ada

metode yang tidak dapat dipertanggungjawabkan kegunaannya (Sukamto & Shalahuddin, 2013). Tabel 2.5 menggambarkan simbol-simbol yang ada pada diagram sekuen.

Tabel 2.5 Simbol-Simbol Diagram Sekuen

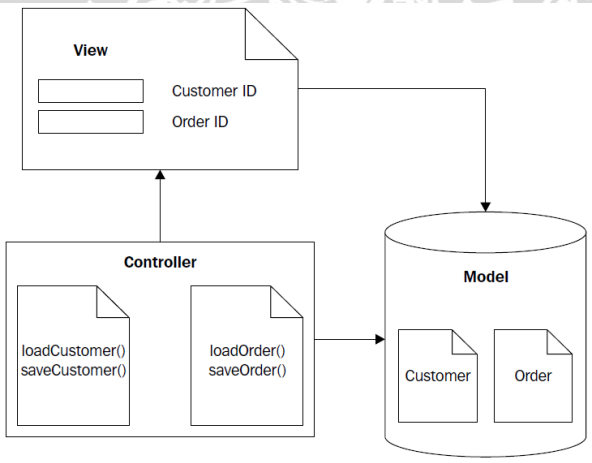
Simbol	Deskripsi
Aktor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
Garis hidup / *lifeline*	Menyatakan kehidupan suatu objek.
Objek 	Menyatakan objek yang berinteraksi pesan.
Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
Pesan tipe *create* 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe *All* 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi/metode tersebut.
Pesan tipe *send* 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirimi.
Pesan tipe *return* 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.

Simbol	Deskripsi
Pesan tipe *destroy* 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada *create* maka ada *destroy*.

Sumber: (Sukamto & Shalahuddin, 2013)

2.6 Pola Arsitektur MVC (*Model, View, Controller*)

MVC (*Model, View, Controller*) merupakan salah satu pola yang dikenal luas sebagai konsep yang melakukan pemisahan proses antara lapisan presentasi (antarmuka), logika bisnis, dan data pada satu aplikasi. MVC dapat mempermudah pekerjaan jika terdapat pengubahan, perbaikan, dan perkembangan dalam sebuah sistem (Abeysinghe, 2009). Gambar 2.5 merepresentasikan pola arsitektur MVC.



Gambar 2.5 Pola Arsitektur MVC (*Model, View, Controller*)
Sumber: (Abeysinghe, 2009)

Terdapat tiga komponen yang menunjukkan lapisan komponen dalam MVC yaitu:

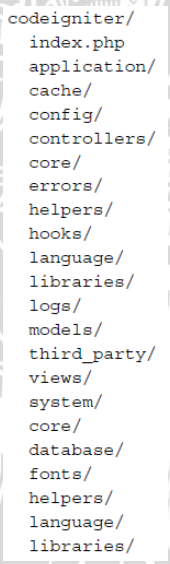
Model, merupakan komponen yang melakukan pengelolaan terhadap akses data yang terjadi pada aplikasi. *Model* bertugas terhadap respon jika *view* melakukan permintaan data dan respon jika *controller* melakukan perintah untuk pengubahan data (Abeysinghe, 2009).

View, merupakan komponen yang bertugas untuk menampilkan data dari *model* ke dalam bentuk interaksi antarmuka. Dengan kata lain, *view* melakukan pengelolaan terhadap informasi yang ditampilkan (Abeysinghe, 2009).

Controller, merupakan komponen yang bertugas melakukan respon dari perintah-perintah pengguna aplikasi seperti pemanggilan *view* atau pengubahan/permintaan data yang melibatkan pemanggilan *model*. Dalam konteks nyata, *controller* berisi kode yang terkait dengan logika proses bisnis aplikasi (Abeysinghe, 2009).

2.7 CodeIgniter

Salah satu *framework* PHP terkenal yang menerapkan pola arsitektur MVC adalah CodeIgniter. *Framework* sendiri merupakan sebuah kerangka kerja yang mempermudah pengembang dalam mengembangkan aplikasi sehingga dapat dilakukan pengubahan dalam aplikasinya dengan mudah dan dapat digunakan untuk aplikasi lainnya yang sejenis. *Framework* CodeIgniter dikembangkan dan dipelihara oleh Ellis Lab. Versi pertama CodeIgniter dirilis ke publik pada 28 Februari 2006. CodeIgniter tersedia pada laman http://codeigniter.com/downloads/ (Orr & Zadik, 2013) (Sukamto & Shalahuddin, 2013). Gambar 2.6 menunjukkan komponen-komponen yang ada pada *framework* CodeIgniter.



Gambar 2.6 Struktur Folder CodeIgniter
Sumber: (Orr & Zadik, 2013)

Terdapat beberapa komponen folder yang wajib diperhatikan yaitu:

1. *application/config*
   Folder ini berisi tentang konfigurasi dari CodeIgniter (Orr & Zadik, 2013).
2. *application/controller*
   Folder ini berisi semua kelas *controller* dari aplikasi. *Controller* pada folder ini memiliki hubungan generalisasi kepada kelas CI_Controller. Fungsi yang terdapat pada *controller* dapat dipanggil melalui URL (Orr & Zadik, 2013).
3. *application/view*

Folder ini berisi semua *file view*. Sebuah komponen *view* berisi konten HTML yang dieksekusi oleh *browser* pengguna aplikasi. Sebuah *view* dapat berupa halaman web atau halaman RSS (Orr & Zadik, 2013).

4. *application/model*
   Folder ini berisi semua kelas *model* dari aplikasi. *Model* pada folder ini memiliki hubungan generalisasi kepada kelas CI_Model (Orr & Zadik, 2013).

2.8 Pengujian Integrasi

Pengujian integrasi atau *integration testing* merupakan fase pengujian perangkat lunak yang mana setiap kelas atau komponen dalam perangkat lunak dikombinasikan dan diuji menjadi satu kesatuan. Pengujian ini bertujuan untuk memastikan hubungan antar kelas bekerja sesuai dengan spesifikasi kebutuhan perangkat lunak. Untuk melakukan pengujian integrasi, penguji dapat menggunakan teknik pengujian *Black Box* (Taiwar, 2013).

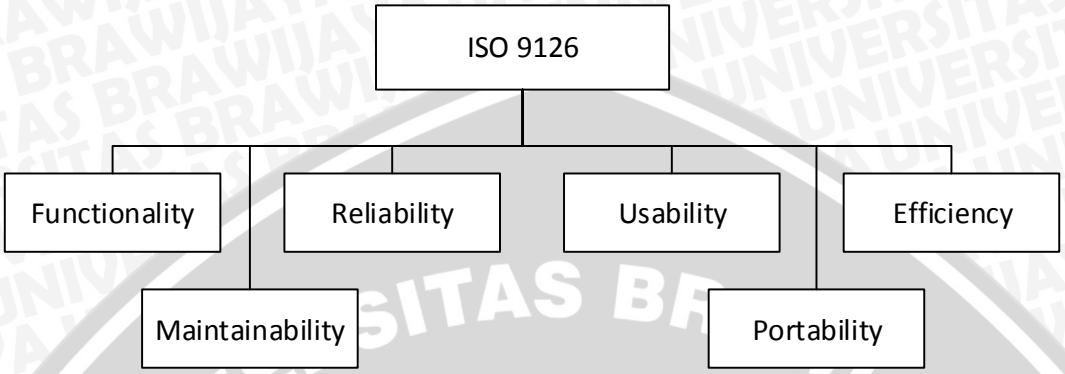
Terdapat empat jenis pendekatan dalam pengujian integrasi yaitu *Big-Bang, Top-Down, Bottom-Up,* dan *Sandwich.* Pendekatan *Big-Bang* melakukan pengujian terhadap semua atau sebagian besar kelas yang sudah diimplementasikan secara bersamaan. Pendekatan ini sangat efektif untuk menghemat waktu dalam pengujian integrasi. Namun, jika kasus uji dan hasilnya tidak didokumentasikan dengan baik maka proses integrasi keseluruhan akan lebih rumit. Sehingga pendekatan ini cocok untuk sistem yang tidak terlalu besar. Sedangkan pada pendekatan *Top-Down*, kelas dengan level tertinggi diujikan terlebih dahulu dan bertahap turun kepada kelas dengan level terendah. Lalu pada pendekatan *Bottom-Up*, pengujian dimulai dengan pengujian kelas secara individu dan bertahap naik kepada beberapa kelas. Dan pada pendekatan *Sandwich*, pengujian integrasi dilakukan berdasarkan kombinasi dari pendekatan *Top-Down* dan *Bottom-Up* (Taiwar, 2013).

2.9 ISO 9126

ISO 9126 merupakan standar internasional yang digunakan untuk mengevaluasi kualitas sebuah produk perangkat lunak. Standar ini dibagi menjadi empat bagian yaitu *quality model* (ISO 9126-1)*, external metrics* (ISO 9126-2)*, internal metrics* (ISO 9126-3)*,* dan *quality in use metrics* (ISO 9126-4). ISO 9126-1 (*quality model*) merupakan pengembangan dari standar kualitas sebelumnya seperti McCall (1977), Boehm (1978), FURPS, dan lainnya (Jamwal, 2010).

Model kualitas ini memiliki enam karakteristik utama yang terbagi menjadi dua jenis yaitu kualitas internal dan eksternal. *Functionality, reliability, usability,* dan *efficiency* merupakan kualitas eksternal sedangkan *maintainability,* dan *portability* masuk ke dalam kualitas internal. Kualitas internal merupakan karakteristik kualitas produk perangkat lunak yang dilihat dari pandangan internal di mana jenis kualitas ini dapat dievaluasi oleh pihak internal seperti pengembang. Sedangkan kualitas eksternal merupakan karakteristik kualitas produk perangkat lunak yang

dilihat dari pandangan eksternal di mana jenis kualitas ini dapat dievaluasi ketika produk telah dioperasikan oleh konsumen (Al-Safadi & Garcia, 2012) (International Standards Office, 2000). Gambar 2.7 merepresentasikan enam karakteristik tersebut.



Gambar 2.7 Karakteristik Utama Model Kualitas ISO 9126
Sumber: (International Standards Office, 2000)

2.9.1 *Functionality*

Merupakan kemampuan sebuah produk perangkat lunak untuk dapat menyediakan fitur-fitur yang dibutuhkan, sehingga karakteristik ini fokus pada apakah perangkat lunak tersebut telah memenuhi kebutuhan konsumen (International Standards Office, 2000). Karakteristik ini dapat dievaluasi dengan pengujian terhadap fungsi-fungsi yang ada pada perangkat lunak dalam bentuk kasus uji (Setialana, 2014). Menurut ISO (2002), suatu perangkat lunak dapat diketahui tingkat fungsionalitasnya dengan Persamaan 2.1. Perangkat lunak dikatakan memenuhi standar jika hasil dari persamaan tersebut mendekati angka 1 (International Standards Office, 2002).

$$X = 1 - \frac{A}{B} \quad (2.1)$$

Keterangan:

X = Tingkat *functionality*.

A = Jumlah fungsi yang bermasalah.

B = Jumlah fungsi yang diuji keseluruhan.

Contoh:

Jika hasil pengujian kotak hitam menghasilkan 2 buah SKPL (Spesifikasi Kebutuhan Perangkat Lunak) yang belum valid dari total 40 pengujian. Maka didapatkan nilai A = 2 dan B = 40, sehingga dapat dihitung dengan cara berikut:

$$X = 1 - \frac{2}{40} = 1 - 0{,}05 = 0{,}95$$

Maka didapatkan nilai X (*functionality*) sangat mendekati 1 sehingga hasil tersebut memenuhi standar.

2.9.2 *Reliability*

Merupakan kemampuan sebuah produk perangkat lunak untuk dapat menjaga kinerjanya ketika digunakan dalam kondisi tertentu (International Standards Office, 2000). Karakteristik ini dapat dievaluasi dengan menggunakan *tool* WAPT (*Web Application Load, Stress, and Performance Testing*). WAPT (*Web Application Performance Tools*) menyediakan pengujian *performance*, *stress* dan *load* pada sebuah situs atau aplikasi web melalui *interface* web. *Tool* ini dapat digunakan untuk mengetahui kestabilan perangkat lunak pada tingkat tertentu dengan menjalankan *load agent* secara bersamaan, di mana *load agent* merupakan *virtual user* yang ukurannya telah ditetapkan oleh standar dalam WAPT (Rina & Tyagi, 2013). Kriteria yang diuji pada karakteristik *reliability* dengan WAPT yaitu *sessions*, *pages*, dan *hits* (Setialana, 2014). Menurut ISO (2002), karakteristik *reliability* dapat dihitung dengan Persamaan 2.2. Hasil dari persamaan tersebut dibandingkan dengan standar Telcordia yaitu 95% dari kasus uji harus berhasil (Asthana & Olivieri, 2009).

$$X = \frac{A}{B} \quad (2.2)$$

Keterangan:

X = Nilai *reliability*.

A = Jumlah berhasil dalam pengujian.

B = Jumlah pengujian keseluruhan.

Contoh:

Jika hasil pengujian dengan WAPT menghasilkan 217 *sessions* berhasil dari 227 kasus uji, 25687 *pages* berhasil dari 25787 kasus uji, dan 16460 *hits* berhasil dari 16519 kasus uji maka dapat dihitung dengan cara berikut:

1. Session: $X = \frac{217}{227} = 0{,}956$
2. Pages: $X = \frac{25687}{25787} = 0{,}996$
3. Hits: $X = \frac{16460}{16519} = 0{,}996$
4. Total: $X = \frac{0{,}956+0{,}996+0{,}996}{3} x100\% = 98{,}26\%$

Maka didapatkan nilai X (*reliability*) sebesar 98,26% yang mana hasil tersebut memenuhi standar Telcordia yaitu di atas 95%.

2.9.3 *Usability*

Merupakan kemampuan sebuah produk perangkat lunak untuk dapat dimengerti, dipelajari, digunakan dan menarik kepada pengguna ketika digunakan dalam kondisi tertentu (International Standards Office, 2000). Karakteristik ini dapat dievaluasi dengan berbagai macam model kuesioner. Menurut Tullis & Stetson (2004), kuesioner SUS (*System Usability Scale*) memiliki pertanyaan yang dapat diandalkan dibandingkan dengan model kuesioner yang lain sebab SUS

merupakan satu-satunya kuesioner yang melibatkan seluruh aspek yang berbeda pada interaksi pengguna terhadap sistem (Tullis & Stetson, 2004). Perhitungan nilai *usability* pada kuesioner SUS menggunakan skala Likert. Skala Likert mengindikasikan persetujuan dan ketidaksetujuan sebuah pernyataan dengan bobot skala 1-5. Skor akhir kuesioner SUS memiliki rentang nilai antara 0-100. Aturan perhitungan skor kuesioner SUS yaitu (Broke, 1996):

1. Kuesioner ganjil dihitung dengan Persamaan 2.3:
   $$Skor\ Ganjil = Bobot\ Jawaban - 1 \quad (2.3)$$
2. Kuesioner genap dihitung dengan Persamaan 2.4:
   $$Skor\ Genap = 5 - Bobot\ Jawaban \quad (2.4)$$
3. Skor akhir dihitung dengan Persamaan 2.5:
   $$Skor\ Akhir = Jumlah\ Skor\ x\ 2{,}5 \quad (2.5)$$

2.9.4 *Efficiency*

Merupakan kemampuan sebuah produk perangkat lunak untuk dapat menyediakan kinerja yang sesuai dengan *resource* yang dipakai (International Standards Office, 2000). Karakteristik ini dapat dievaluasi dengan menggunakan *tools* WAPT dengan kriteria waktu respon (*response time*) (Setialana, 2014). Hasil dari kriteria waktu respon tersebut akan dibandingkan dengan standar Jakob Nielsen yang dapat dilihat pada Tabel 2.6.

Tabel 2.6 Standar Jakob Nielsen

Waktu Respon	Pandangan Pengguna
< 0,1 detik	Pengguna merasakan respon yang sangat cepat dari web.
< 1,0 detik	Pengguna merasakan adanya jeda namun masih fokus dalam web tersebut.
< 10 detik	Jika waktu respon web mendekati 10 detik, penelitian membuktikan bahwa perhatian pengguna terhadap web tersebut akan menurun tajam.
≥ 10 detik	Kemungkinan besar pengguna akan beralih dari web tersebut.

Sumber: (Nielsen, 1993)

2.9.5 *Maintainability*

Merupakan kemampuan sebuah produk perangkat lunak untuk dapat dimodifikasi, di mana modifikasi tersebut dapat berupa perbaikan atau adaptasi perangkat lunak terhadap perubahan lingkungan atau spesifikasi kebutuhan (International Standards Office, 2000). Karakteristik *maintainability* dapat dievaluasi dengan menggunakan persamaan MI (*Maintainability Index*) (Coleman, et al., 1994). Perhitungan MI dapat menggunakan Persamaan 2.6.

$$MI = 171 - 5{,}2\ x \ln(aveV) - 0{,}23\ x\ aveV(g) - 16{,}2\ x \ln(aveLOC) \quad (2.6)$$

Keterangan:

MI = Maintainability Index.

aveV = Rata-rata *Halstead Volume*.

aveV(g) = Rata-rata *Cyclomatic Complexity*.

aveLOC = Rata-rata *Line of Code*.

Untuk dapat mempermudah pencarian elemen MI seperti aveV, aveV(g), dan aveLOC dapat menggunakan *tools* DMSSearchEngine. Perangkat lunak dapat memenuhi karakteristik *maintainability* apabila memiliki nilai MI di atas 65 yaitu pada kategori *Medium* MI dan *High* MI (Setialana, 2014). Standar MI dapat dilihat pada Tabel 2.7.

Tabel 2.7 Standar *Maintainability Index* (MI)

Nilai MI	Kategori	Sifat
≥ 85	*High* MI	*Highly Maintainable*
65 – 84,99	*Medium* MI	*Moderate Maintainable*
< 65	*Low* MI	*Difficult to Maintain*

Sumber: (Coleman, et al., 1994)

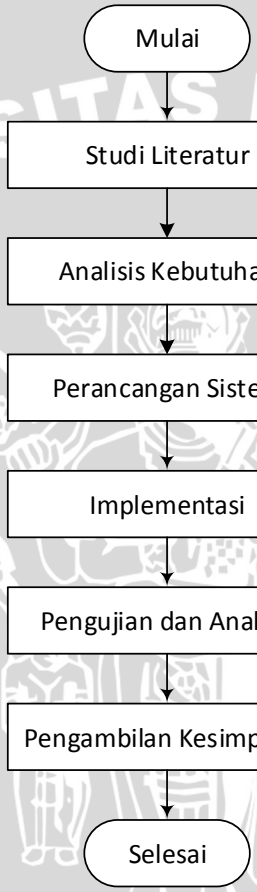
2.9.6 *Portability*

Merupakan kemampuan sebuah produk perangkat lunak untuk dapat dipindahkan dalam lingkungan yang berbeda (International Standards Office, 2000). Karakteristik *portability* sebuah perangkat lunak berbasis web dapat dinilai melalui kompatibilitas terhadap 80% *desktop browser* dan juga 80% *mobile browser* yang digunakan oleh sebagian besar masyarakat di dunia. Maka dari itu, diperlukan uji *portability* perangkat lunak berbasis web tersebut pada tujuh *desktop browser* dan lima *mobile browser* (Salonen, 2012).

Pengujian terhadap berbagai jenis *web browser* ini dapat menggunakan sebuah *tools* BrowseEmAll. BrowseEmAll dapat menguji *portability* sebuah perangkat lunak berbasis web dengan menjalankan berbagai *virtual browser* dengan melakukan pengujian antarmuka sistem baik resolusi, CSS maupun JavaScript pada sistem tersebut (Setialana, 2014). BrowseEmAll dapat memperlihatkan *error* dan *warning* dari antarmuka sistem yang diuji. Maka dari itu, pemenuhan karakteristik *portability* dapat dilihat dengan tidak adanya *error* maupun *warning* dari sistem.

BAB III
METODOLOGI PENELITIAN

Bab ini membahas metode yang digunakan dalam penelitian yang terdiri dari beberapa tahapan. Beberapa tahapan tersebut yaitu studi literatur, analisis kebutuhan, perancangan sistem, implementasi, pengujian dan analisis, serta pengambilan kesimpulan dan saran. Gambar 3.1 merepresentasikan alur kerja dari tahapan yang telah diuraikan sebelumnya.



Gambar 3.1 Diagram Alir Tahapan Penelitian

3.1 Studi Literatur

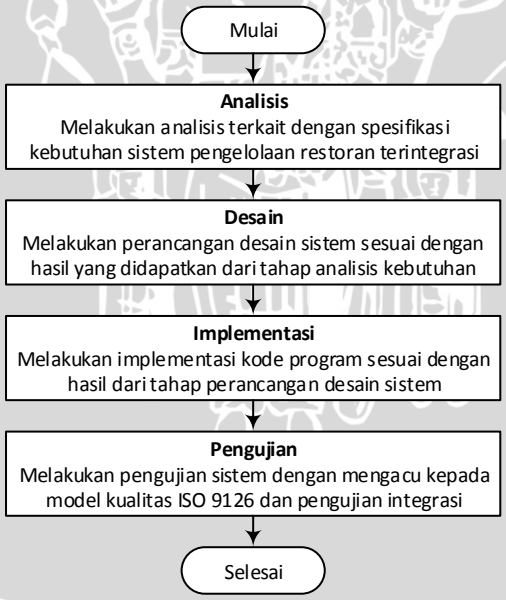
Studi literatur memaparkan kajian pustaka dan dasar teori yang mendukung dilakukannya penelitian ini. Pustaka dan teori yang terkait dengan penelitian ini meliputi:

1. Restoran
2. Rekayasa Perangkat Lunak
3. Model *Waterfall*
4. UML (*Unified Modeling Language*)
   a. Diagram *Use Case*

b. Diagram Kelas
c. Diagram Aktivitas
d. Diagram Sekuen

5. Pola Arsitektur MVC (Model, View, Controller)
6. CodeIgniter
7. Pengujian Integrasi
8. ISO 9126
   a. Functionality
   b. Reliability
   c. Usability
   d. Efficiency
   e. Maintainability
   f. Portability

Model pengembangan sistem yang digunakan pada penelitian ini adalah model *waterfall*. Pemilihan model pengembangan tersebut didasarkan karena kebutuhan sistem yang dibangun relatif stabil dan sudah tetap di tahap awal pembangunan sehingga cocok terhadap model *waterfall* yang harus meminimalisir perubahan spesifikasi kebutuhan. Tahapan pengembangan yang dilakukan berdasarkan model *waterfall* pada penelitian ini dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram Alir Pengembangan Sistem dengan Model *Waterfall*

Pengembangan sistem dengan model *waterfall* pada penelitian ini melibatkan empat tahap yaitu:

1. Analisis

   Pada tahap ini, spesifikasi kebutuhan dari sistem pengelolaan restoran terintegrasi dianalisis dengan mengkaji penelitian-penelitian terdahulu. Metode tersebut dipilih sebab proses, fitur dan bagian sistem di sebagian

besar restoran dapat ditemukan lebih jelas dan efisien dibandingkan dengan survei pada banyak restoran. Sehingga, teknik yang digunakan dalam tahap ini adalah melakukan studi literatur. Hasil dari tahap ini berupa spesifikasi kebutuhan perangkat lunak yang disertai dengan identifikasi aktor pada sistem dan diagram *use case* beserta skenarionya.

2. Desain
   Pada tahap desain, sistem akan dirancang berdasarkan hasil yang didapatkan pada tahap analisis. Hasil dari tahap ini berupa desain arsitektural, desain diagram kelas, diagram aktivitas, diagram sekuen, dan desain antarmuka sistem.
3. Implementasi
   Pada tahap ini, perancangan sistem yang telah dibuat pada tahap desain akan diubah menjadi kode program. Kode program tersebut diimplementasikan dengan bahasa PHP pada sisi *back end* dan sintaks HTML, CSS, dan Javascript pada sisi *front end*. Hasil dari tahap ini berupa sistem yang sesuai dengan rancangan sistem pada tahap desain.
4. Pengujian
   Pada tahap pengujian, sistem akan diuji menggunakan model kualitas ISO 9126 dan pengujian integrasi. Sistem diuji hanya pada enam karakteristik utama ISO 9126 yaitu *functionality*, *reliability*, *usability*, *efficiency*, *maintainability*, dan *portability*. Hasil dari tahap ini berupa sistem yang telah memenuhi standar dari model kualitas ISO 9126.

3.2 Analisis Kebutuhan

Analisis kebutuhan yang dilakukan pada penelitian ini mengacu kepada penelitian-penelitian terdahulu yang terkait dengan pengembangan sistem pengelolaan restoran terintegrasi. Sehingga teknik yang dilakukan dalam pengumpulan data menggunakan teknik studi literatur atau *literature review*. Literatur yang diacu terdiri dari empat penelitian yang telah disebutkan pada bagian kajian pustaka yaitu "*Real-Time Process Management System in a Restaurant by Sharing Food Order Information*" oleh Shimmura, et al. (2009), "*Intelligent Restaurant System Smartmenu*" oleh Pieskä, et al. (2013), "*Development and Evaluation of an RFID-based e-Restaurant System for Customer-centric Service*" oleh Tan & Chang (2010), dan "*Development of an RFID-based Sushi Management System*" oleh Ngai, et al. (2008). Pengacuan terhadap empat penelitian tersebut ditujukan untuk mendapatkan fungsi-fungsi yang umum diimplementasikan pada sistem pengelolaan restoran.

Berdasarkan acuan dari penelitian yang telah disebutkan tersebut, maka kebutuhan fungsional untuk kegiatan operasional restoran yang akan dibangun pada penelitian ini meliputi:

1. Area fungsional pemasaran bagian POS (*Point of Sales*)

a. Bagian ini harus memiliki fungsi untuk memesan makanan serta melihat status pesanan makanan tersebut secara *real-time* yang dilakukan oleh pelayan (Shimmura, et al., 2009) (Pieskä, et al., 2013) (Tan & Chang, 2010).
b. Bagian ini dapat mengelola dan mencetak pembayaran tagihan menu makanan yang dilakukan oleh kasir (Pieskä, et al., 2013) (Ngai, et al., 2008).

2. Area fungsional operasional bagian dapur
   a. Bagian ini harus memiliki fungsi untuk menampilkan daftar pesanan tamu yang disertai jumlah total pesanan, nomor meja, dan waktu penyiapan menu pesanan secara *real-time* untuk koki (Shimmura, et al., 2009) (Pieskä, et al., 2013).
   b. Bagian ini harus menyediakan fitur untuk memberikan notifikasi kepada koki jika penyiapan pesanan melebihi batas waktu yang ditentukan (Shimmura, et al., 2009) (Tan & Chang, 2010).
   c. Bagian ini harus memiliki fungsi untuk mencari dan mencetak menu pesanan ulang oleh koki jika lembar pesanan hilang (Shimmura, et al., 2009).
   d. Bagian ini harus memiliki fungsi untuk memvalidasi dan mencetak lembar pesanan yang telah disiapkan oleh koki (Shimmura, et al., 2009) (Pieskä, et al., 2013).

3. Area pendukung manajerial
   a. Bagian ini dapat membantu manajer untuk mengelola menu makanan dan minuman yang tersedia pada restoran (Pieskä, et al., 2013) (Ngai, et al., 2008).
   b. Bagian ini dapat membantu manajer untuk mengelola informasi restoran dan pengguna sistem (Pieskä, et al., 2013) (Ngai, et al., 2008).
   c. Bagian ini dapat membantu manajer untuk melihat histori transaksi kasir (Ngai, et al., 2008).
4. Sistem harus menyediakan fitur autentikasi terhadap pengguna dengan tujuan agar hanya pengguna yang dikenali oleh sistem yang dapat mengoperasikannya.

Sedangkan kebutuhan non fungsional pada penelitian ini mengacu kepada model kualitas ISO 9126 yang meliputi:

1. *Reliability*, sistem dapat melayani perintah pengguna dengan batas toleransi *error* hingga 5%.
2. *Usability,* sistem dapat digunakan dengan mudah oleh pengguna.
3. *Efficiency,* sistem dapat melayani perintah pengguna dengan batas waktu respon di bawah 10 detik.
4. *Maintainability,* sistem dapat dipelihara dengan mudah yang ditunjukkan dengan nilai MI (*Maintainability Index*) lebih dari sama dengan 65.
5. *Portability,* sistem dapat dijalankan di berbagai *web browser* yang berbeda.

Setelah melakukan identifikasi kebutuhan fungsional dan non fungsional maka dapat dilakukan identifikasi aktor. Ketiga elemen tersebut digunakan sebagai acuan pemodelan sistem dengan diagram *use case* beserta skenario *use case* sehingga dapat dihasilkan gambaran eksternal sebuah sistem ini.

3.3 Perancangan Sistem

Perancangan sistem pada penelitian ini dibuat berdasarkan hasil pemodelan diagram *use case* beserta skenario *use case* yang telah dilakukan pada tahap analisis kebutuhan. Kedua elemen tersebut digunakan sebagai acuan dalam pemodelan diagram kelas, diagram aktivitas, dan diagram sekuen. Pada penelitian ini, perancangan sistem dibagi menjadi beberapa tahap dengan penjelasan sebagai berikut:

1. Pada tahap awal, akan dirancang desain arsitektural dari sistem. Di mana arsitektur yang digunakan adalah model *client-server.*
2. Pada tahap kedua, akan dirancang diagram kelas yang akan dibagi menjadi dua tahap pengerjaan. Pertama diagram kelas level analisis akan dimodelkan lalu diikuti dengan pemodelan diagram kelas level implementasi. Diagram kelas level analisis digunakan untuk merepresentasikan atribut data dan interaksi yang digunakan dalam sistem sedangkan diagram kelas level implementasi digunakan untuk merepresentasikan struktur dari kelas yang diimplementasikan dalam sistem. Atribut pada diagram kelas level analisis didapatkan dari pengamatan dokumen lembar pesanan dan tagihan pembayaran pada beberapa restoran yang dapat dilihat pada bagian Lampiran. Kemudian, diagram kelas level implementasi dibagi menjadi dua jenis yaitu *model* dan *controller* sebab sistem yang dikembangkan menggunakan pola arsitektur MVC (*Model, View, Controller*).
3. Pada tahap ketiga, akan dirancang basis data dengan menggunakan ERD (*Entity Relationship Diagram*) yang mengacu kepada diagram kelas level analisis.
4. Pada tahap keempat akan dibuat diagram aktivitas yang mengacu kepada skenario *use case*.
5. Pada tahap kelima, diagram sekuen akan dibuat dengan mengacu kepada diagram aktivitas. Operasi yang telah didefinisikan pada diagram kelas digunakan sebagai acuan dalam pembuatan diagram sekuen.
6. Pada tahap akhir perancangan sistem akan dilakukan pemodelan antarmuka untuk interaksi aktor pelayan, kasir, koki dan manajer terhadap sistem.

3.4 Implementasi

Implementasi sistem yang dilakukan pada penelitian ini mengacu kepada perancangan sistem. Implementasi sistem ini meliputi:

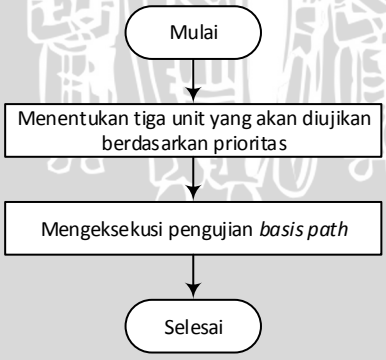
1. Pembuatan kode *front end* yang berupa sintaks-sintaks HTML, CSS, dan JavaScript dengan *framework* Bootstrap serta gambar-gambar komponen antarmuka untuk implementasi antarmuka pengguna. *Framework* Bootstrap dipilih agar antarmuka yang dibuat dapat menyesuaikan dengan perangkat yang dimiliki aktor pelayan, kasir, koki dan manajer.
2. Pembuatan basis data pada MySQL *database server* untuk implementasi skema basis data yang telah dirancang.
3. Pembuatan kode *back end* yang berupa bahasa pemrograman PHP dengan *framework* CodeIgniter. *Framework* CodeIgniter dipilih karena sistem ini menggunakan pola arsitektur MVC (*Model, View, Controller*).

3.5 Pengujian dan Analisis

Setelah tahap implementasi selesai dilakukan maka sistem siap diuji. Pengujian sistem pada penelitian ini dibagi menjadi dua tahap utama. Tahap pertama sistem diuji menggunakan paradigma rekayasa perangkat lunak yaitu pengujian unit, pengujian integrasi dan pengujian validasi. Dan pada tahap kedua akan dilakukan pengujian dengan model kualitas ISO 9126.

3.5.1 Pengujian Unit

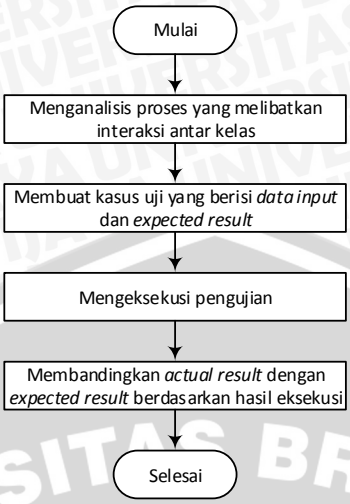
Pengujian unit pada penelitian ini dilakukan untuk memastikan unit-unit yang memiliki prioritas tinggi memiliki hasil implementasi yang sesuai dengan harapan di awal analisis kebutuhan. Pengujian unit yang dilakukan dengan menggunakan teknik *white box* dengan jenis pengujian *basis path*. Unit yang akan diujikan pada pengujian ini akan dipilih berdasarkan tiga unit dengan prioritas tinggi. Tahapan yang dilakukan pada pengujian unit dapat dilihat pada Gambar 3.3.



Gambar 3.3 Tahapan Pengujian Unit

3.5.2 Pengujian Integrasi

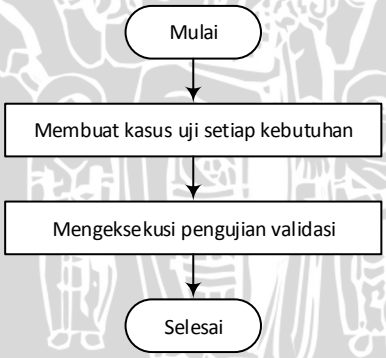
Pengujian integrasi yang dilakukan pada penelitian ini menggunakan pendekatan *Big-Bang* di mana kelas yang diujikan adalah kelas-kelas yang saling berinteraksi. Pada pengujian ini, teknik yang digunakan adalah teknik *black box*. Tahapan yang dilakukan pada pengujian integrasi ini dapat dilihat pada Gambar 3.4.



Gambar 3.4 Tahapan Pengujian Integrasi

3.5.3 Pengujian Validasi

Pengujian validasi pada penelitian ini dilakukan untuk memastikan seluruh spesifikasi kebutuhan perangkat lunak telah diimplementasikan sesuai dengan yang diharapkan. Pada pengujian ini, teknik yang digunakan adalah teknik *black box*. Tahapan yang dilakukan pada pengujian validasi dapat dilihat pada Gambar 3.5.



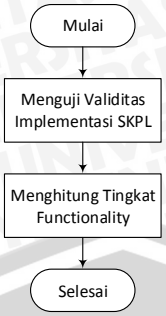
Gambar 3.5 Tahapan Pengujian Validasi

3.5.4 Pengujian Berbasis ISO 9126

Tahap awal pengujian pada penelitian ini mengacu kepada karakteristik utama yang ada pada model kualitas ISO 9126. Karakteristik atau kriteria yang diuji meliputi *functionality, realiability, usability, efficiency, maintainability, portability*.

3.5.4.1 *Functionality*

Pada pengujian karakteristik *functionality* semua implementasi dari spesifikasi kebutuhan perangkat lunak akan dinilai valid atau tidak. Setelah itu akan dihitung tingkat *functionality* berdasarkan aturan dari ISO. Tahapan yang dilakukan dapat dilihat pada Gambar 3.6.



Gambar 3.6 Tahapan Uji Karakteristik *Functionality*

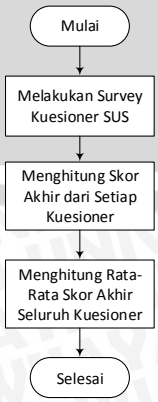
3.5.4.2 *Reliability*

Pengujian karakteristik *reliability* pada penelitian ini fokus pada kriteria *sessions, pages,* dan *hits* dengan menggunakan *tool* WAPT (*Web Application Performance Tools*). Hasil dari pengujian dibandingkan dengan standar Telcordia. Gambar 3.7 merepresentasikan tahapan dalam bentuk diagram alir.

Gambar 3.7 Tahapan Uji Karakteristik *Reliability*

3.5.4.3 *Usability*

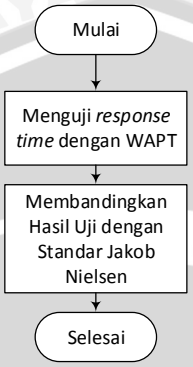
Pengujian karakteristik *usability* pada penelitian ini menggunakan model kuesioner SUS (*System Usability Scale*) yang terdiri dari sepuluh butir pernyataan. Setiap kuesioner yang telah dijawab oleh responden akan dihitung berdasarkan aturan kuesioner SUS. Lalu akan dihitung rata-rata dari keseluruhan skor setiap kuesioner. Gambar 3.8 memperlihatkan tahapan pada pengujian karakteristik *usability*.



Gambar 3.8 Tahapan Uji Karakteristik *Usability*

3.5.4.4 *Efficiency*

Pengujian karakteristik *efficiency* dilakukan untuk menilai waktu respon (*response time*) sistem dengan menggunakan *tool* WAPT. Hasil dari pengujian dibandingkan dengan standar Jakob Nielsen. Gambar 3.9 merepresentasikan tahapan pada pengujian karakteristik ini.



Gambar 3.9 Tahapan Uji Karakteristik *Efficiency*

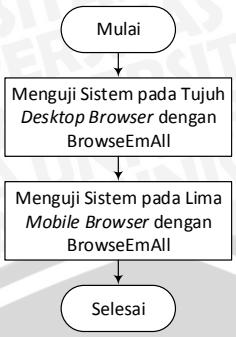
3.5.4.5 *Maintainability*

Pengujian karakteristik *maintainability* pada penelitian ini menggunakan persamaan MI (*Maintainability Index*). Hasil dari perhitungan MI akan dibandingkan dengan standar MI. Gambar 3.10 memperlihatkan alur dari pengujian karakteristik *maintainability*.

Gambar 3.10 Tahapan Uji Karakteristik *Maintainability*

3.5.4.6 *Portability*

Pengujian pada karakteristik *portability* pada penelitian ini dilakukan pada tujuh jenis *dekstop browser* dan lima jenis *mobile browser* dengan menggunakan *tool* BrowseEmAll. Sistem memenuhi *portability* bila tidak ditemukan *error* pada hasil pengujian menggunakan *tool* BrowseEmAll. Gambar 3.11 memperlihatkan tahapan dari pengujian karakteristik *portability*.



Gambar 3.11 Tahapan Uji Karakteristik *Portability*

3.6 Pengambilan Kesimpulan dan Saran

Tahap akhir dalam penelitian ini adalah pengambilan kesimpulan. Tahap ini dilakukan setelah tahapan sebelumnya yaitu dari analisis kebutuhan hingga pengujian selesai. Tahap ini terdiri dari dua bagian yaitu kesimpulan dan saran. Kesimpulan digunakan untuk menjawab rumusan masalah yang telah diuraikan sebelumnya. Sedangkan saran digunakan sebagai perbaikan kesalahan-kesalahan yang terjadi dan penyempurnaan penulisan serta pertimbangan atas pengembangan sistem selanjutnya. Pengambilan kesimpulan dan saran tersebut mengacu kepada hasil dari pengujian dan analisis.

BAB IV
PERANCANGAN DAN IMPLEMENTASI

Bab ini memaparkan perancangan dan implementasi dari sistem pengelolaan restoran terintegrasi. Perancangan yang dilakukan meliputi dua tahap utama. Tahap pertama yang dilakukan adalah analisis kebutuhan. Kemudian tahap kedua yaitu perancangan sistem. Sedangkan pada tahap implementasi terdiri dari pembahasan terkait spesifikasi lingkungan implementasi, batasan-batasan dalam implementasi, implementasi basis data, tiap kelas dalam *file* program, dan algoritma sistem, serta implementasi antarmuka sistem.

4.1 Gambaran Umum Sistem

Sistem pengelolaan restoran terintegrasi merupakan sistem yang dapat membantu kegiatan operasional di area fungsional yang berbeda pada restoran. Sistem ini mencakup dua area fungsional pada restoran yaitu pemasaran dan operasional. Kedua area tersebut saling bertukar informasi sehingga bagian dalam sistem ini terintegrasi. Nama sistem pengelolaan restoran terintegrasi ini adalah OtseResto. Kata Otser sendiri diambil dari kebalikan kata "resto" yang terinspirasi dari bahasa *walikan* yang populer di daerah Malang. Gambar 4.1 merepresentasikan logo dari sistem yang dikembangkan.

Gambar 4.1 Logo OtseResto

Sistem ini berbasis web sehingga pemesanan makanan dapat dilakukan secara nirkabel dari perangkat bergerak dalam hal ini perangkat tablet yang disediakan di meja makan. Di samping itu, *web server* dari OtseResto ini dioperasikan pada jaringan lokal agar waktu respon lebih cepat dan stabil dibandingkan dengan penggunaan *web server* di jaringan luar.

Sistem yang dikembangkan ini bersifat generik. Di mana sistem tersebut tidak memiliki konsumen yang spesifik. OtseResto sendiri memiliki tiga bagian utama yaitu area POS (*Point of Sale*) untuk pelayan dan kasir, area dapur untuk koki, dan area manajerial untuk manajer atau pemilik restoran.

1. Area POS (*Point of Sale*)

   Area POS merupakan area yang berfungsi untuk mendukung kegiatan penjualan pada restoran. Area ini terletak pada area fungsional pemasaran. Area POS sendiri memiliki dua area yaitu pemesanan, dan pembayaran. Area pemesanan digunakan oleh pelayan untuk melayani tamu dalam memilih

menu pesanan. Sedangkan area pembayaran digunakan oleh kasir untuk memproses tagihan menu pesanan tamu.

2. Area Dapur

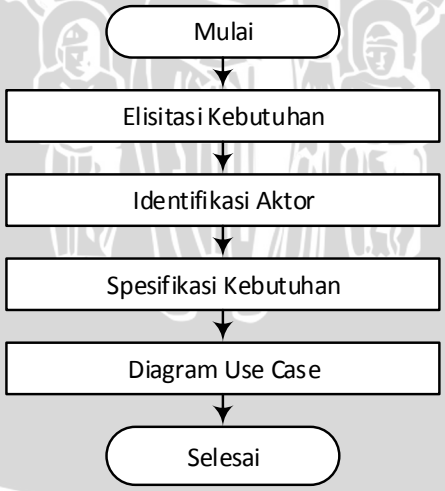
   Area dapur merupakan area yang berfungsi untuk mendukung kegiatan penyiapan menu pesanan tamu. Area ini terletak pada area fungsional operasional. Area dapur sendiri digunakan oleh koki untuk melihat daftar menu pesanan tamu.

3. Area Manajerial

   Area manajerial merupakan area yang berfungsi untuk mendukung pengelolaan informasi level atas restoran. Informasi tersebut berupa daftar menu makanan dan minuman yang tersedia pada restoran beserta harga-harganya, informasi dari restoran serta pengguna yang menggunakan sistem ini, dan dapat melacak histori dari transaksi pada kasir. Area manajerial sendiri digunakan oleh manajer atau pemilik restoran.

4.2 Analisis Kebutuhan

Tahap analisis kebutuhan pada penelitian ini diawali dengan elisitasi kebutuhan, identifikasi aktor yang terlibat dalam sistem, pendefinisian daftar spesifikasi kebutuhan yang kemudian ditranslasikan ke dalam diagram *use case*. Tahapan ini dilakukan untuk menggambarkan kebutuhan-kebutuhan yang dimiliki oleh sistem agar dapat memenuhi kebutuhan restoran pada umumnya. Gambar 4.2 merepresentasikan tahapan dari analisis kebutuhan.



Gambar 4.2 Diagram Alir Analisis Kebutuhan

4.2.1 Elisitasi Kebutuhan

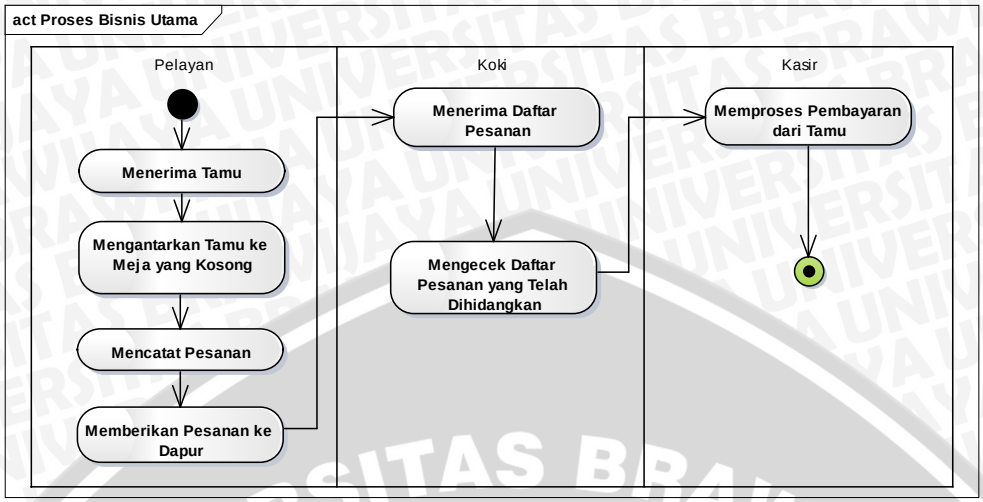
Tahap ini dilakukan untuk mengetahui masalah yang perlu dipecahkan dan mengenali pemangku kepentingan dari sistem. Teknik yang digunakan dalam elisitasi kebutuhan adalah studi literatur (*literature review*) dan pengamatan dokumen. Studi literatur yang dilakukan mengacu kepada empat penelitian

terdahulu yang telah dipaparkan pada bagian Kajian Pustaka, sedangkan pengamatan dokumen dilakukan dengan pengumpulan lembar pesanan dan struk pembayaran pada beberapa restoran yang dapat dilihat pada bagian Lampiran. Studi literatur dilakukan untuk mendapatkan fitur-fitur umum terkait kegiatan operasional utama restoran sedangkan pengamatan dokumen dilakukan guna mendapatkan data yang umum digunakan pada restoran.

Berdasarkan hasil dari teknik studi literatur maka ditemukan kebutuhan yang harus ada dalam sistem pengelolaan restoran yaitu meliputi:

1. Area fungsional pemasaran bagian POS (*Point of Sales*)
    a. Bagian ini harus memiliki fungsi untuk memesan makanan serta melihat status pesanan makanan tersebut secara *real-time* yang dilakukan oleh pelayan (SKPL_01).
    b. Bagian ini dapat mengelola dan mencetak pembayaran tagihan menu makanan yang dilakukan oleh kasir (SKPL_02).
2. Area fungsional operasional bagian dapur
    a. Bagian ini harus memiliki fungsi untuk menampilkan daftar pesanan tamu yang disertai jumlah total pesanan, nomor meja, dan waktu penyiapan menu pesanan secara *real-time* untuk koki (SKPL_03).
    b. Bagian ini harus menyediakan fitur untuk memberikan notifikasi kepada koki jika penyiapan pesanan melebihi batas waktu yang ditentukan (SKPL_04).
    c. Bagian ini harus memiliki fungsi untuk mencari dan mencetak menu pesanan ulang oleh koki jika lembar pesanan hilang (SKPL_05).
    d. Bagian ini harus memiliki fungsi untuk memvalidasi dan mencetak lembar pesanan yang telah disiapkan oleh koki (SKPL_06).
3. Area pendukung manajerial
    a. Bagian ini dapat membantu manajer untuk mengelola menu makanan dan minuman yang tersedia pada restoran (SKPL_07).
    b. Bagian ini dapat membantu manajer untuk mengelola informasi restoran dan pengguna sistem (SKPL_08).
    c. Bagian ini dapat membantu manajer untuk melihat histori transaksi kasir (SKPL_09).
4. Umum
    a. Sistem harus menyediakan fitur autentikasi terhadap pengguna dengan tujuan agar hanya pengguna yang dikenali oleh sistem yang dapat mengoperasikannya (SKPL_10).

Gambar 4.3 menjelaskan proses bisnis pada prosedur pelayanan di kebanyakan restoran.



Gambar 4.3 Proses Bisnis Utama Restoran

4.2.2 Identifikasi Aktor

Tahap ini dilakukan untuk mengidentifikasi aktor yang terlibat secara langsung terhadap sistem yang dikembangkan. Pengidentifikasian aktor pada tahap ini mengacu kepada tahap elisitasi kebutuhan. Tabel 4.1 merepresentasikan aktor-aktor tersebut beserta deskripsinya.

Tabel 4.1 Identifikasi Aktor

Aktor	Deskripsi
Pelayan	Pelayan merupakan pengguna yang hanya memiliki hak akses pada bagian POS khususnya bagian pemesanan.
Kasir	Kasir merupakan pengguna yang hanya memiliki hak akses pada bagian POS khususnya bagian pembayaran.
Koki	Koki merupakan pengguna yang memiliki hak untuk mengakses bagian dapur.
Manajer	Manajer merupakan pengguna yang memiliki hak untuk mengakses bagian manajerial.
Pengunjung	Pengunjung merupakan pengguna yang belum teridentifikasi oleh sistem.

4.2.3 Spesifikasi Kebutuhan

Spesifikasi kebutuhan yang disediakan pada sistem ini mengacu kepada elisitasi kebutuhan pada bagian sebelumnya. Penentuan prioritas pada daftar spesifikasi kebutuhan fungsional didasarkan atas tingkat keterlibatannya terhadap proses bisnis utama yang telah diusulkan pada tahap elisitasi kebutuhan. Tabel 4.2 menjelaskan spesifikasi kebutuhan fungsional, sedangkan Tabel 4.3 menjelaskan spesifikasi kebutuhan non fungsional dari sistem ini.

Tabel 4.2 Spesifikasi Kebutuhan Fungsional

Nomor SKPL	Kebutuhan	Use Case	Aktor	Prioritas
SKPL_01_1	Sistem dapat membuat menu pemesanan makanan dan minuman	Membuat Pemesanan Menu	Pelayan	Tinggi
SKPL_01_2	Sistem dapat melihat status menu pemesanan makanan dan minuman secara *real-time*	Melihat Status Pemesanan Menu	Pelayan	Tinggi
SKPL_02_1	Sistem dapat membuat tagihan pembayaran tamu	Membuat Tagihan Pembayaran	Kasir	Tinggi
SKPL_02_2	Sistem dapat melihat tagihan pembayaran tamu secara *real-time*	Melihat Tagihan Pembayaran	Kasir	Tinggi
SKPL_03	Sistem dapat menampilkan daftar pesanan tamu secara *real-time*	Melihat Daftar Pesanan	Koki	Tinggi
SKPL_04	Sistem dapat memberikan pemberitahuan kepada koki jika penyiapan pesanan melebihi batas waktu yang ditentukan	Notifikasi Pesanan	Koki	Tinggi
SKPL_02_3	Sistem dapat mencetak tagihan pembayaran tamu	Mencetak Tagihan Pembayaran	Kasir	Sedang
SKPL_05	Sistem dapat mencari pesanan jika lembar menu pesanan hilang	Mencari Pesanan	Koki	Sedang
SKPL_06_1	Sistem dapat memvalidasi pesanan yang telah disiapkan oleh koki	Memvalidasi Pesanan	Koki	Sedang
SKPL_06_2	Sistem dapat mencetak lembar pesanan	Mencetak Pesanan	Koki	Sedang
SKPL_07	Sistem dapat mengelola (menambah, melihat, mengubah, dan menghapus) menu makanan dan minuman	Kelola Menu Restoran	Manajer	Rendah

SKPL_08_1	Sistem dapat mengelola (menambah, melihat, mengubah, dan menghapus) data pengguna sistem	Kelola Pengguna	Manajer	Rendah
SKPL_08_2	Sistem dapat mengubah informasi restoran	Mengubah Informasi Restoran	Manajer	Rendah
SKPL_09	Sistem dapat menampilkan histori transaksi kasir	Melihat Histori Transaksi	Manajer	Rendah
SKPL_10	Sistem harus menyediakan fitur autentikasi berupa login	Login	Pengunjung	Rendah

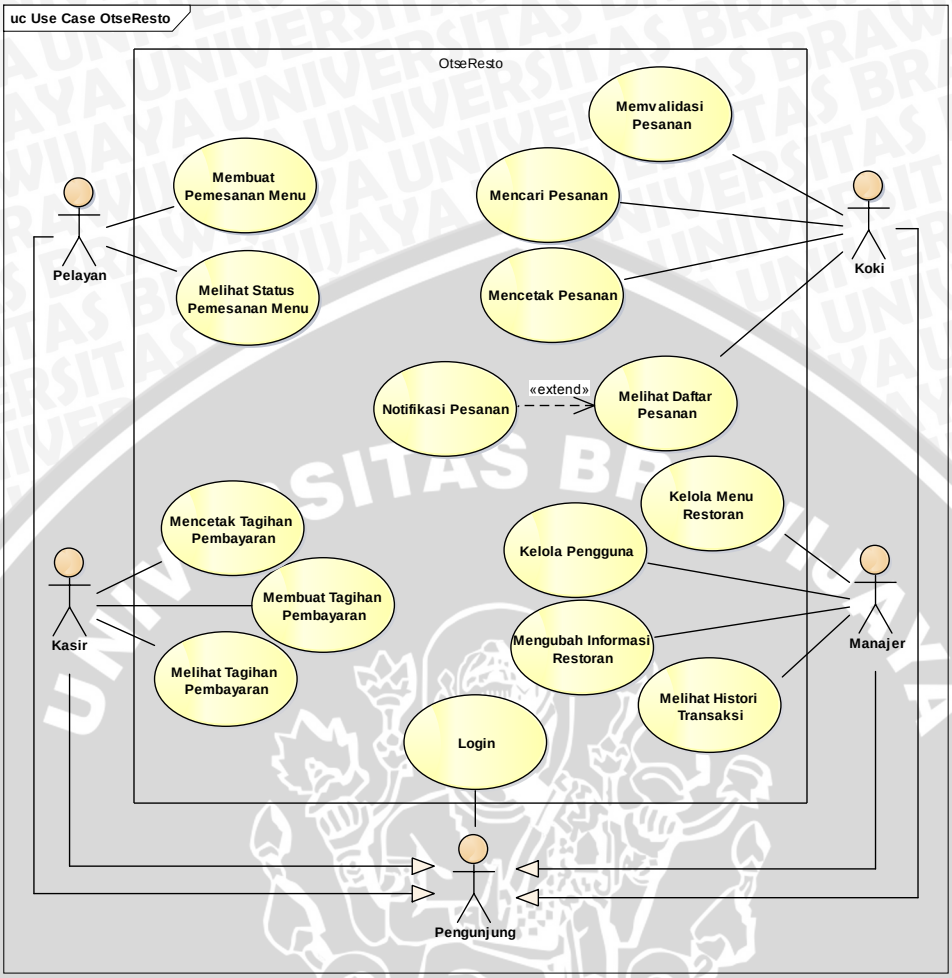
Tabel 4.3 Spesifikasi Kebutuhan Non Fungsional

Nomor SKPL	Parameter	Deskripsi Kebutuhan
SKPL_11	*Reliability*	Sistem dapat melayani perintah pengguna dengan batas toleransi *error* hingga 5%.
SKPL_12	*Usability*	Sistem dapat digunakan dengan mudah oleh pengguna.
SKPL_13	*Efficiency*	Sistem dapat melayani perintah pengguna dengan batas waktu respons di bawah 10 detik.
SKPL_14	*Maintainability*	Sistem dapat dipelihara dengan mudah yang ditunjukkan dengan nilai MI (*Maintainability Index*) lebih dari sama dengan 65.
SKPL_15	*Portability*	Sistem dapat dijalankan di berbagai *web browser* yang berbeda.

4.2.4 Diagram *Use Case*

Diagram *use case* merupakan salah satu diagram UML (*Unified Modeling Language*) yang memodelkan perilaku sistem dari tampak luar. Diagram ini menggambarkan aktor, *use case*, dan hubungan terkait keduanya pada sistem yang dibangun.

Setiap *use case* yang dimodelkan akan dijelaskan lebih detail pada bagian skenario *use case*. Tabel skenario *use case* terdiri dari nomor SKPL, nama *use case* tersebut, aktor yang menggunakannya, tujuan dan deskripsi *use case*, beserta skenario utama dan skenario alternatif yang dilakukan ketika menggunakan *use case* tersebut. Gambar 4.4 menjelaskan diagram *use case* dari sistem secara keseluruhan.



Gambar 4.4 Diagram *Use Case* OtseResto

1. *Use Case* Membuat Pemesanan Menu

Use case ini memiliki tingkat prioritas yang tinggi sebab masuk ke dalam domain permasalahan dan mendukung proses bisnis utama pelayanan restoran dalam memesan menu makanan dan minuman. Skenario membuat pemesanan menu makanan dan minuman pada restoran dapat dilihat pada Tabel 4.4.

Tabel 4.4 Skenario Membuat Pemesanan Menu

Skenario Kasus pada Sistem	
Nomor SKPL	SKPL_01_1
Nama	Membuat Pemesanan Menu
Tujuan	Agar pelayan dapat melakukan pemesanan menu makanan dan minuman yang ada di restoran
Deskripsi	*Use case* ini menjelaskan bagaimana proses pemesanan menu makanan dan minuman
Aktor	Pelayan

Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	**Reaksi Sistem**
1. Aktor menekan tombol Pemesanan Menu pada sistem	2. Sistem mengecek status autentikasi
	3. Sistem menampilkan halaman Pemesanan Menu yang berisi *form* pemesanan menu makanan
4. Aktor mengisi *form* pemesanan menu berupa nama pelanggan, nomor meja, dan menu pesanan kemudian menekan tombol Buat	5. Sistem menerima masukan dari aktor serta menampilkan notifikasi berhasil
Skenario Alternatif 1: Jika sesi aktor tidak valid	
	3. Sistem menampilkan halaman login
Skenario Alternatif 2: Jika *input* menu pesanan tidak valid	
	5. Sistem mengembalikan ke halaman Pemesanan Menu dan menampilkan notifikasi gagal
Kondisi Akhir	Pesanan menu makanan dan minuman disimpan ke dalam sistem

2. *Use* Case Melihat Status Pemesanan Menu

Use case ini juga memiliki tingkat prioritas yang tinggi sebab masuk ke dalam domain permasalahan untuk berbagi informasi antara dapur dan ruang makan. Skenario melihat status pemesanan menu makanan dan minuman dapat dilihat pada Tabel 4.5.

Tabel 4.5 Skenario Melihat Status Pemesanan Menu

Skenario Kasus pada Sistem	
Nomor SKPL	SKPL_01_2
Nama	Melihat Status Pemesanan Menu
Tujuan	Agar pelayan dapat melihat status pemesanan menu makanan dan minuman yang telah dipesan
Deskripsi	*Use case* ini menjelaskan bagaimana pelayan melihat status pemesan menu makanan dan minuman

Aktor	Pelayan
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	**Reaksi Sistem**
1. Aktor menekan tombol Status Pemesanan pada sistem	2. Sistem mengecek status autentikasi
	3. Sistem menampilkan halaman status pemesanan menu secara *real-time*
Skenario Alternatif 1: Jika sesi aktor tidak valid	
	3. Sistem menampilkan halaman login
Kondisi Akhir	Sistem menampilkan status pemesanan menu makanan dan minuman yang aktif secara *real-time*

3. *Use Case* Membuat Tagihan Pembayaran

Use case ini juga memiliki tingkat prioritas yang tinggi sebab masuk ke dalam domain permasalahan dan mendukung proses bisnis utama pelayanan restoran dalam membuat tagihan makanan dan minuman yang dikonsumsi. Skenario membuat tagihan pembayaran atas makanan dan minuman yang telah dikonsumsi dapat dilihat pada Tabel 4.6.

Tabel 4.6 Skenario Membuat Tagihan Pembayaran

Skenario Kasus pada Sistem	
Nomor SKPL	SKPL_02_1
Nama	Membuat Tagihan Pembayaran
Tujuan	Agar kasir dapat membuat tagihan pembayaran berdasarkan menu makanan dan minuman yang telah dikonsumsi oleh tamu
Deskripsi	*Use case* ini menjelaskan bagaimana proses pembuatan tagihan pembayaran
Aktor	Kasir
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman Tagihan Pembayaran
Aksi Aktor	**Reaksi Sistem**

1. Aktor menekan tombol Proses pada sistem	2. Sistem mengecek status autentikasi
	3. Sistem menampilkan form tagihan pembayaran berdasarkan ID Tagihan
4. Aktor mengisi *form* tagihan pembayaran berupa tunai atau ID referensi kemudian menekan tombol Cetak	5. Sistem memproses dan mencetak struk tagihan pembayaran
Skenario Alternatif 1: Jika sesi aktor tidak valid	
	3. Sistem menampilkan halaman login
Skenario Alternatif 2: Jika *input* tunai dan ID referensi tidak valid	
	5. Sistem menampilkan halaman Tagihan Pembayaran beserta notifikasi gagal
Kondisi Akhir	Pesanan menu makanan disimpan ke dalam sistem

4. *Use Case* Melihat Tagihan Pembayaran

Use case ini juga memiliki tingkat prioritas yang tinggi sebab masuk ke dalam domain permasalahan dan mendukung proses bisnis utama pelayanan restoran dalam melihat daftar tagihan pembayaran yang aktif atau belum dibayar. Skenario melihat daftar tagihan yang aktif atau belum dibayar dapat dilihat pada Tabel 4.7.

Tabel 4.7 Skenario Melihat Tagihan Pembayaran

Skenario Kasus pada Sistem	
Nomor SKPL	SKPL_02_2
Nama	Melihat Tagihan Pembayaran
Tujuan	Agar kasir dapat melihat daftar tagihan pembayaran yang aktif secara *real-time*
Deskripsi	*Use case* ini menjelaskan bagaimana proses melihat daftar tagihan pembayaran
Aktor	Kasir
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	**Reaksi Sistem**

1. Aktor menekan tombol Tagihan Pembayaran pada sistem	2. Sistem mengecek status autentikasi
	3. Sistem menampilkan halaman Tagihan Pembayaran secara *real-time*
Skenario Alternatif 1: Jika sesi aktor tidak valid	
	3. Sistem menampilkan halaman login
Kondisi Akhir	Sistem menampilkan daftar tagihan pembayaran secara *real-time*

5. *Use Case* Melihat Daftar Pesanan

Use case ini juga memiliki tingkat prioritas yang tinggi sebab masuk ke dalam domain permasalahan dan mendukung proses bisnis utama pelayanan restoran dalam menampilkan daftar pesanan yang belum disiapkan oleh koki di dapur. Skenario melihat daftar pesanan yang harus disiapkan dapat dilihat pada Tabel 4.8.

Tabel 4.8 Skenario Melihat Daftar Pesanan

Skenario Kasus pada Sistem	
Nomor SKPL	SKPL_03
Nama	Melihat Daftar Pesanan
Tujuan	Agar koki dapat melihat daftar pesanan yang harus disiapkan
Deskripsi	*Use case* ini menjelaskan proses melihat daftar pesanan secara *real-time*
Aktor	Koki
Skenario Utama	
Kondisi Awal	Aktor telah logn dan sistem menampilkan halaman utama aktor
Aksi Aktor	**Reaksi Sistem**
1. Aktor menekan tombol Daftar Pesanan pada sistem	2. Sistem mengecek status autentikasi
	3. Sistem menampilkan halaman Daftar Pesanan secara *real-time*
Skenario Alternatif 1: Jika sesi aktor tidak valid	
	3. Sistem menampilkan halaman login
Kondisi Akhir	Sistem menampilkan daftar pesanan secara *real-time*

6. *Use Case* Notifikasi Pesanan

Use case ini juga memiliki tingkat prioritas yang tinggi sebab masuk ke dalam domain permasalahan dan mendukung proses bisnis utama pelayanan restoran dalam memberikan notifikasi jika pesanan belum disiapkan oleh koki dalam jangka waktu tertentu. Skenario pemberian notifikasi pesanan kepada koki dapat dilihat pada Tabel 4.9.

Tabel 4.9 Skenario Notifikasi Pesanan

Skenario Kasus pada Sistem	
Nomor SKPL	SKPL_04
Nama	Notifikasi Pesanan
Tujuan	Agar koki dapat memprioritaskan pesanan yang sudah antre lama
Deskripsi	*Use case* ini menjelaskan bagaimana sistem memberikan notifikasi pesanan
Aktor	Koki
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman Daftar Pesanan
Aksi Aktor	**Reaksi Sistem**
	1. Sistem melakukan perhitungan waktu penyiapan pesanan
	2. Sistem menampilkan notifikasi peringatan jika melewati batas waktu yang ditentukan
Kondisi Akhir	Sistem memberikan notifikasi pesanan berupa perubahan warna daftar pesanan

7. *Use Case* Mencetak Tagihan Pembayaran

Use case ini memiliki tingkat prioritas yang sedang sebab tidak masuk ke dalam domain permasalahan namun mendukung proses bisnis utama pelayanan restoran dalam mencetak tagihan pembayaran. Skenario mencetak tagihan pembayaran dapat dilihat pada Tabel 4.10.

Tabel 4.10 Skenario Mencetak Tagihan Pembayaran

Skenario Kasus pada Sistem	
Nomor SKPL	SKPL_02_3
Nama	Mencetak Tagihan Pembayaran

Tujuan	Agar kasir dapat melakukan pencetakan tagihan pembayaran
Deskripsi	*Use case* ini menjelaskan bagaimana proses mencetak tagihan pembayaran
Aktor	Kasir
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman Tagihan Pembayaran
Aksi Aktor	**Reaksi Sistem**
1. Aktor menekan tombol Cetak pada sistem	2. Sistem mengecek status autentikasi
	3. Sistem mencetak tagihan pembayaran
	4. Sistem menampilkan halaman Tagihan Pembayaran kembali
Skenario Alternatif 1: Jika sesi aktor tidak valid	
	3. Sistem menampilkan halaman login
Kondisi Akhir	Sistem mencetak tagihan pembayaran

8. *Use Case* Mencari Pesanan

Use case ini juga memiliki tingkat prioritas yang sedang sebab tidak masuk ke dalam domain permasalahan namun mendukung proses bisnis utama pelayanan restoran untuk mencari pesanan jika lembar pesanan yang telah dicetak hilang. Skenario pencarian pesanan yang telah selesai dihidangkan dapat dilihat pada Tabel 4.11.

Tabel 4.11 Skenario Mencari Pesanan

Skenario Kasus pada Sistem	
Nomor SKPL	SKPL_05
Nama	Mencari Pesanan
Tujuan	Agar koki dapat mencari lembar pesanan jika lembar pesanan yang sebelumnya dicetak telah hilang
Deskripsi	*Use case* ini menjelaskan bagaimana sistem melakukan pencarian lembar pesanan
Aktor	Koki
Skenario Utama	

Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	**Reaksi Sistem**
1. Aktor menekan tombol Cari Pesanan pada sistem	2. Sistem mengecek status autentikasi
	3. Sistem menampilkan halaman Cari Pesanan beserta formnya.
4. Aktor mengisi form untuk mencari pesanan dan menekan tombol Cari	5. Sistem melakukan pencarian data
	6. Sistem mengembalikan data dan menampilkannya pada halaman Cari Pesanan
Skenario Alternatif 1: Jika sesi aktor tidak valid	
	3. Sistem menampilkan halaman login
Kondisi Akhir	Sistem menampilkan hasil pencarian

9. *Use Case* Memvalidasi Pesanan

Use case ini juga memiliki tingkat prioritas yang sedang sebab tidak masuk ke dalam domain permasalahan namun mendukung proses bisnis utama pelayanan restoran dalam memvalidasi pesanan yang telah diproses oleh koki. Skenario memvalidasi pesanan yang telah disiapkan dapat dilihat pada Tabel 4.12.

Tabel 4.12 Skenario Memvalidasi Pesanan

Skenario Kasus pada Sistem	
Nomor SKPL	SKPL_06_1
Nama	Memvalidasi Pesanan
Tujuan	Agar koki dapat memberikan informasi bahwa pesanan telah selesai disiapkan
Deskripsi	*Use case* ini menjelaskan bagaimana sistem memvalidasi pesanan
Aktor	Koki
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman Daftar Pesanan
Aksi Aktor	**Reaksi Sistem**
1. Aktor menekan tombol Validasi pada sistem	2. Sistem mengecek status autentikasi

	3. Sistem mengubah status pesanan sekaligus mencetak lembar pesanan
	4. Sistem menampilkan halaman Daftar Pesanan kembali
Skenario Alternatif 1: Jika sesi aktor tidak valid	
	3. Sistem menampilkan halaman login
Kondisi Akhir	Sistem mengubah status pesanan sekaligus mencetak lembar pesanan

10. *Use Case* Mencetak Pesanan

Use case ini juga memiliki tingkat prioritas yang sedang sebab tidak masuk ke dalam domain permasalahan namun mendukung proses bisnis utama pelayanan restoran dalam mencetak lembar pesanan yang telah disiapkan oleh koki. Skenario mencetak lembar pesanan dapat dilihat pada Tabel 4.13.

Tabel 4.13 Skenario Mencetak Pesanan

Skenario Kasus pada Sistem	
Nomor SKPL	SKPL_06_2
Nama	Mencetak Pesanan
Tujuan	Agar koki dapat mencetak lembar pesanan
Deskripsi	*Use case* ini menjelaskan bagaimana sistem mencetak lembar pesanan
Aktor	Koki
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan pesanan pada halaman pencarian pesanan
Aksi Aktor	**Reaksi Sistem**
1. Aktor menekan tombol Cetak pada sistem	2. Sistem mengecek status autentikasi
	3. Sistem mencetak lembar pesanan
	4. Sistem menampilkan halaman Cari Pesanan kembali
Skenario Alternatif 1: Jika sesi aktor tidak valid	
	3. Sistem menampilkan halaman login

Kondisi Akhir	Sistem mencetak lembar pesanan

11. *Use Case* Kelola Menu Restoran

Use case ini memiliki tingkat prioritas yang rendah sebab tidak masuk ke dalam domain permasalahan dan juga tidak mendukung proses bisnis utama pelayanan restoran namun *use case* ini harus tersedia dalam sistem untuk penyediaan data-data menu makanan dan minuman yang tersedia pada restoran. Skenario pengelolaan menu restoran dapat dilihat pada Tabel 4.14.

Tabel 4.14 Skenario Kelola Menu Restoran

Skenario Kasus pada Sistem	
Nomor SKPL	SKPL_07
Nama	Kelola Menu Restoran
Tujuan	Agar manajer dapat mengelola menu makanan dan minuman yang tersedia pada restoran
Deskripsi	*Use case* ini menjelaskan bagaimana proses pengelolaan menu makanan dan minuman
Aktor	Manajer
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	**Reaksi Sistem**
1. Aktor mengunjungi halaman pengelolaan menu makanan dan minuman	2. Sistem mengecek status autentikasi
	3. Sistem menampilkan halaman Pengelolaan Menu
4. Aktor dapat menambah, melihat, mengubah, dan menghapus menu makanan dan minuman	5. Sistem melakukan proses terkait dengan aksi menambah, melihat, mengubah, atau menghapus menu makanan dan minuman
Skenario Alternatif 1: Jika sesi aktor tidak valid	
	3. Sistem menampilkan halaman login
Skenario Alternatif 2: Jika *input* harga tidak valid	
	5. Sistem mengembalikan menampilkan notifikasi gagal

Kondisi Akhir	Sistem dapat membuat, melihat, mengubah, dan menghapus menu makanan dan minuman pada sistem

12. *Use Case* Kelola Pengguna

Use case ini juga memiliki tingkat prioritas yang rendah sebab tidak masuk ke dalam domain permasalahan dan juga tidak mendukung proses bisnis utama pelayanan restoran namun *use case* ini harus tersedia dalam sistem untuk pengelolaan pengguna yang dapat mengoperasikan sistem. Skenario pengelolaan data pengguna dapat dilihat pada Tabel 4.15.

Tabel 4.15 Skenario Kelola Pengguna

Skenario Kasus pada Sistem	
Nomor SKPL	SKPL_08_1
Nama	Kelola Pengguna
Tujuan	Agar manajer dapat mengelola pengguna yang dapat mengakses sistem
Deskripsi	*Use case* ini menjelaskan bagaimana proses pengelolaan data pengguna
Aktor	Manajer
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	**Reaksi Sistem**
1. Aktor mengunjungi halaman pengelolaan pengguna	2. Sistem mengecek status autentikasi
	3. Sistem menampilkan halaman Pengelolaan Pengguna
4. Aktor dapat menambah, melihat, mengubah, dan menghapus data pengguna	5. Sistem melakukan proses terkait dengan aksi menambah, melihat, mengubah, atau menghapus data pengguna
Skenario Alternatif 1: Jika sesi aktor tidak valid	
	3. Sistem menampilkan halaman login
Skenario Alternatif 2: Jika *input* nomor induk tidak valid	
	5. Sistem mengembalikan menampilkan notifikasi gagal

Kondisi Akhir	Sistem dapat membuat, melihat, mengubah, dan menghapus data pengguna pada sistem

13. *Use Case* Mengubah Informasi Restoran

Use case ini juga memiliki tingkat prioritas yang rendah sebab tidak masuk ke dalam domain permasalahan dan juga tidak mendukung proses bisnis utama pelayanan restoran namun *use case* ini harus tersedia dalam sistem untuk penetapan informasi restoran. Skenario pengubahan informasi restoran dapat dilihat pada Tabel 4.16.

Tabel 4.16 Skenario Mengubah Informasi Restoran

Skenario Kasus pada Sistem	
Nomor SKPL	SKPL_08_2
Nama	Mengubah Informasi Restoran
Tujuan	Agar manajer dapat menyesuaikan informasi restoran yang dikelolanya
Deskripsi	*Use case* ini menjelaskan proses pengubahan informasi restoran
Aktor	Manajer
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	**Reaksi Sistem**
1. Aktor menekan tombol Ubah Informasi Restoran pada sistem	2. Sistem mengecek status autentikasi
	3. Sistem memberikan *form* pengubahan informasi restoran
4. Aktor mengisi *form* pengubahan informasi restoran dan kemudian menekan tombol Ubah	5. Sistem menyimpan pengubahan dan menampilkan notifikasi berhasil
Skenario Alternatif 1: Jika sesi aktor tidak valid	
	3. Sistem menampilkan halaman login
Skenario Alternatif 2: Jika *input* telepon, jumlah meja, pajak dan diskon tidak valid	
	5. Sistem tidak menyimpan pengubahan dan menampilkan notifikasi gagal

Kondisi Akhir	Data informasi restoran disimpan oleh sistem

14. *Use Case* Melihat Histori Transaksi

Use case ini juga memiliki tingkat prioritas yang rendah sebab tidak masuk ke dalam domain permasalahan dan juga tidak mendukung proses bisnis utama pelayanan restoran namun *use case* ini harus tersedia dalam sistem untuk membantu manajer dalam mengetahui histori transaksi yang dilakukan kasir. Skenario melihat histori transaksi dapat dilihat pada Tabel 4.17.

Tabel 4.17 Skenario Melihat Histori Transaksi

Skenario Kasus pada Sistem	
Nomor SKPL	SKPL_09
Nama	Melihat Histori Transaksi
Tujuan	Agar manajer dapat melihat histori seluruh transaksi yang dilakukan kasir
Deskripsi	*Use case* ini menjelaskan proses melihat histori seluruh transaksi
Aktor	Manajer
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	**Reaksi Sistem**
1. Aktor menekan tombol Daftar Histori Transaksi pada sistem	2. Sistem mengecek status autentikasi
	3. Sistem menampilkan halaman Histori Transaksi
Skenario Alternatif 1: Jika sesi aktor tidak valid	
	3. Sistem menampilkan halaman login
Kondisi Akhir	Sistem menampilkan seluruh histori transaksi

15. Use Case Login

Use case ini juga memiliki tingkat prioritas yang rendah sebab tidak masuk ke dalam domain permasalahan dan juga tidak mendukung proses bisnis utama pelayanan restoran namun *use case* ini harus tersedia dalam sistem agar dapat mengenali pengguna yang seharusnya dapat mengakses fungsi sistem sesuai hak aksesnya masing-masing. Skenario login dapat dilihat pada Tabel 4.18.

Tabel 4.18 Skenario Login

Skenario Kasus pada Sistem	
Nomor SKPL	SKPL_10
Nama	Login
Tujuan	Agar sistem dapat mengenali pengguna yang mengakses sistem
Deskripsi	*Use case* ini menjelaskan proses login oleh aktor
Aktor	Pengunjung
Skenario Utama	
Kondisi Awal	Aktor mengakses halaman login
Aksi Aktor	**Reaksi Sistem**
1. Aktor memasukkan *username* dan *password*	2. Sistem memeriksa validitas *input* aktor
	3. Sistem menampilkan halaman utama aktor
Skenario Alternatif 1: Jika *input username* atau *password* tidak valid	
	3. Sistem menampilkan notifikasi kesalahan *input*
Kondisi Akhir	Sistem menampilkan halaman utama sesuai hak akses aktor yang teridentifikasi oleh sistem

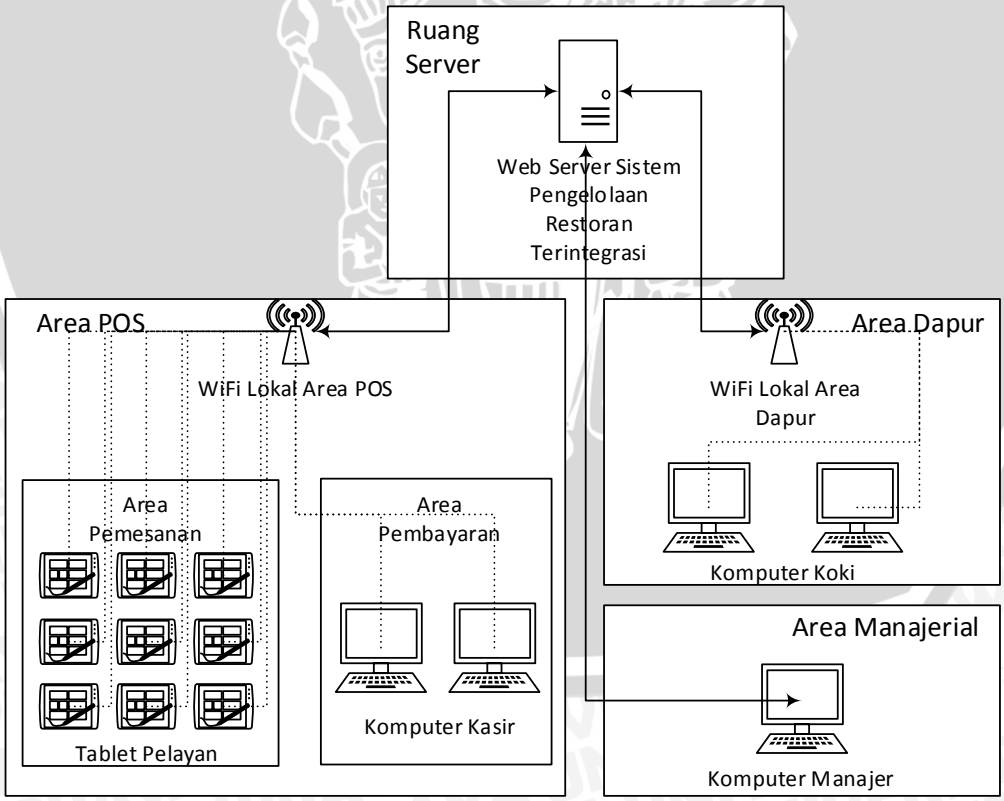
4.3 Perancangan Sistem

Tahap perancangan sistem pada penelitian ini dilakukan dengan enam tahap pengerjaan yaitu merancang arsitektur dari sistem yang dibangun, memodelkan diagram kelas, kemudian melakukan pemodelan diagram ER (*Entity Relationship*), lalu memodelkan diagram aktivitas yang mengacu kepada skenario *use case*, lalu akan dilakukan pemodelan dengan diagram sekuen berdasarkan diagram aktivitas, dan pada tahap akhir perancangan sistem akan dilakukan pemodelan antarmuka sistem. Perancangan sistem yang dilakukan menggunakan pendekatan beriorientasi objek dengan diagram UML (*Unified Modeling Language*). Gambar 4.5 merepresentasikan tahapan dari perancangan sistem.

Gambar 4.5 Diagram Alir Perancangan Sistem

4.3.1 Perancangan Arsitektural

Sistem pengelolaan restoran terintegrasi ini menggunakan arsitektur *client-server* yang bertujuan untuk dapat diakses dari berbagai area di dalam restoran. Desain arsitektural sistem yang dikembangkan dapat dilihat pada Gambar 4.6.

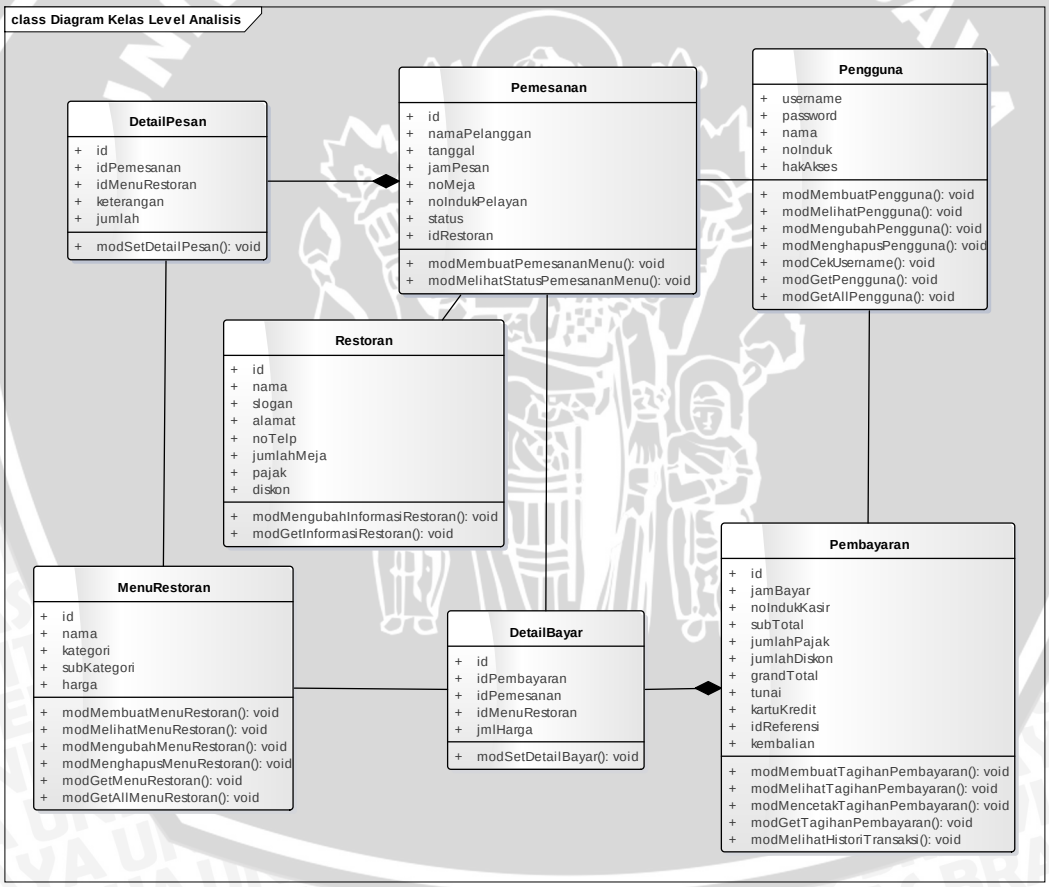


Gambar 4.6 Arsitektur OtseResto

4.3.2 Perancangan Diagram Kelas

Perancangan diagram kelas pada penelitian ini dibagi menjadi dua tahap yaitu perancangan diagram kelas level analisis dan diagram kelas level implementasi. Diagram kelas level analisis digunakan untuk merepresentasikan atribut data dan interaksi yang digunakan dalam basis data sedangkan diagram kelas level implementasi digunakan untuk merepresentasikan struktur dari kelas yang diimplementasikan dalam sistem.

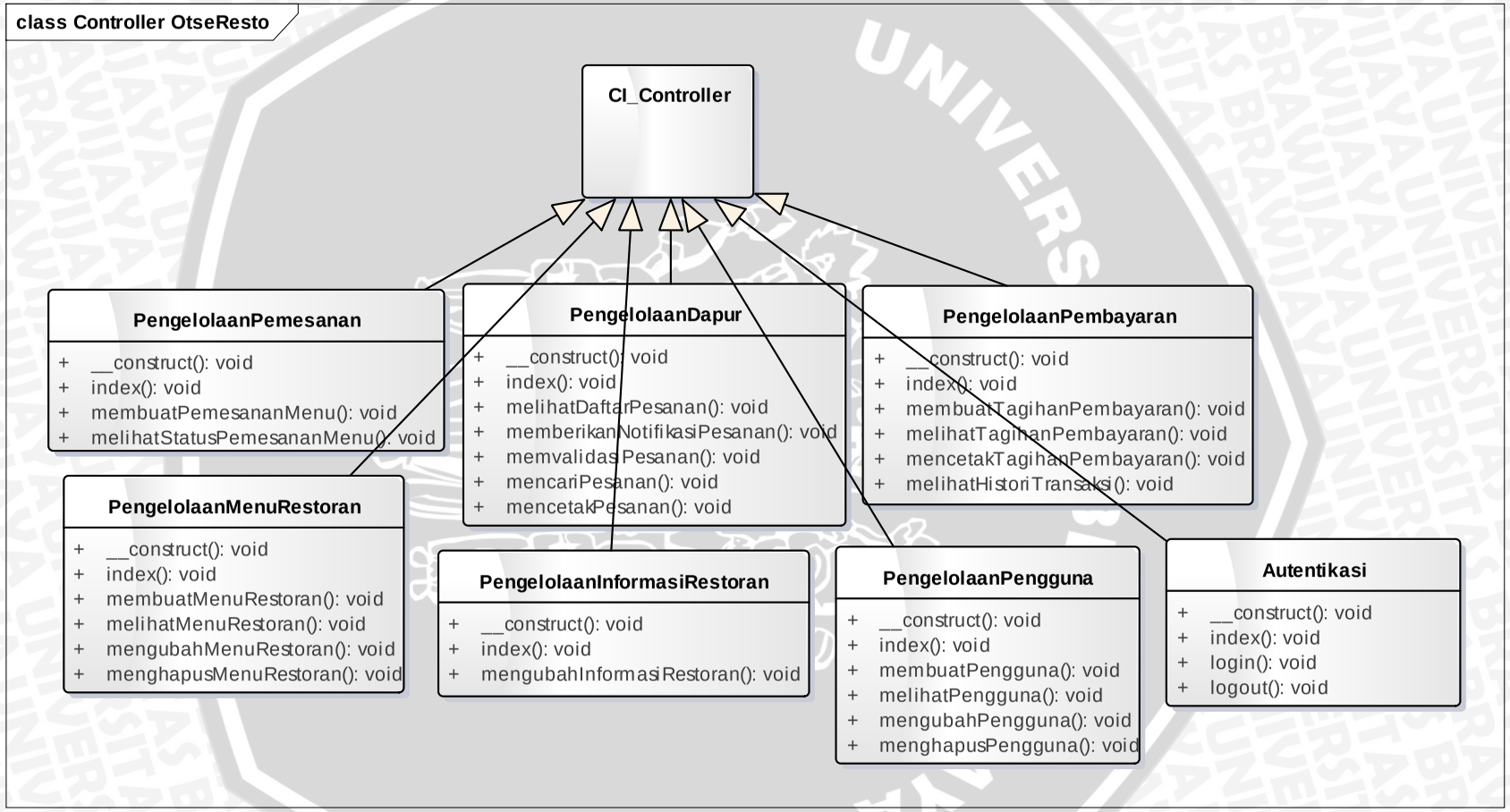
Atribut pada diagram kelas level analisis didapatkan dari pengamatan dokumen lembar pesanan dan tagihan pembayaran pada beberapa restoran yang dapat dilihat pada bagian Lampiran. Kemudian, diagram kelas level implementasi sendiri dibagi menjadi dua jenis yaitu kelas proses (*controller)* dan kelas data (*model*). Hal tersebut dilakukan sebab sistem ini menggunakan pola arsitektur MVC (*Model, View Controller*). Gambar 4.7 merepresentasikan diagram kelas level analisis.



Gambar 4.7 Diagram Kelas OtseResto Level Analisis

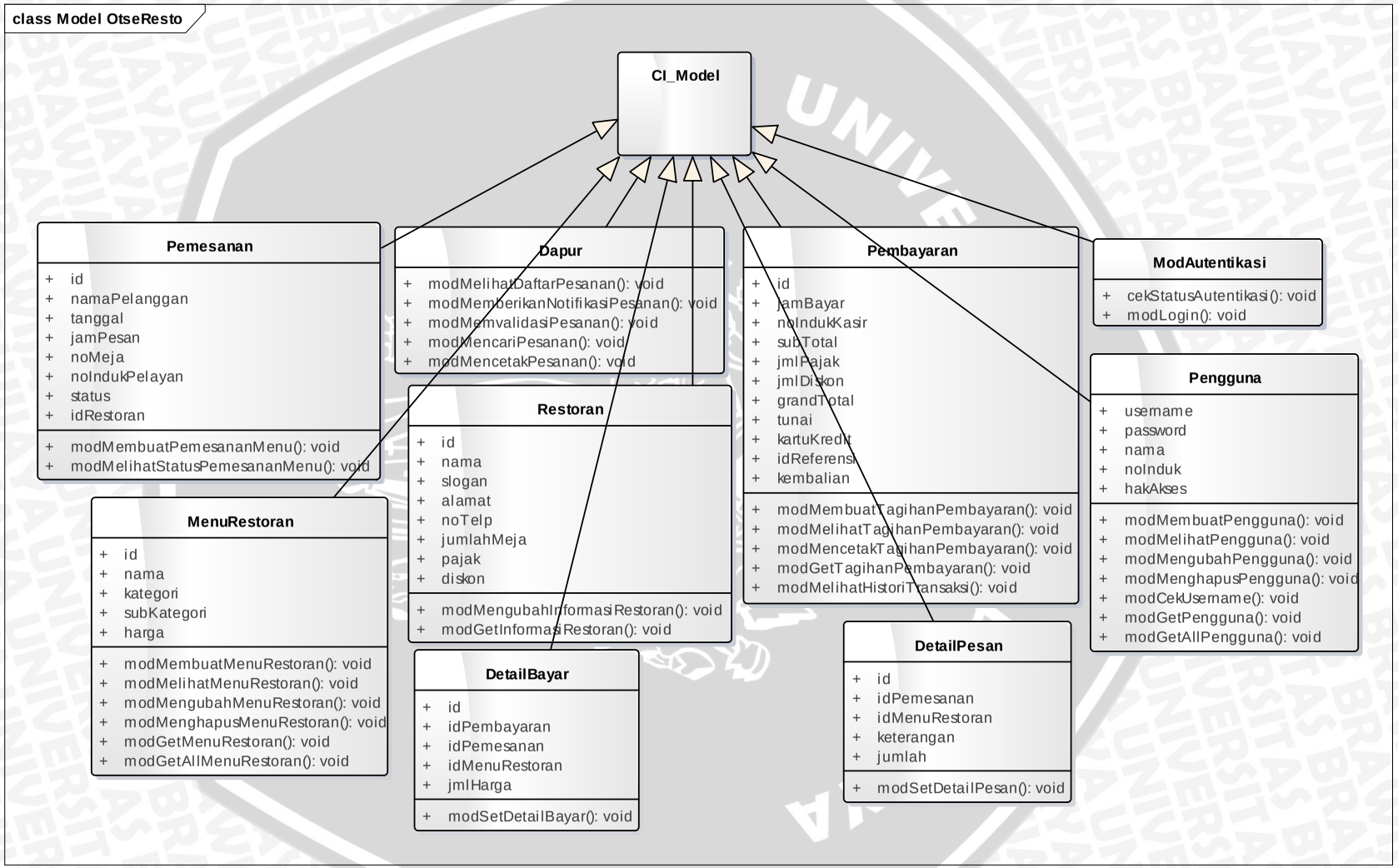
Sedangkan Gambar 4.8 dan Gambar 4.9 merepresentasikan diagram kelas level implementasi. Gambar 4.8 fokus pada bagian *Controller* dari sistem. Sedangkan Gambar 4.9 fokus pada bagian *Model* dari sistem.

UNIVERSITAS BRAWIJAYA
repository.ub.ac.id



Gambar 4.8 Diagram Kelas *Controller* OtseResto Level Implementasi

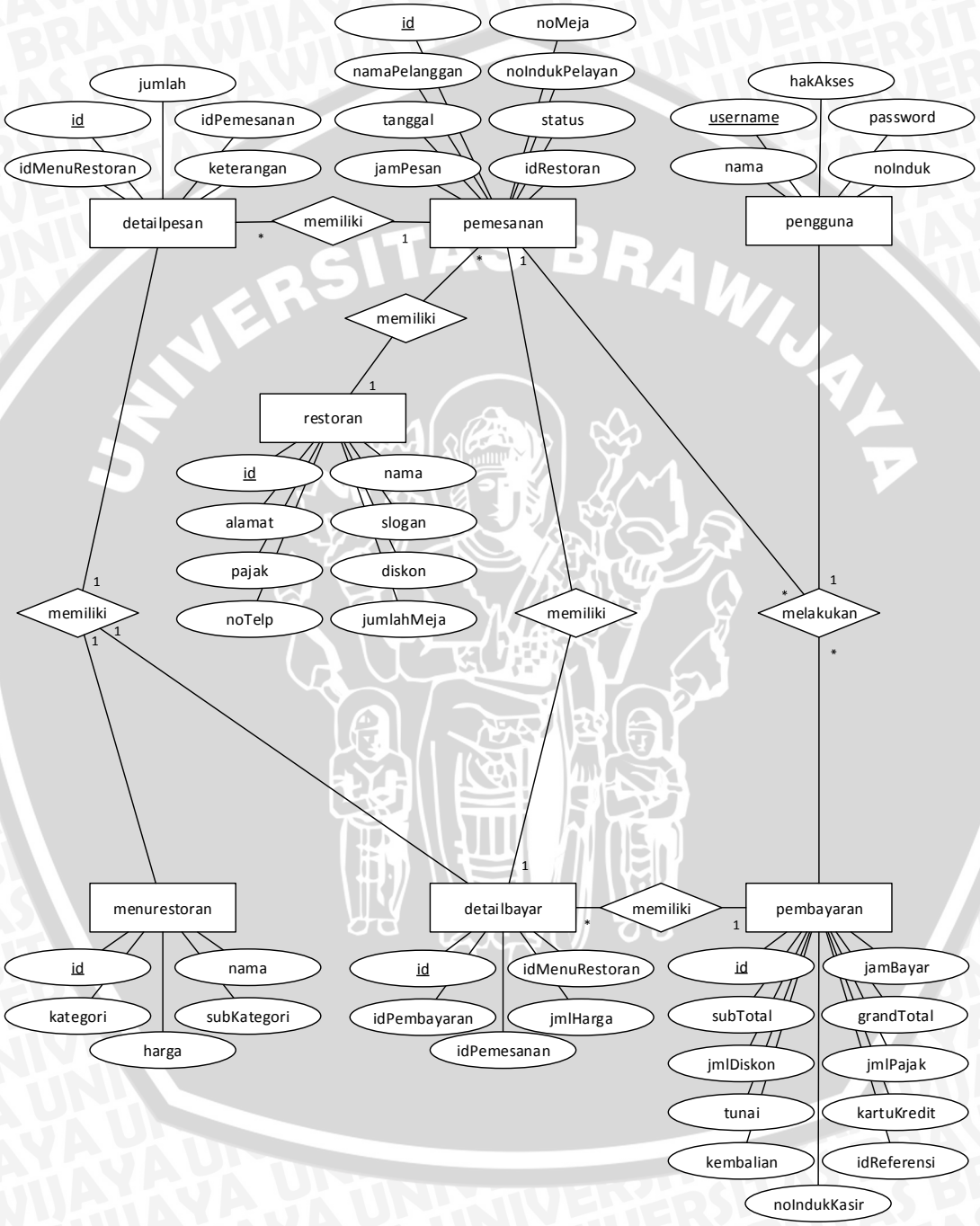
UNIVERSITAS BRAWIJAYA
repository.ub.ac.id



Gambar 4.9 Diagram Kelas *Mode*l OtseResto Level Implementasi

4.3.3 Perancangan Basis Data

Berdasarkan diagram kelas level analisis yang telah dimodelkan sebelumnya maka dapat dimodelkan bentuk *Entity Relationship Diagram* (ERD) dari sistem ini. Gambar 4.10 memperlihatkan model ERD tersebut.



Gambar 4.10 *Entity Relationship Diagram* (ERD) OtseResto

4.3.4 Perancangan Diagram Aktivitas

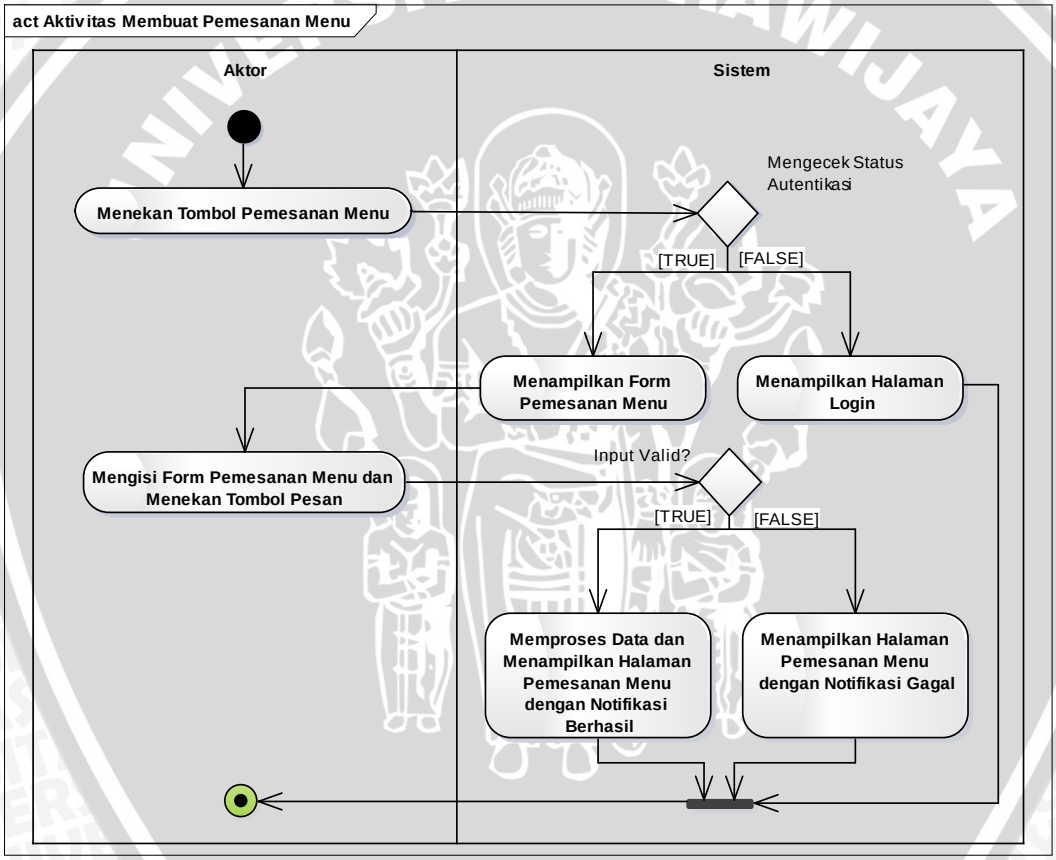
Diagram aktivitas menggambarkan aktivitas yang dapat dilakukan oleh sistem. Diagram ini mengacu kepada skenario *use case* yang telah dibuat sebelumnya.

1. Diagram Aktivitas Area POS Pemesanan

Diagram aktivitas area POS tepatnya area pemesanan menggambarkan aktivitas yang dilakukan oleh sistem beserta interaksinya dengan aktor. Aktor yang mengoperasikan bagian ini adalah pelayan.

a. Diagram Aktivitas Membuat Pemesanan Menu

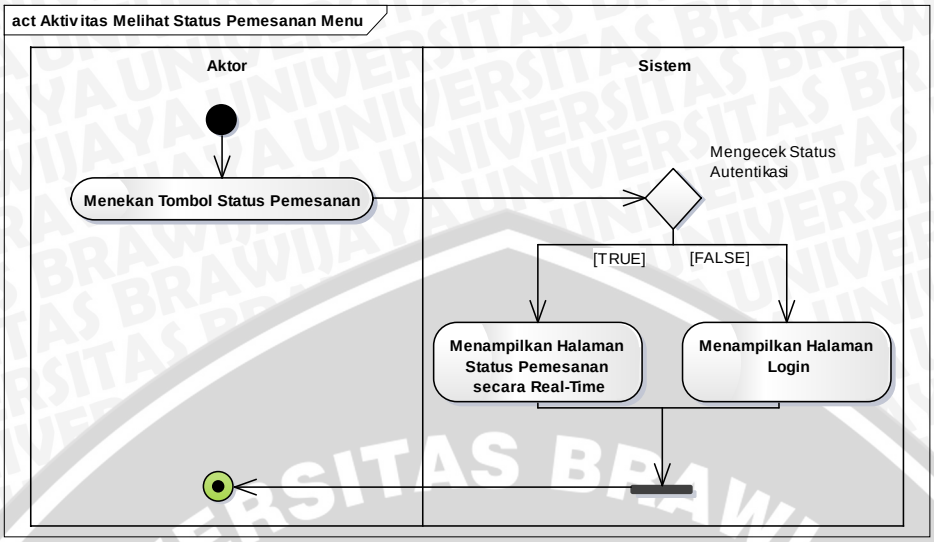
Diagram aktivitas membuat pemesanan menu merupakan aktivitas ketika pelayan ingin membuat pemesanan menu yang baru untuk diproses di dapur. Gambar 4.11 menjelaskan aktivitas sistem dalam membuat pemesanan menu.



Gambar 4.11 Diagram Aktivitas Membuat Pemesanan Menu

b. Diagram Aktivitas Melihat Status Pemesanan Menu

Diagram aktivitas melihat status pemesanan menu merupakan aktivitas ketika pelayan ingin melihat status pemesanan yang telah disiapkan oleh koki. Gambar 4.12 menjelaskan aktivitas sistem untuk melihat status pemesanan menu.



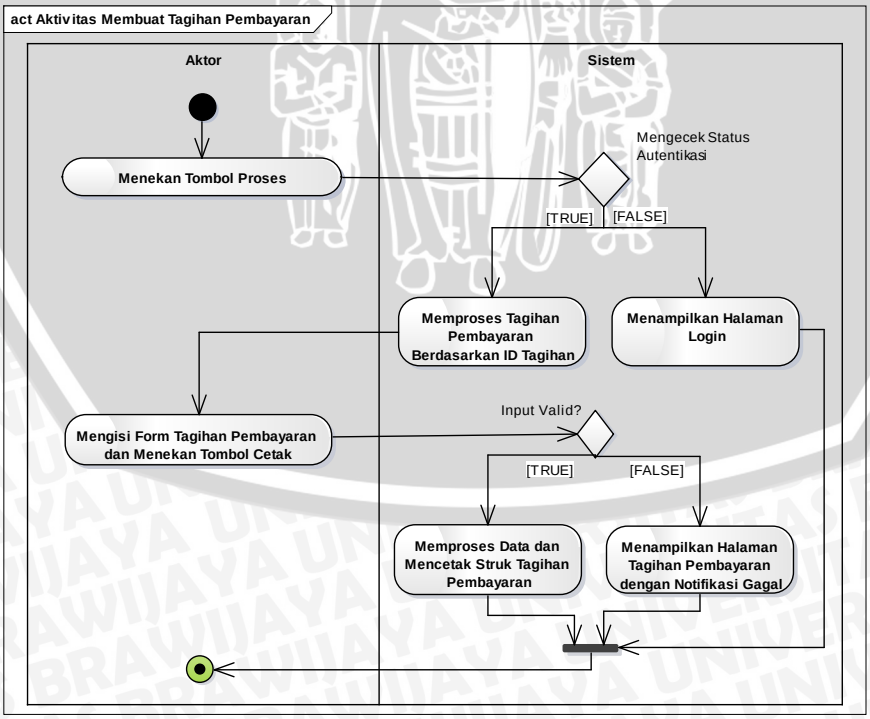
Gambar 4.12 Diagram Aktivitas Melihat Status Pemesanan Menu

2. Diagram Aktivitas Area POS Pembayaran

Diagram aktivitas area POS pembayaran menggambarkan aktivitas yang dilakukan oleh sistem beserta interaksinya dengan aktor. Aktor yang mengoperasikan bagian ini adalah kasir.

a. Diagram Aktivitas Membuat Tagihan Pembayaran

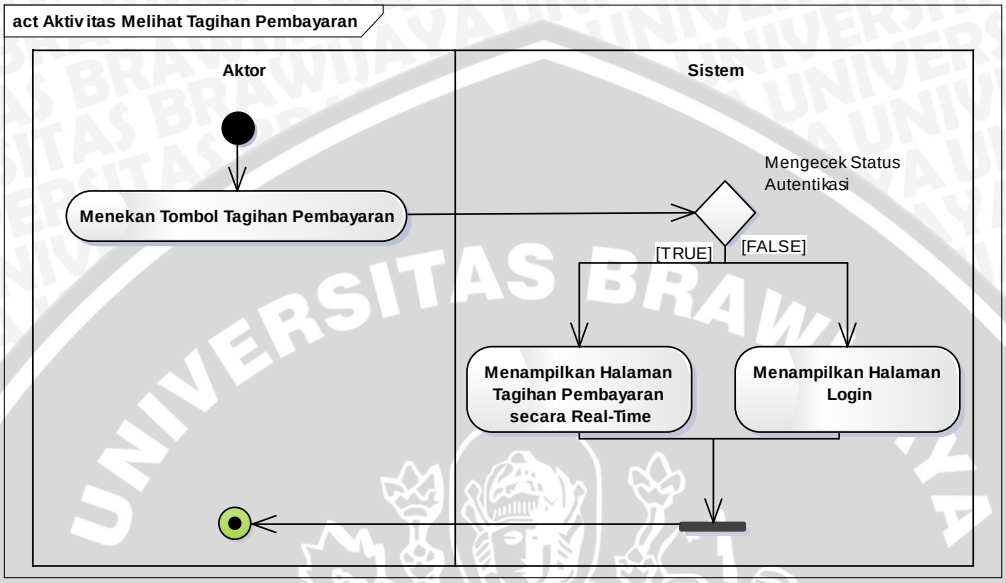
Diagram aktivitas membuat tagihan pembayaran merupakan aktivitas ketika kasir ingin membuat tagihan pembayaran jika ada tamu restoran yang ingin membayar tagihannya. Gambar 4.13 menjelaskan aktivitas sistem untuk membuat tagihan pembayaran.



Gambar 4.13 Diagram Aktivitas Membuat Tagihan Pembayaran

b. Diagram Aktivitas Melihat Tagihan Pembayaran

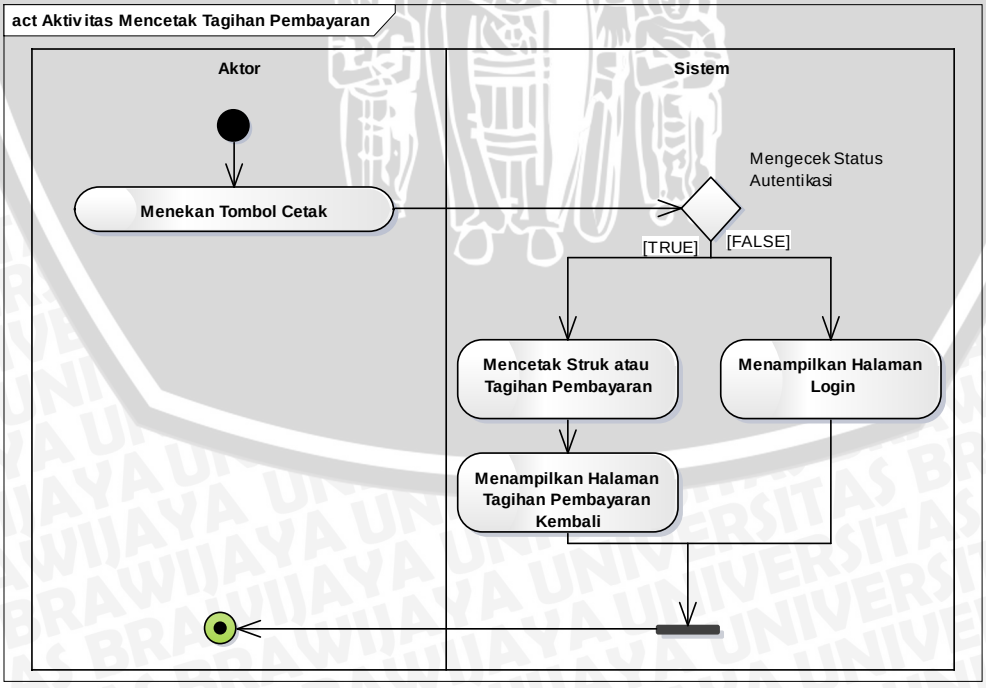
Diagram aktivitas melihat tagihan pembayaran merupakan aktivitas ketika kasir ingin melihat daftar tagihan pembayaran yang yang belum dibayar. Gambar 4.14 menjelaskan aktivitas sistem untuk melihat daftar tagihan pembayaran.



Gambar 4.14 Diagram Aktivitas Melihat Tagihan Pembayaran

c. Diagram Aktivitas Mencetak Tagihan Pembayaran

Diagram aktivitas mencetak tagihan pembayaran merupakan aktivitas ketika kasir ingin mencetak ulang struk atau tagihan pembayaran. Gambar 4.15 menjelaskan aktivitas sistem untuk mencetak tagihan pembayaran.



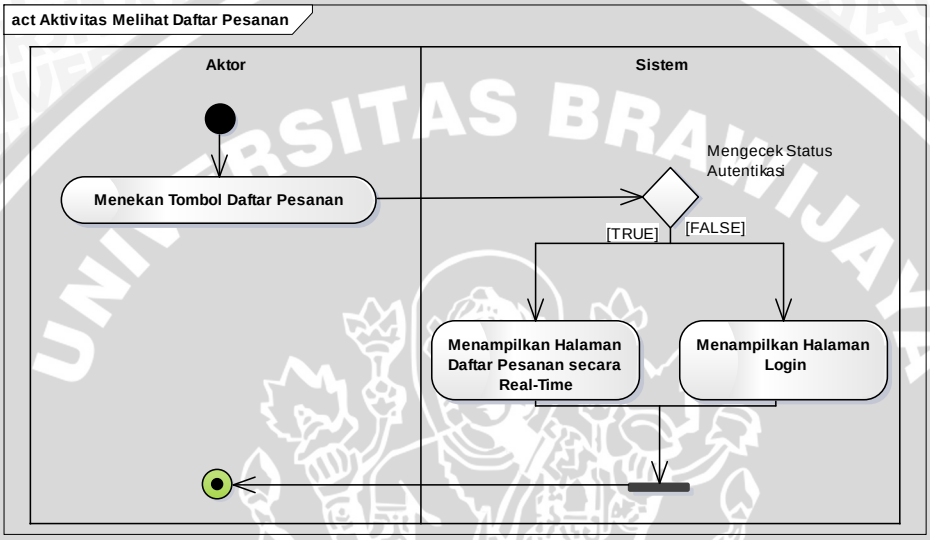
Gambar 4.15 Diagram Aktivitas Mencetak Tagihan Pembayaran

3. Diagram Aktivitas Area Dapur

Diagram aktivitas area dapur menggambarkan aktivitas yang dilakukan oleh sistem beserta interaksinya dengan aktor. Aktor yang mengoperasikan area dapur adalah koki.

a. Diagram Aktivitas Melihat Daftar Pesanan

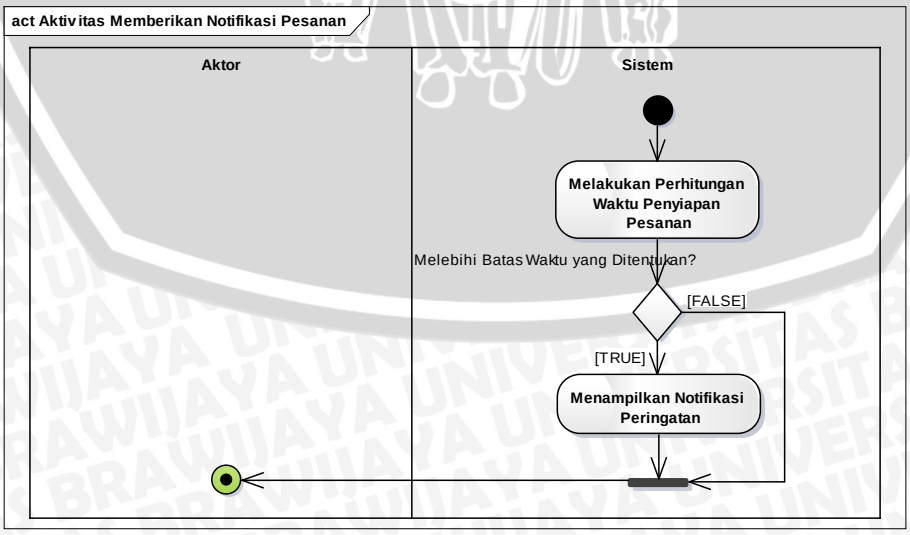
Diagram aktivitas melihat daftar pesanan merupakan aktivitas ketika koki ingin melihat daftar pesanan menu restoran yang sedang disiapkan. Gambar 4.16 menjelaskan aktivitas sistem untuk melihat daftar pesanan.



Gambar 4.16 Diagram Aktivitas Melihat Daftar Pesanan

b. Diagram Aktivitas Notifikasi Pesanan

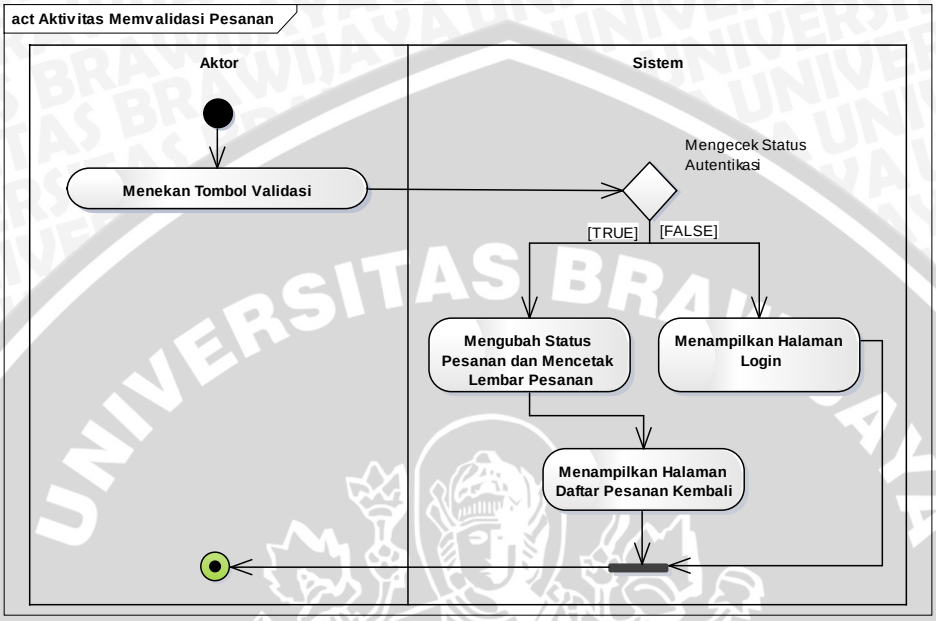
Diagram aktivitas notifikasi pesanan merupakan aktivitas ketika sistem memberikan notifikasi terhadap koki terkait dengan pesanan yang harus diprioritaskan untuk disiapkan. Gambar 4.17 menjelaskan aktivitas sistem untuk memberikan notifikasi pesanan.



Gambar 4.17 Diagram Aktivitas Notifikasi Pesanan

c. Diagram Aktivitas Memvalidasi Pesanan

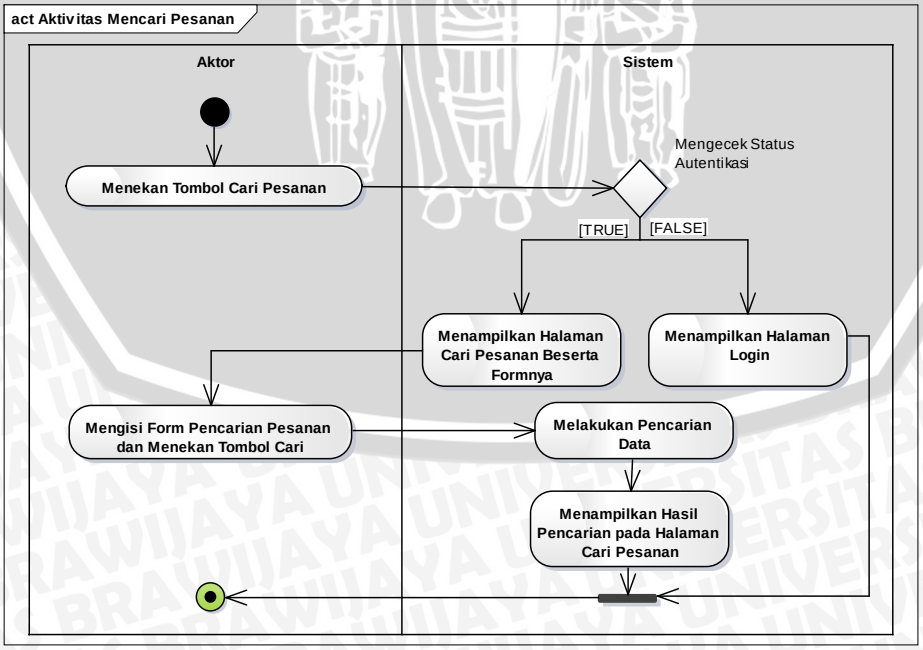
Diagram aktivitas mevalidasi pesanan merupakan aktivitas ketika koki telah selesai menyiapkan suatu pesanan dan mengubah status pesanan tersebut. Gambar 4.18 menjelaskan aktivitas sistem untuk memvalidasi pesanan.



Gambar 4.18 Diagram Aktivitas Memvalidasi Pesanan

d. Diagram Aktivitas Mencari Pesanan

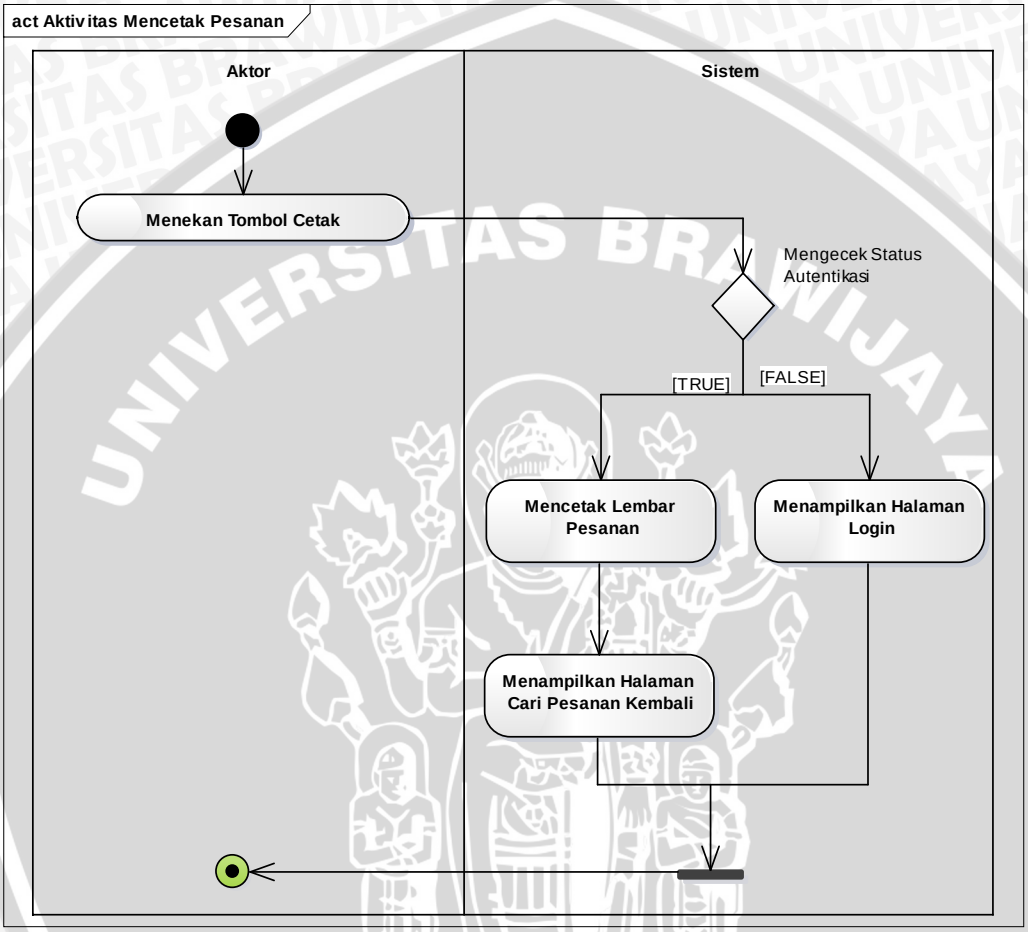
Diagram aktivitas mencari pesanan merupakan aktivitas ketika koki ingin mencari pesanan yang lembar pesanannya hilang. Gambar 4.19 menjelaskan aktivitas sistem untuk mencari pesanan.



Gambar 4.19 Diagram Aktivitas Mencari Pesanan

e. Diagram Aktivitas Mencetak Pesanan

Diagram aktivitas mencetak pesanan merupakan aktivitas ketika koki ingin mencetak pesanan yang sudah dicari untuk dicetak ulang karena lembar pesanannya hilang. Gambar 4.20 menjelaskan aktivitas sistem untuk mencetak pesanan.



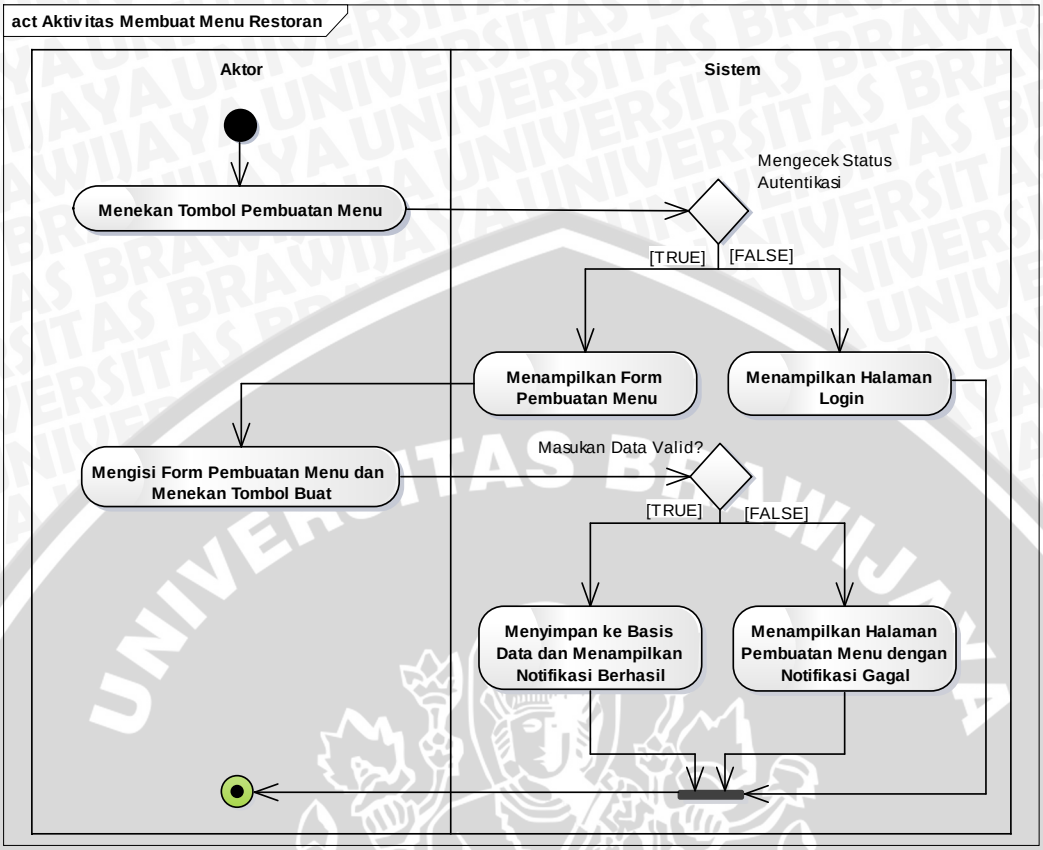
Gambar 4.20 Diagram Aktivitas Mencetak Pesanan

4. Diagram Aktivitas Area Manajerial

Diagram aktivitas area manajerial menggambarkan aktivitas yang dilakukan oleh sistem beserta interaksinya dengan aktor. Aktor yang mengoperasikan area manajerial adalah manajer.

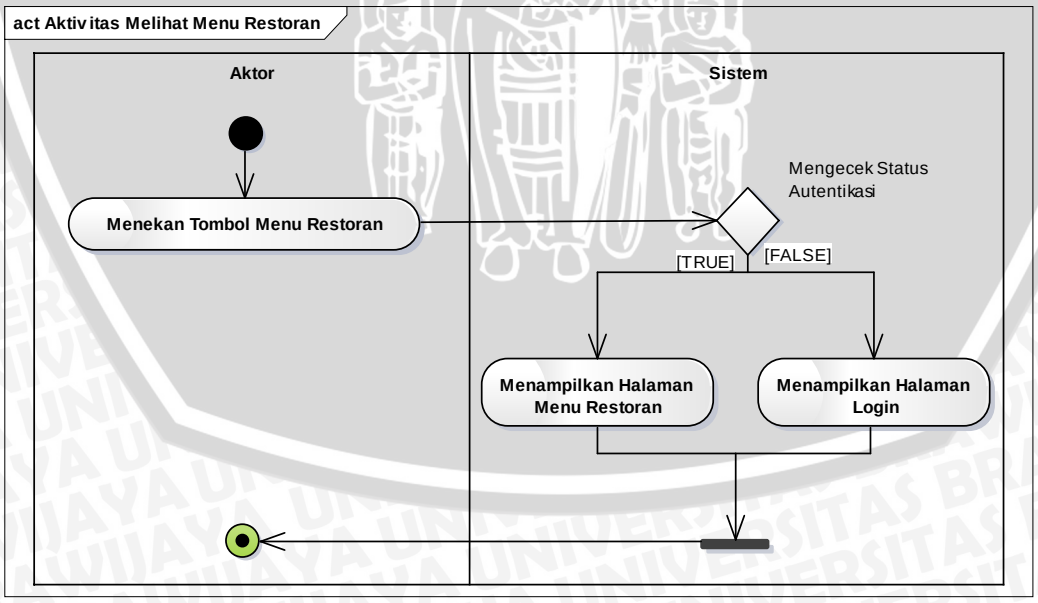
a. Diagram Aktivitas Kelola Menu Restoran

Diagram aktivitas kelola menu restoran merupakan aktivitas ketika manajer ingin mengelola menu restoran yang disediakan oleh restoran. Gambar 4.21 menjelaskan aktivitas sistem untuk membuat menu makanan atau minuman yang tersedia di restoran.



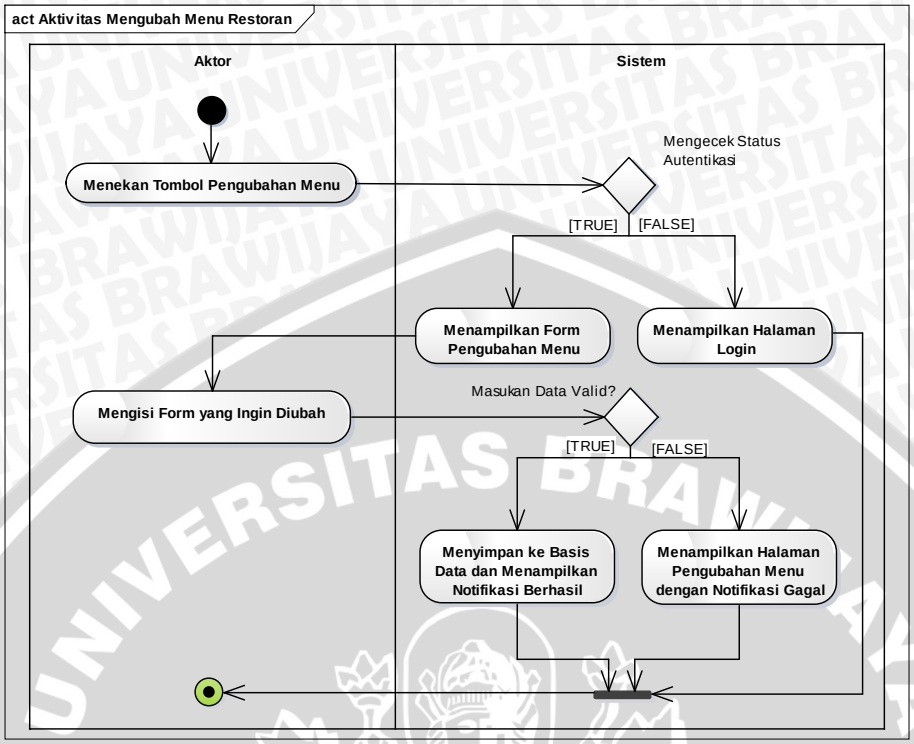
Gambar 4.21 Diagram Aktivitas Membuat Menu Restoran

Gambar 4.22 menjelaskan aktivitas sistem untuk melihat daftar menu makanan dan minuman yang tersedia di restoran.



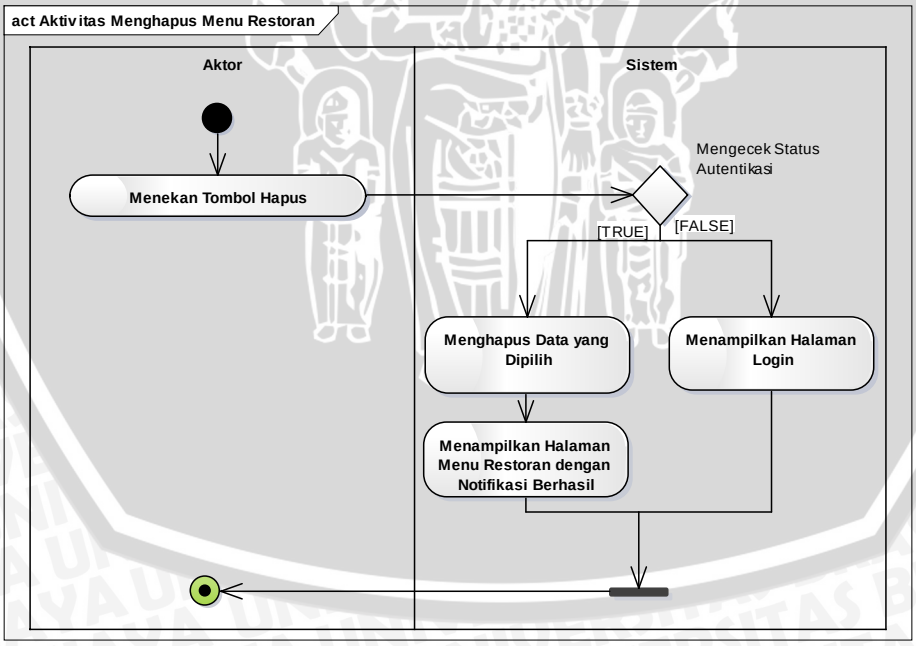
Gambar 4.22 Diagram Aktivitas Melihat Menu Restoran

Gambar 4.23 menggambarkan aktivitas sistem untuk mengubah menu makanan atau minuman yang tersedia di restoran.



Gambar 4.23 Diagram Aktivitas Mengubah Menu Restoran

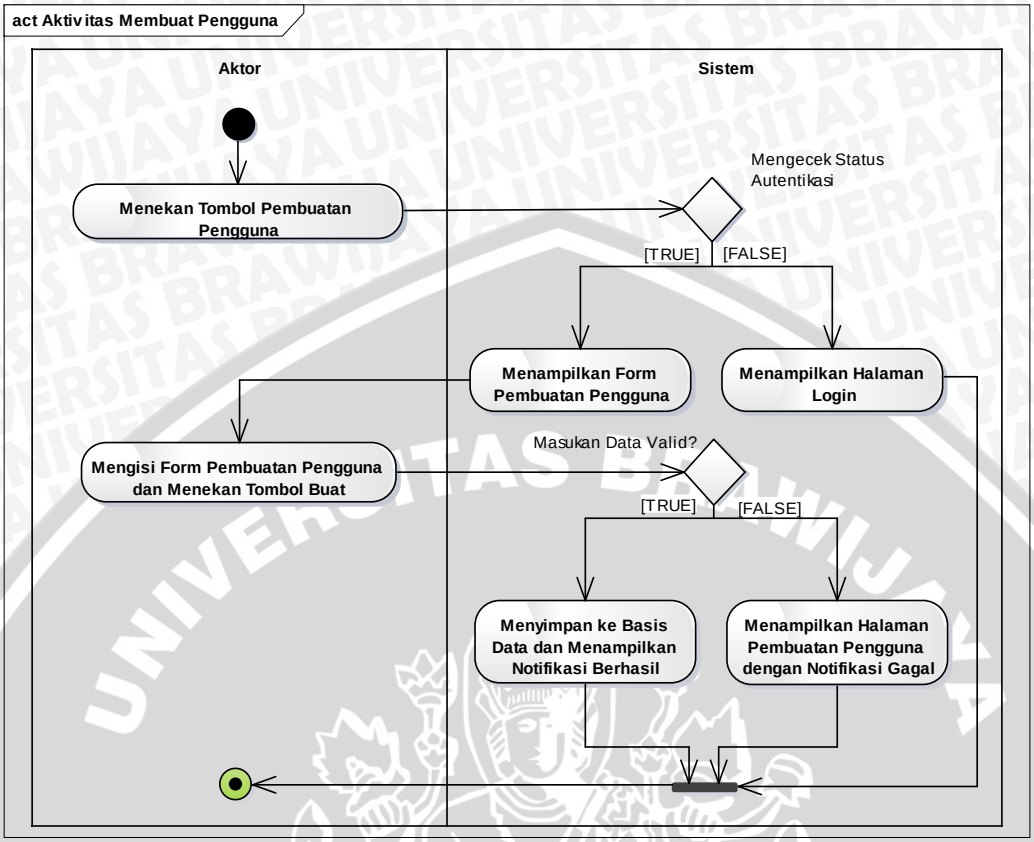
Gambar 4.24 menggambarkan aktivitas sistem untuk menghapus menu makanan atau minuman yang tersedia di restoran.



Gambar 4.24 Diagram Aktivitas Menghapus Menu Restoran

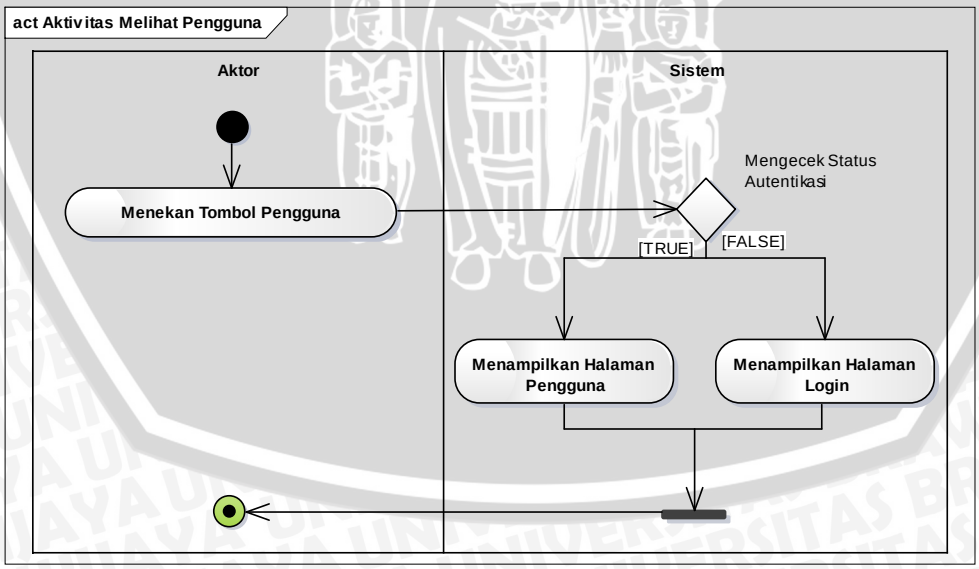
b. Diagram Aktivitas Kelola Pengguna

Diagram aktivitas kelola pengguna merupakan aktivitas ketika manajer ingin mengelola pengguna yang dapat mengakses sistem. Gambar 4.25 menjelaskan aktivitas sistem untuk membuat pengguna.



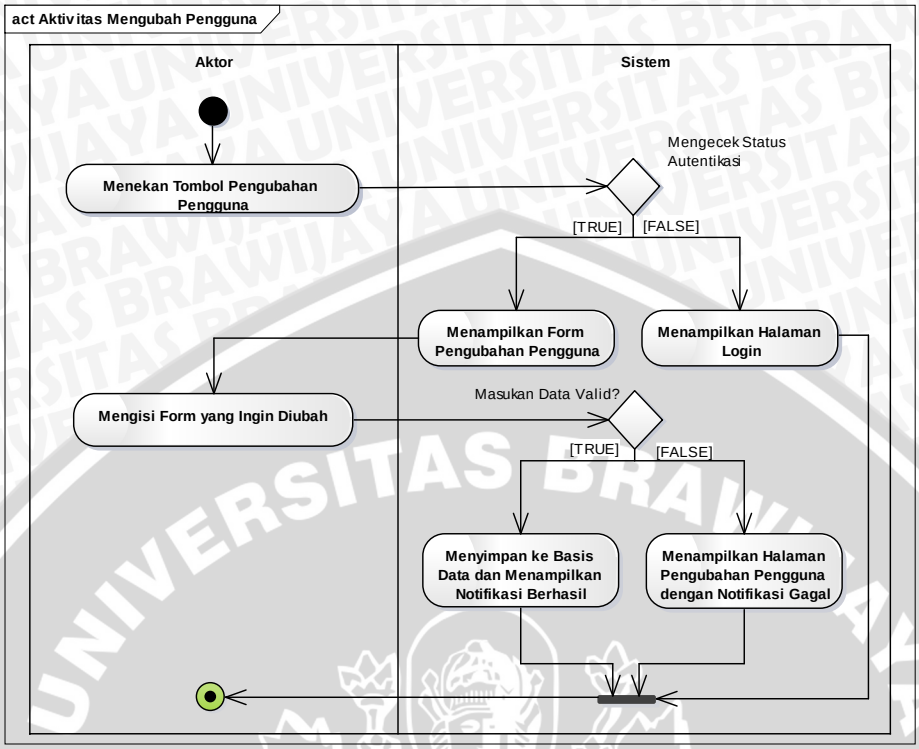
Gambar 4.25 Diagram Aktivitas Membuat Pengguna

Gambar 4.26 menggambarkan aktivitas sistem terkait melihat daftar pengguna yang dapat mengakses sistem.



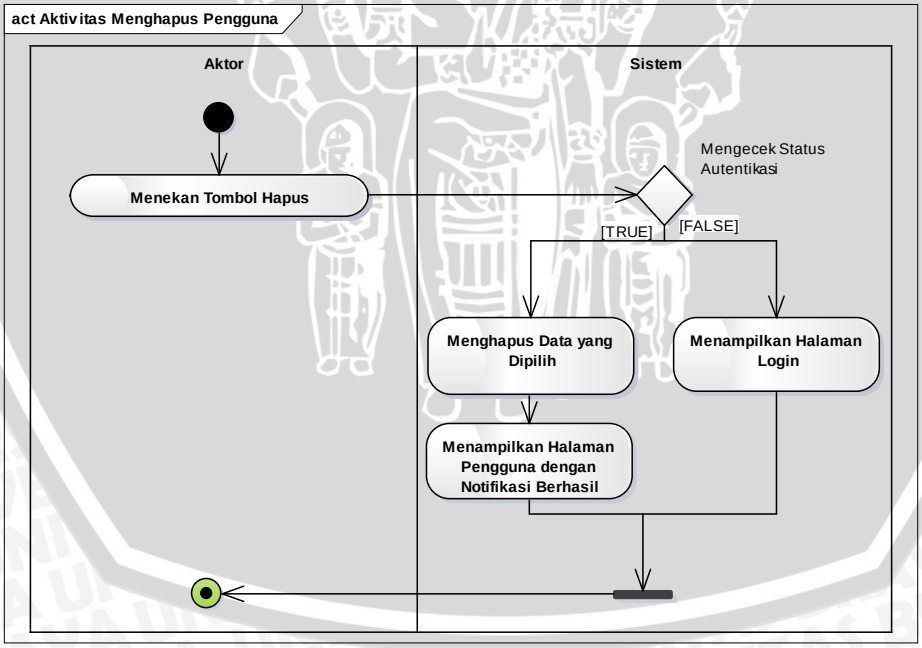
Gambar 4.26 Diagram Aktivitas Melihat Pengguna

Gambar 4.27 menggambarkan aktivitas sistem untuk mengubah pengguna.



Gambar 4.27 Diagram Aktivitas Mengubah Pengguna

Gambar 4.28 menggambarkan aktivitas sistem untuk menghapus pengguna.

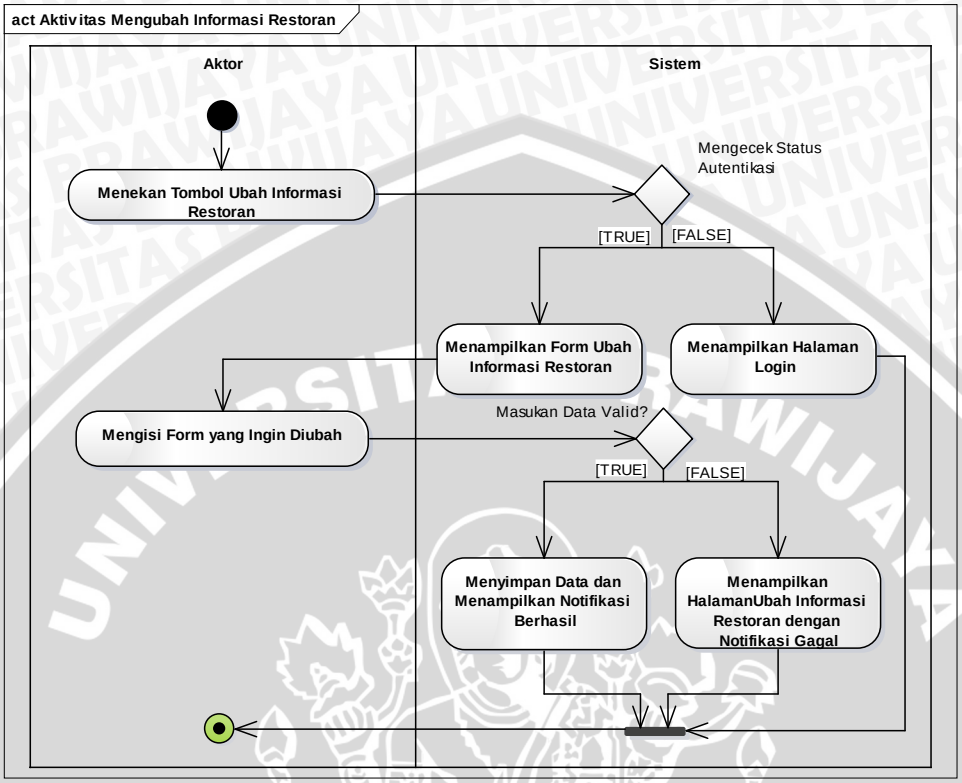


Gambar 4.28 Diagram Aktivitas Menghapus Pengguna

c. Diagram Aktivitas Mengubah Informasi Restoran

Diagram aktivitas mengubah informasi restoran merupakan aktivitas ketika manajer ingin mengubah informasi-informasi umum terkait restoran beserta pajak

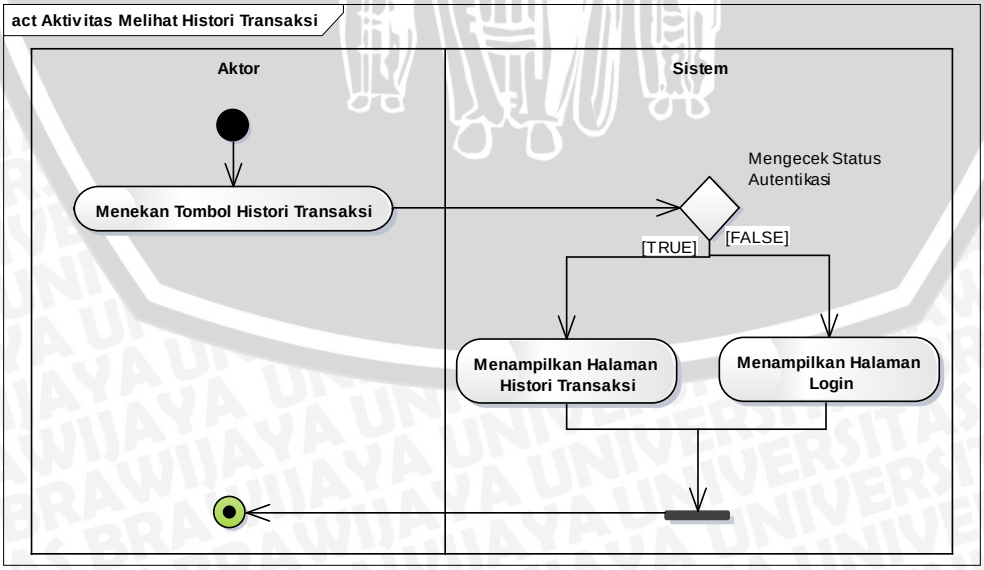
dan diskon yang diterapkan. Gambar 4.29 menjelaskan aktivitas sistem untuk mengubah informasi restoran.



Gambar 4.29 Diagram Aktivitas Mengubah Informasi Restoran

d. Diagram Aktivitas Melihat Histori Transaksi

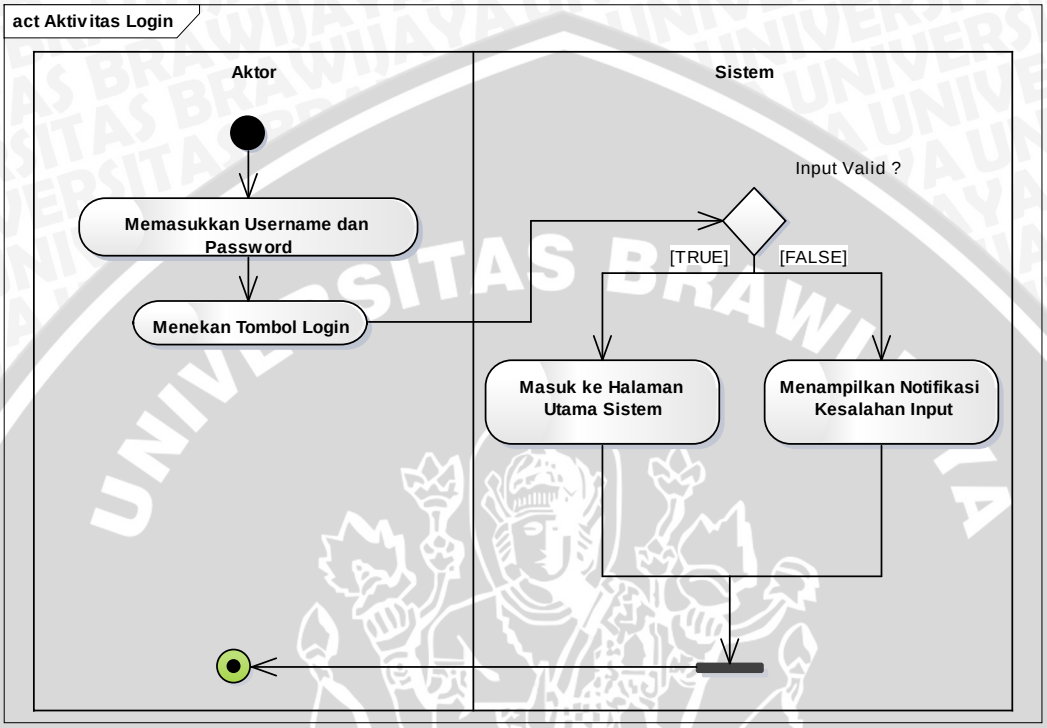
Diagram aktivitas melihat histori transaksi merupakan aktivitas ketika manajer ingin mengawasi ringkasan histori transaksi yang telah dilakukan oleh kasir. Gambar 4.30 menjelaskan aktivitas sistem untuk melihat histori transaksi.



Gambar 4.30 Diagram Aktivitas Melihat Histori Transaksi

5. Diagram Aktivitas Login

Diagram aktivitas login merupakan aktivitas ketika pengguna yang telah teregistrasi pada sistem ingin masuk ke dalam sistem. Gambar 4.31 menjelaskan aktivitas sistem untuk aktivitas login.



Gambar 4.31 Diagram Aktivitas Login

4.3.5 Perancangan Diagram Sekuen

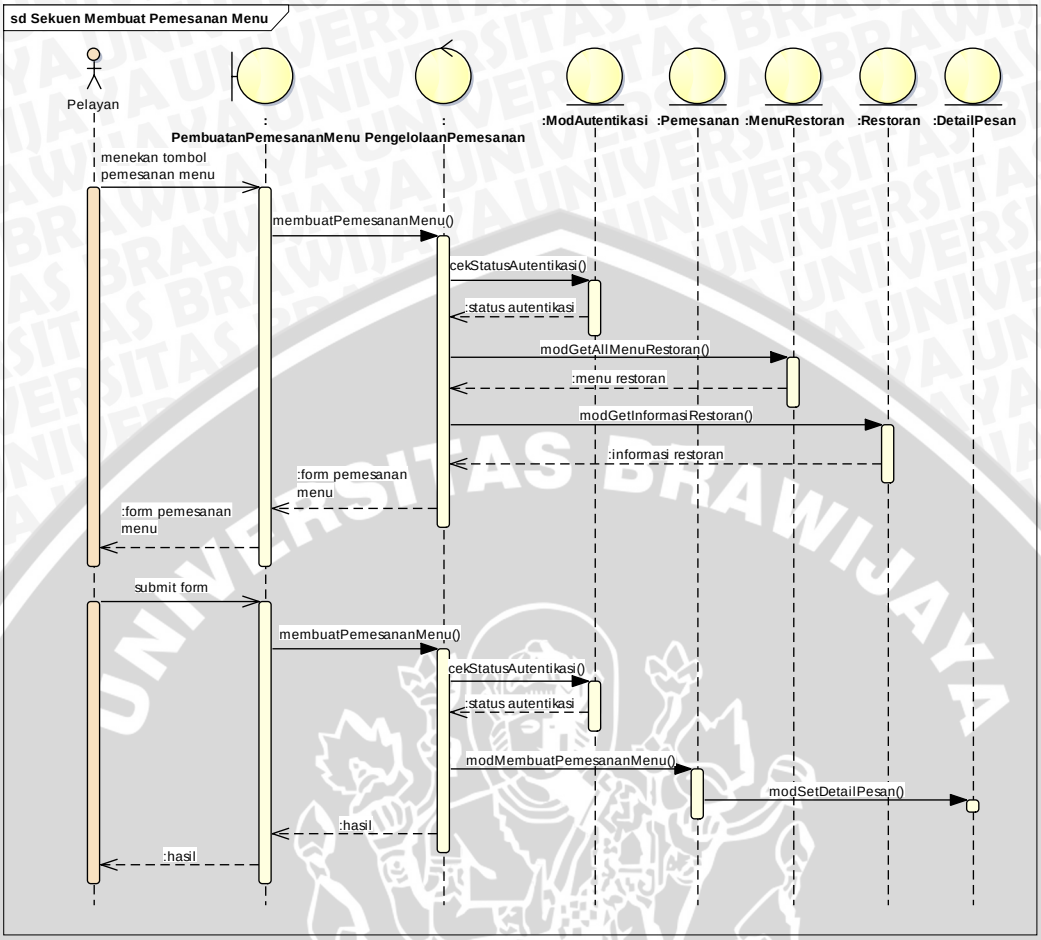
Diagram sekuen merepresentasikan model alir jalannya proses interaksi antar objek. Pemodelan diagram sekuen ini mengacu kepada beberapa diagram yang telah dibuat sebelumnya seperti operasi yang ada pada kelas diagram, skenario *use case*, dan diagram aktivitas.

1. Diagram Sekuen Area POS Pemesanan

Diagram sekuen area POS pemesanan menggambarkan urutan interaksi antar objek pada area POS tepatnya pemesanan. Aktor yang berinteraksi di sini adalah pelayan.

a. Diagram Sekuen Membuat Pemesanan Menu

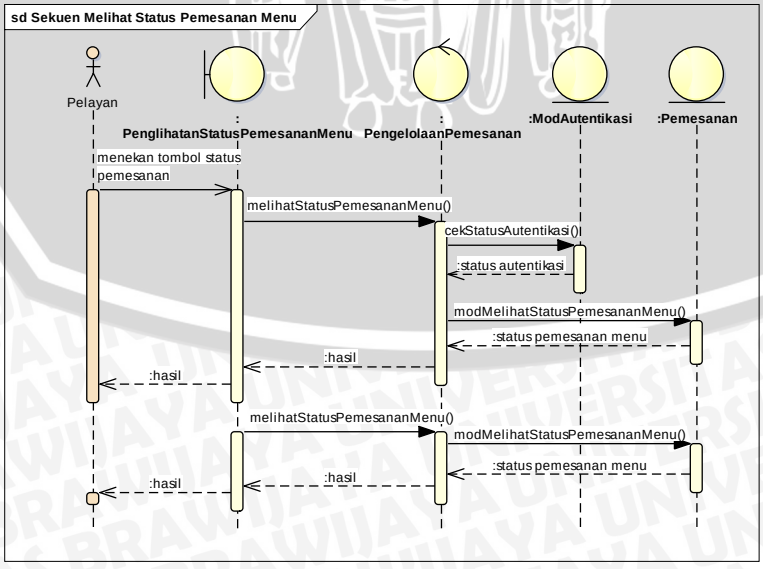
Gambar 4.32 menjelaskan urutan interaksi objek sistem untuk membuat pemesanan menu makanan dan minuman yang tersedia pada restoran.



Gambar 4.32 Diagram Sekuen Membuat Pemesanan Menu

b. Diagram Sekuen Melihat Status Pemesanan Menu

Gambar 4.33 menjelaskan urutan interaksi objek sistem untuk melihat status dari pemesanan menu yang telah disiapkan oleh koki.



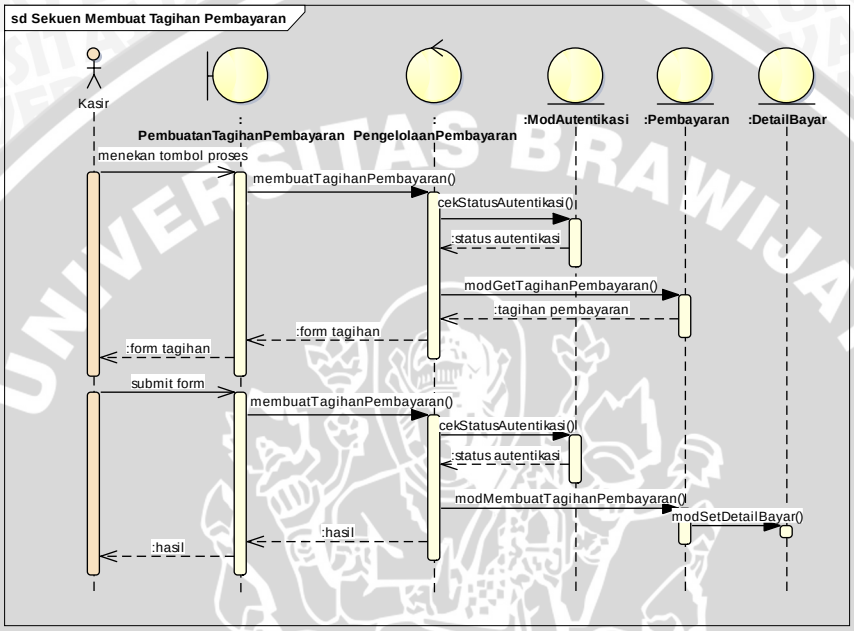
Gambar 4.33 Diagram Sekuen Melihat Status Pemesanan Menu

2. Diagram Sekuen Area POS Pembayaran

Diagram sekuen area POS pembayaran menggambarkan urutan interaksi antar objek pada area POS tepatnya pembayaran. Aktor yang berinteraksi di sini adalah kasir.

a. Diagram Sekuen Membuat Tagihan Pembayaran

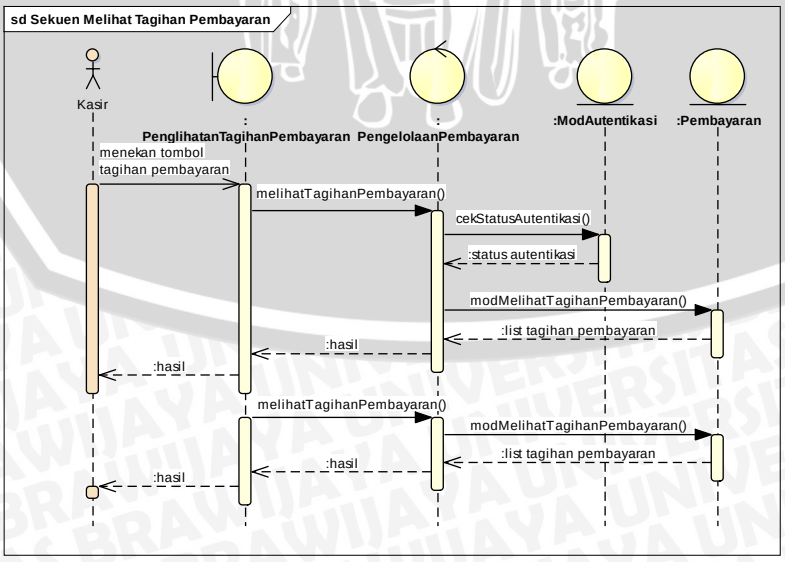
Gambar 4.34 menjelaskan urutan interaksi objek sistem untuk membuat tagihan pembayaran.



Gambar 4.34 Diagram Sekuen Membuat Tagihan Pembayaran

b. Diagram Sekuen Melihat Tagihan Pembayaran

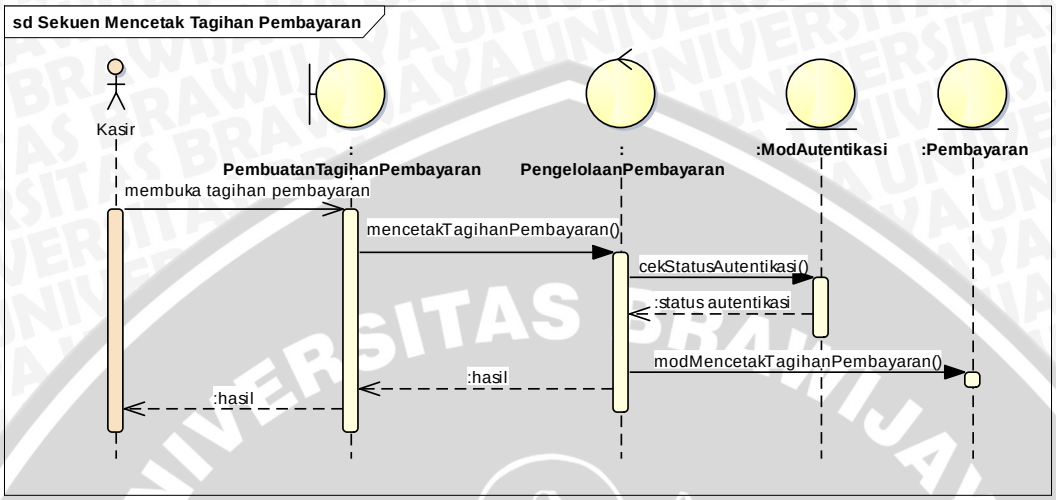
Gambar 4.35 menjelaskan urutan interaksi objek sistem untuk melihat tagihan pembayaran.



Gambar 4.35 Diagram Sekuen Melihat Tagihan Pembayaran

c. Diagram Sekuen Mencetak Tagihan Pembayaran

Gambar 4.36 menjelaskan urutan interaksi objek sistem untuk mencetak tagihan pembayaran.



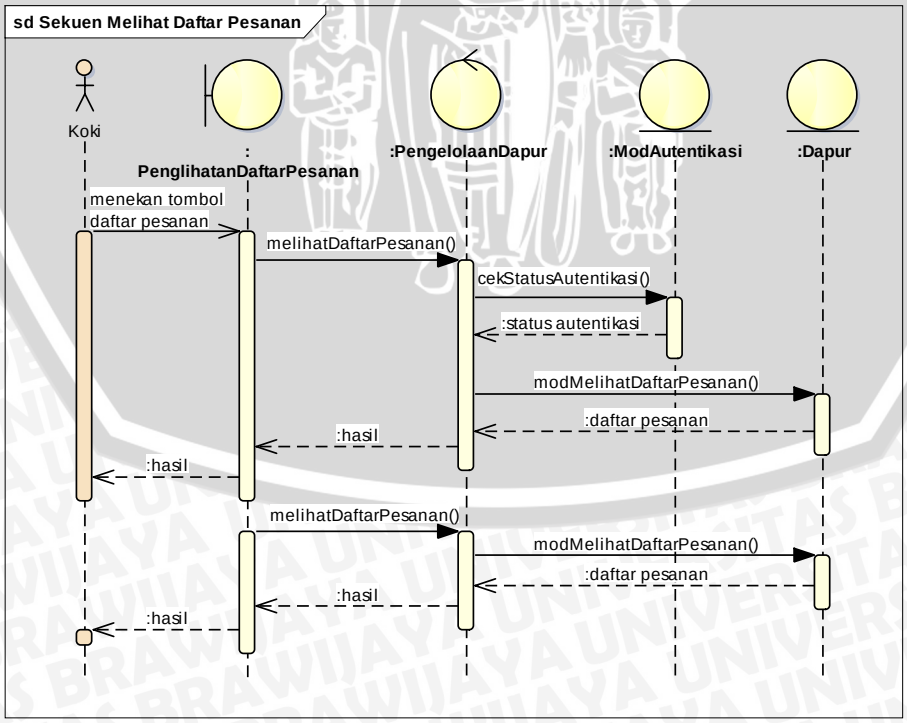
Gambar 4.36 Diagram Sekuen Mencetak Tagihan Pembayaran

3. Diagram Sekuen Area Dapur

Diagram sekuen area dapur menggambarkan urutan interaksi antar objek pada area dapur. Aktor yang berinteraksi di sini adalah koki.

a. Diagram Sekuen Melihat Daftar Pesanan

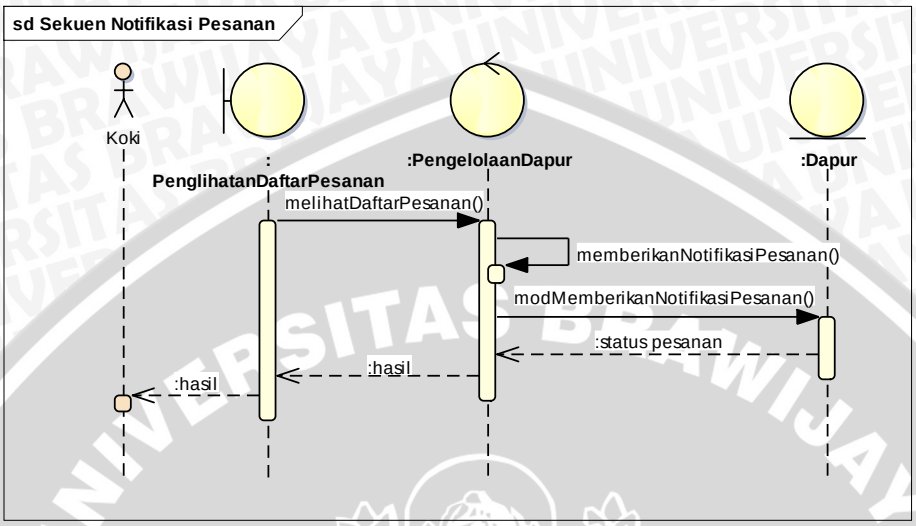
Gambar 4.37 menjelaskan urutan interaksi objek sistem untuk melihat daftar pesanan.



Gambar 4.37 Diagram Sekuen Melihat Daftar Pesanan

b. Diagram Sekuen Notifikasi Pesanan

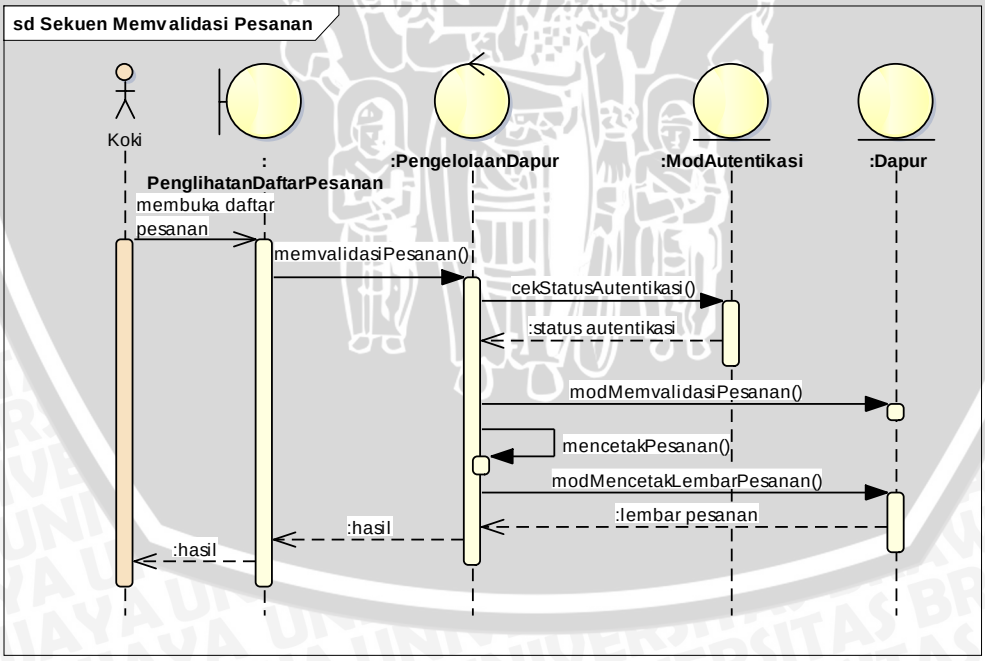
Gambar 4.38 menjelaskan urutan interaksi objek sistem untuk memberikan notifikasi pesanan.



Gambar 4.38 Diagram Sekuen Notifikasi Pesanan

c. Diagram Sekuen Memvalidasi Pesanan

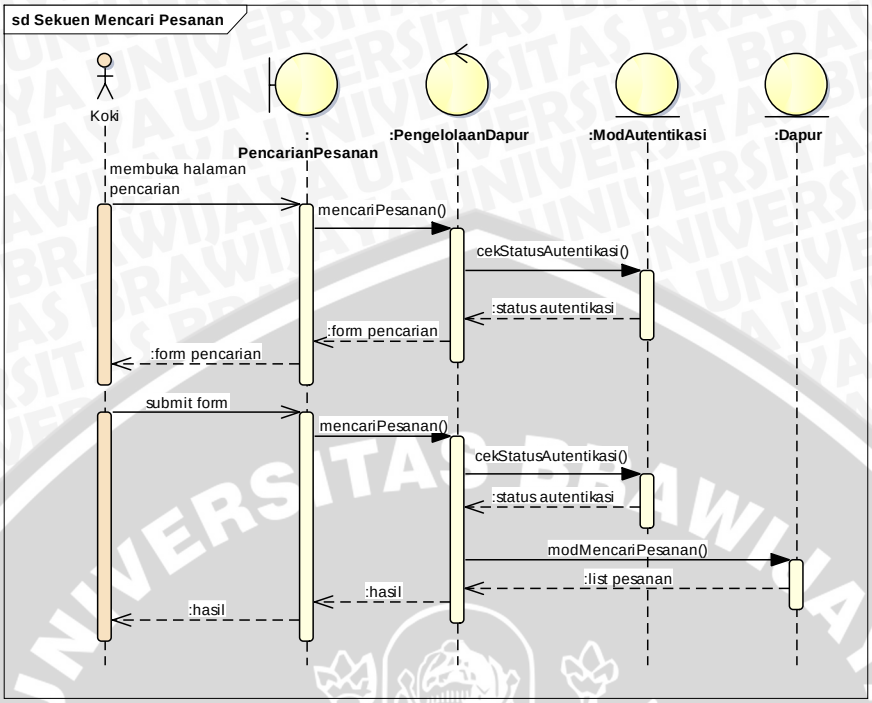
Gambar 4.39 menjelaskan urutan interaksi objek sistem untuk memvalidasi pesanan.



Gambar 4.39 Diagram Sekuen Memvalidasi Pesanan

d. Diagram Sekuen Mencari Pesanan

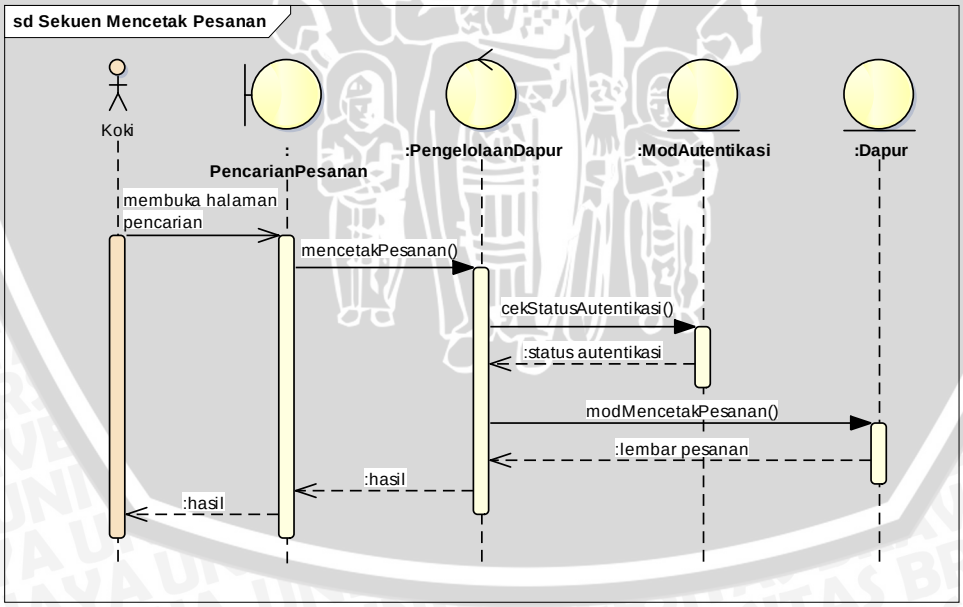
Gambar 4.40 menjelaskan urutan interaksi objek sistem untuk mencari pesanan.



Gambar 4.40 Diagram Sekuen Mencari Pesanan

e. Diagram Sekuen Mencetak Pesanan

Gambar 4.41 menjelaskan urutan interaksi objek sistem untuk mencetak pesanan.



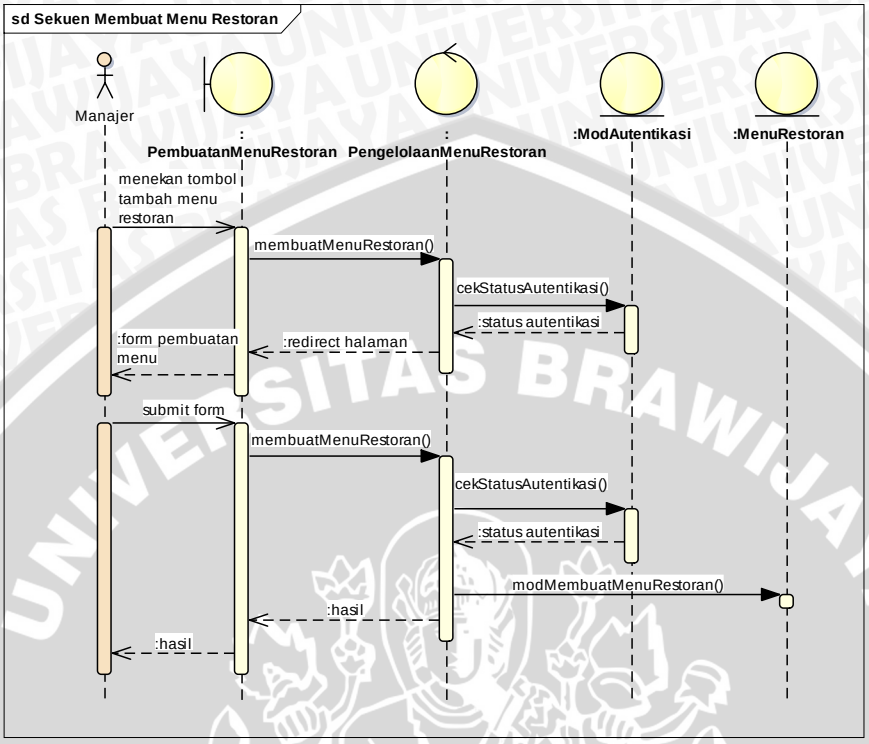
Gambar 4.41 Diagram Sekuen Mencetak Pesanan

4. Diagram Sekuen Area Manajerial

Diagram sekuen area manajerial menggambarkan urutan interaksi antar objek pada area manajerial. Aktor yang berinteraksi di sini adalah manajer.

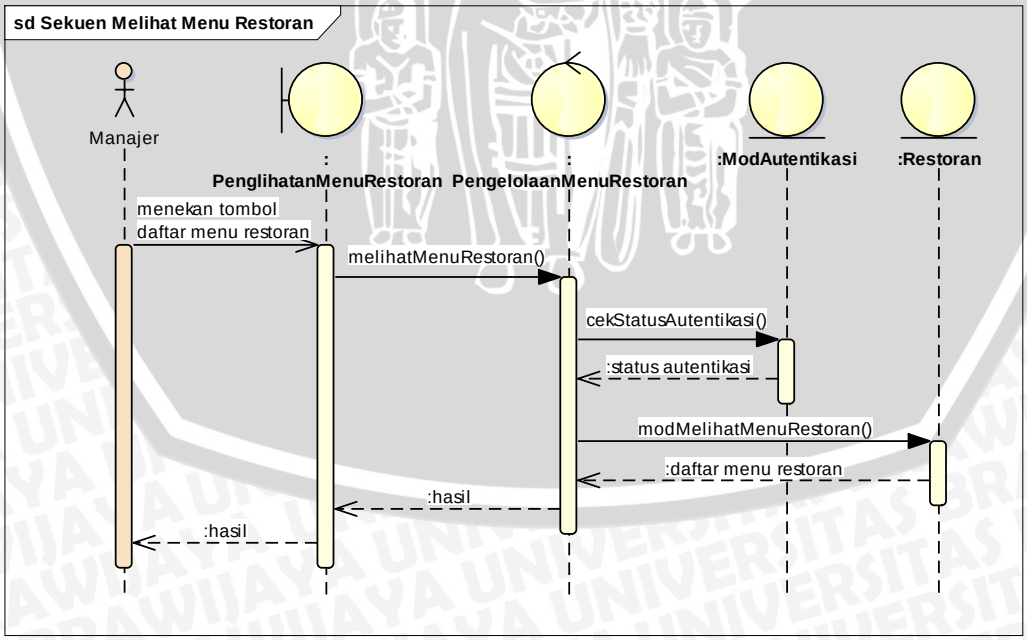
a. Diagram Sekuen Kelola Menu Restoran

Gambar 4.42 menjelaskan urutan interaksi objek sistem untuk membuat menu makanan atau minuman yang tersedia di restoran.



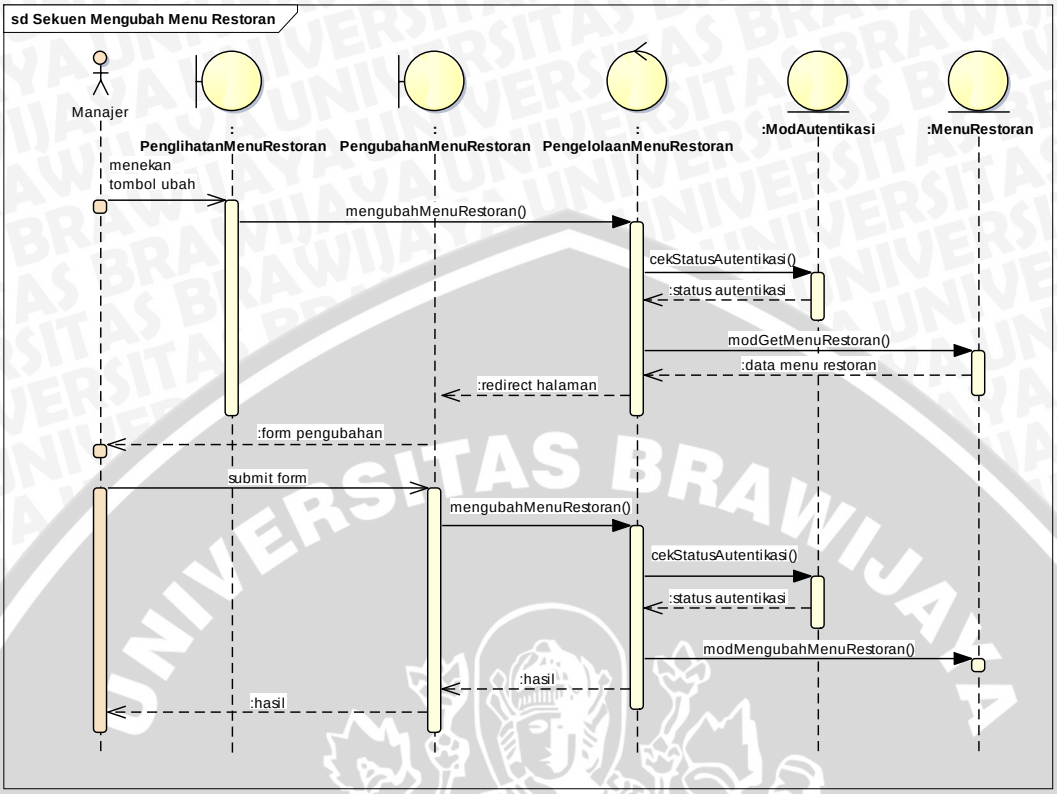
Gambar 4.42 Diagram Sekuen Membuat Menu Restoran

Gambar 4.43 menjelaskan urutan interaksi objek sistem untuk melihat daftar menu makanan dan minuman yang tersedia di restoran.



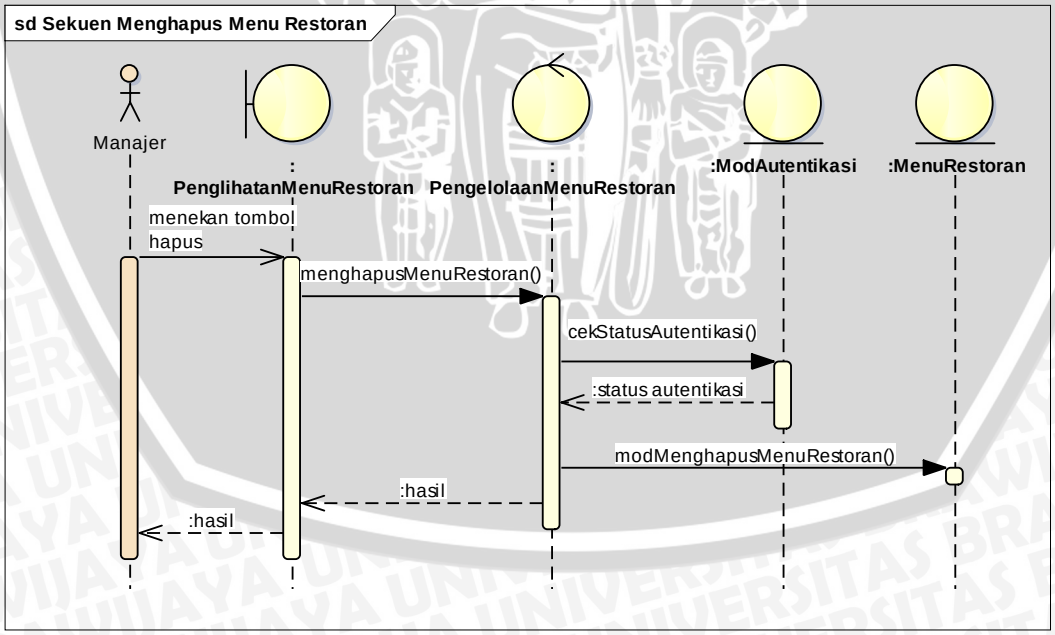
Gambar 4.43 Diagram Sekuen Melihat Menu Restoran

Gambar 4.44 menjelaskan urutan interaksi objek sistem untuk mengubah menu makanan atau minuman yang tersedia di restoran.



Gambar 4.44 Diagram Sekuen Mengubah Menu Restoran

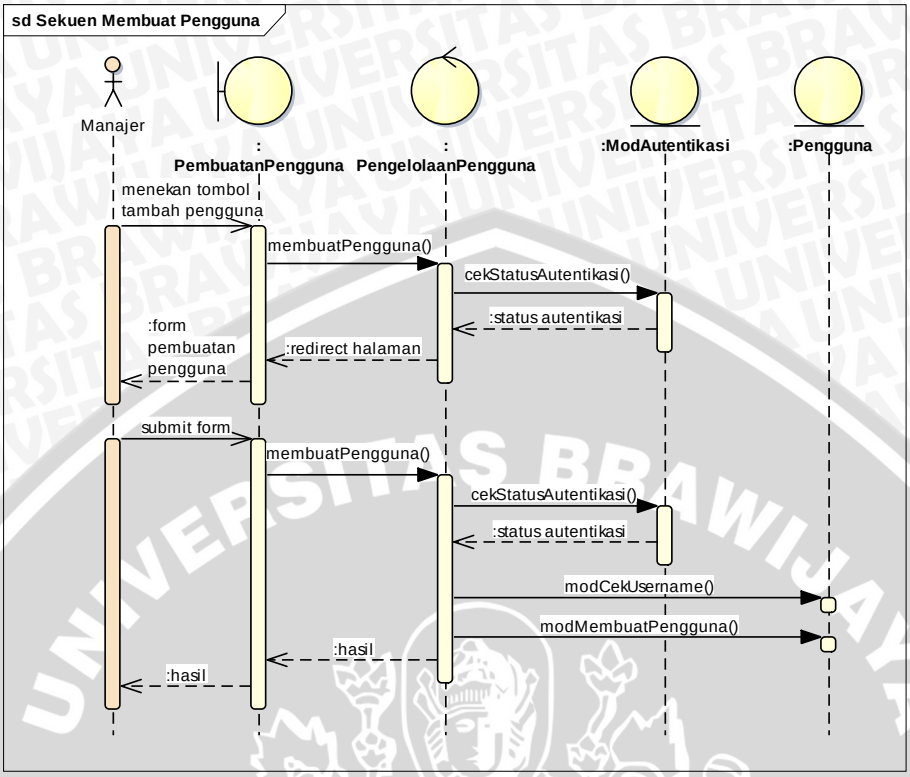
Gambar 4.45 menggambarkan urutan interaksi objek sistem untuk menghapus menu makanan atau minuman yang tersedia di restoran.



Gambar 4.45 Diagram Sekuen Menghapus Menu Restoran

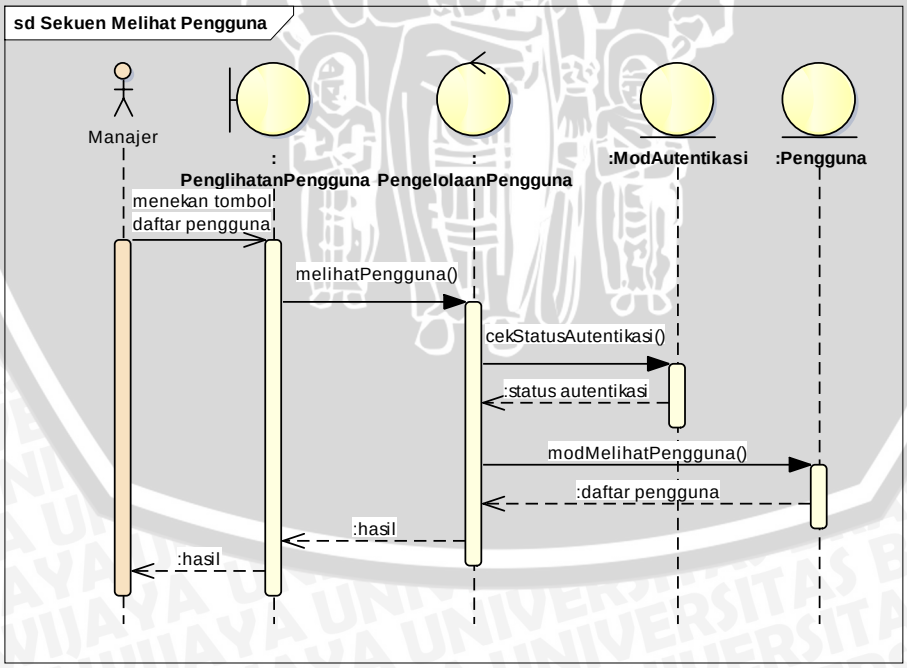
b. Diagram Sekuen Kelola Pengguna

Gambar 4.46 menjelaskan urutan interaksi objek sistem untuk membuat pengguna.



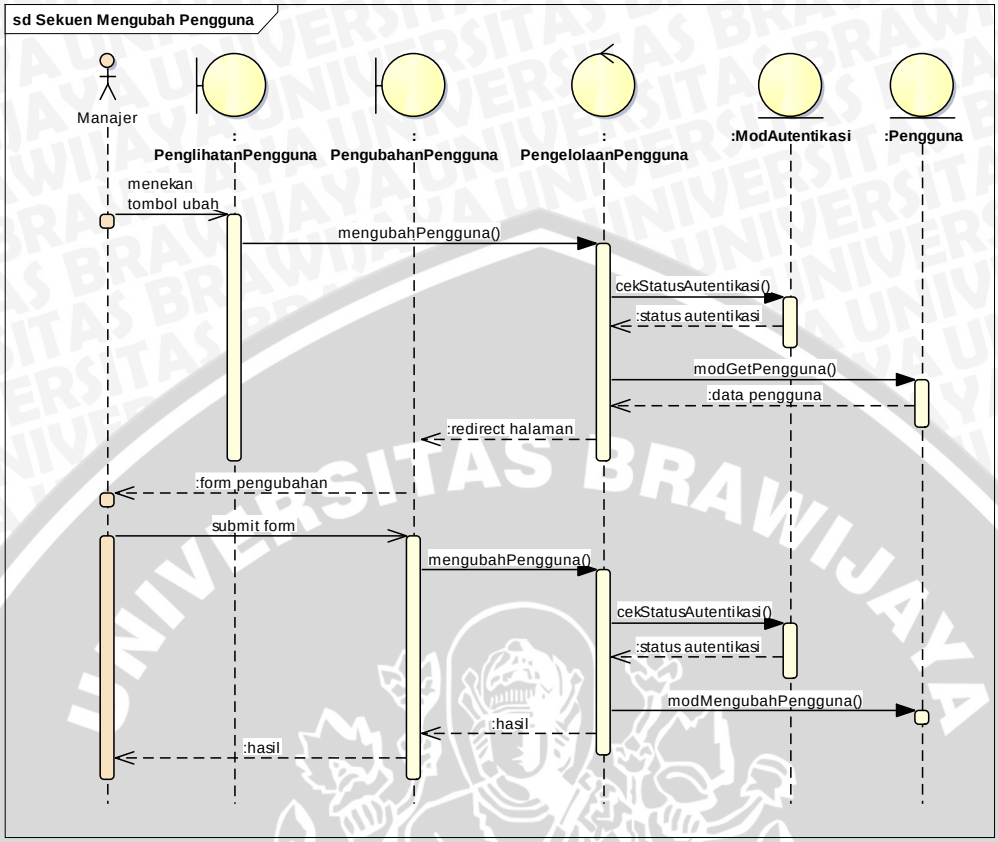
Gambar 4.46 Diagram Sekuen Membuat Pengguna

Gambar 4.47 menjelaskan urutan interaksi objek sistem untuk melihat daftar pengguna.



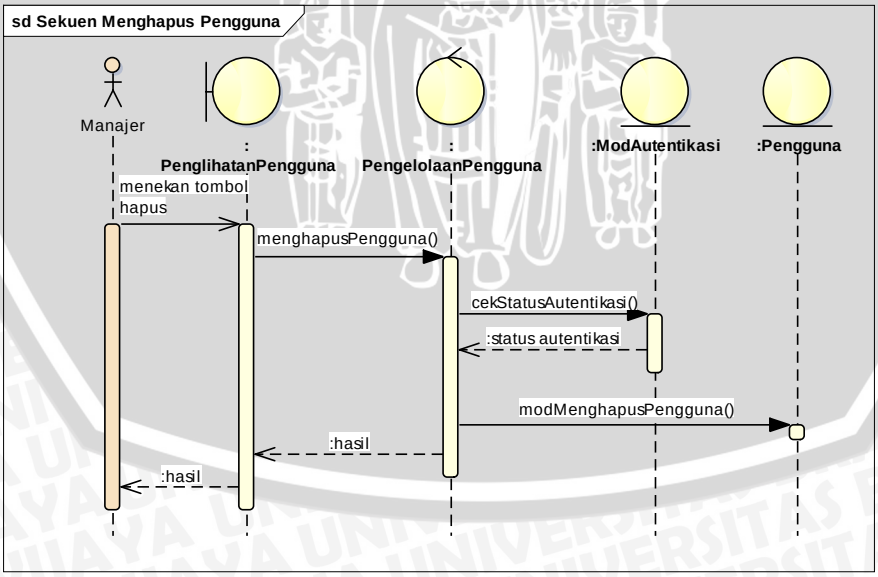
Gambar 4.47 Diagram Sekuen Melihat Pengguna

Gambar 4.48 menjelaskan urutan interaksi objek sistem untuk mengubah pengguna.



Gambar 4.48 Diagram Sekuen Mengubah Pengguna

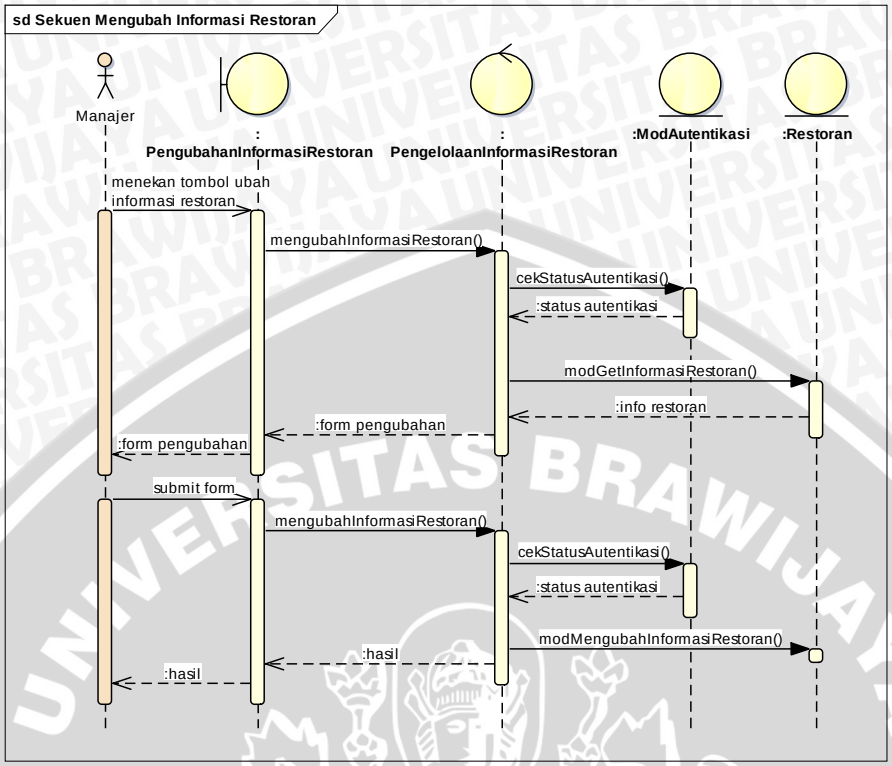
Gambar 4.49 menjelaskan urutan interaksi objek sistem untuk menghapus pengguna.



Gambar 4.49 Diagram Sekuen Menghapus Pengguna

c. Diagram Sekuen Mengubah Informasi Restoran

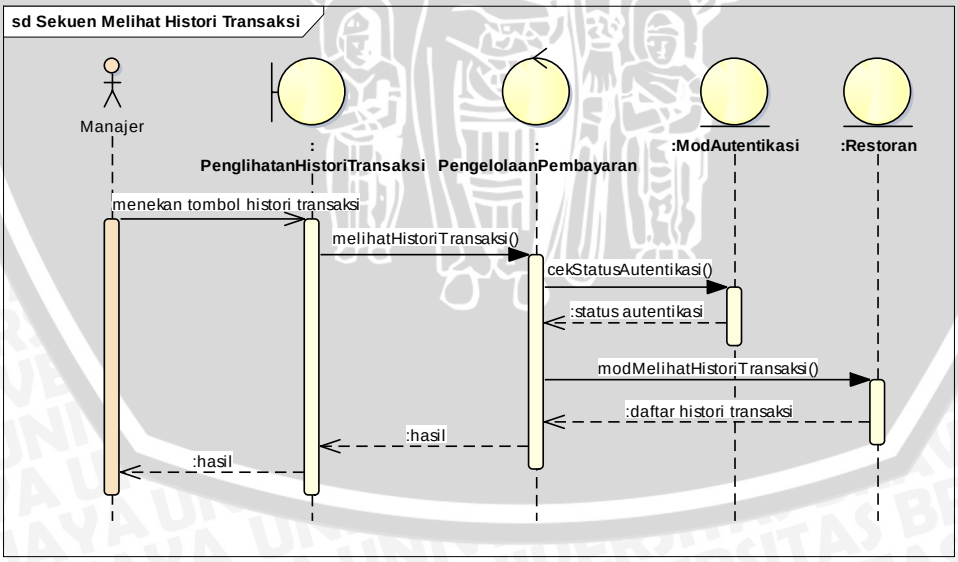
Gambar 4.50 menjelaskan urutan interaksi objek sistem untuk mengubah informasi restoran.



Gambar 4.50 Diagram Sekuen Mengubah Informasi Restoran

d. Diagram Sekuen Melihat Histori Transaksi

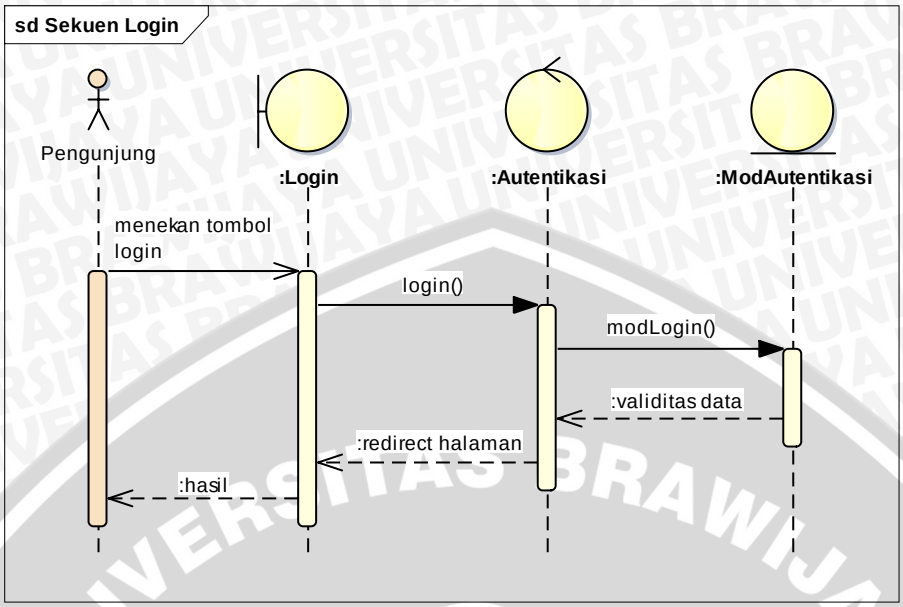
Gambar 4.51 menjelaskan urutan interaksi objek sistem untuk melihat histori transaksi.



Gambar 4.51 Diagram Sekuen Melihat Histori Transaksi

15. Diagram Sekuen Login

Gambar 4.52 memaparkan urutan interaksi objek sistem untuk login.

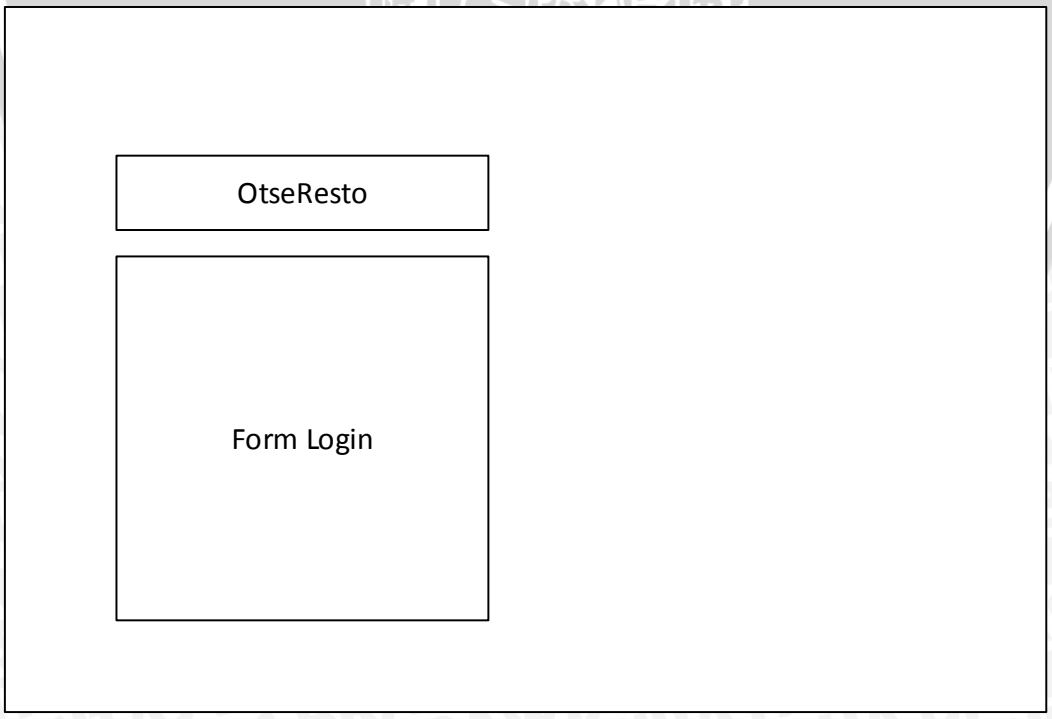


Gambar 4.52 Diagram Sekuen Login

4.3.6 Perancangan Antarmuka

Pada bagian ini akan dijelaskan tentang perancangan antarmuka dari sistem pengelolaan restoran terintegrasi atau OtseResto. Antarmuka yang dimodelkan yaitu antarmuka untuk halaman *login*, halaman utama dari pelayan, kasir, koki dan manajer.

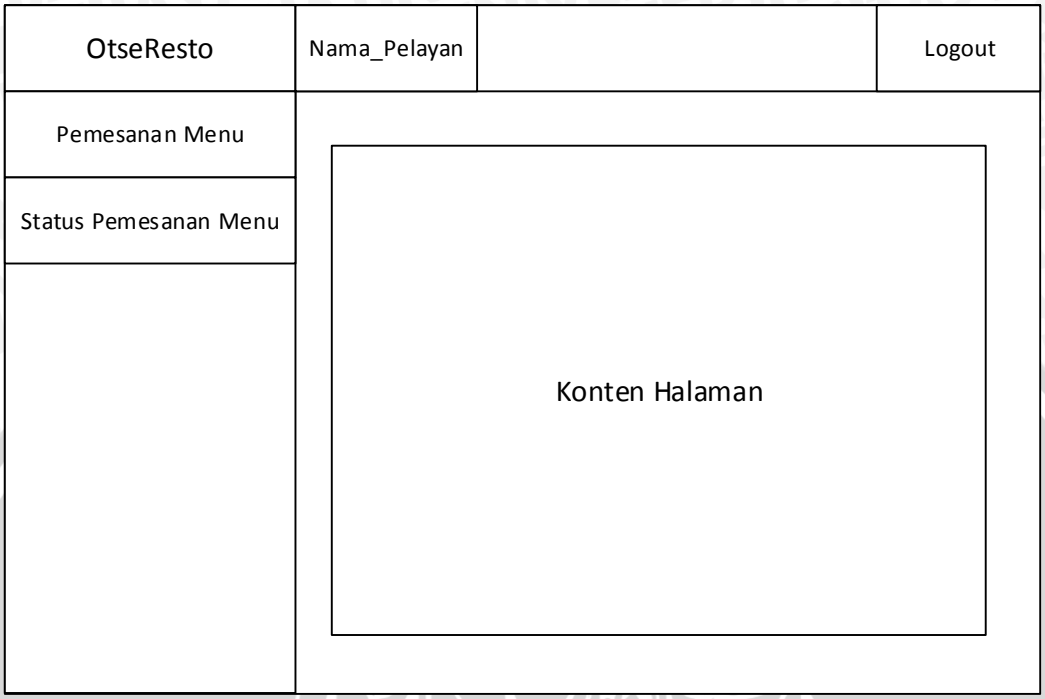
1. Perancangan Antarmuka *Login*
   Gambar 4.53 merepresentasikan rancangan antarmuka untuk *login* ke sistem.



Gambar 4.53 Perancangan Antarmuka *Login*

16. Perancangan Antarmuka Pelayan

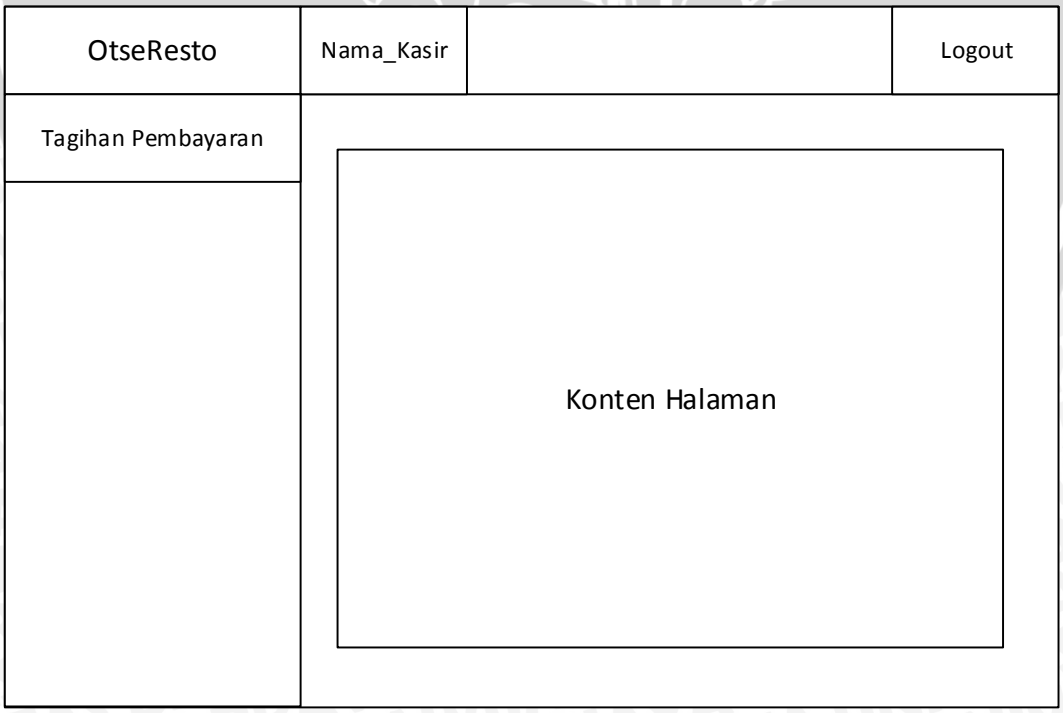
Gambar 4.54 merepresentasikan rancangan antarmuka untuk aktor pelayan.



Gambar 4.54 Perancangan Antarmuka Pelayan

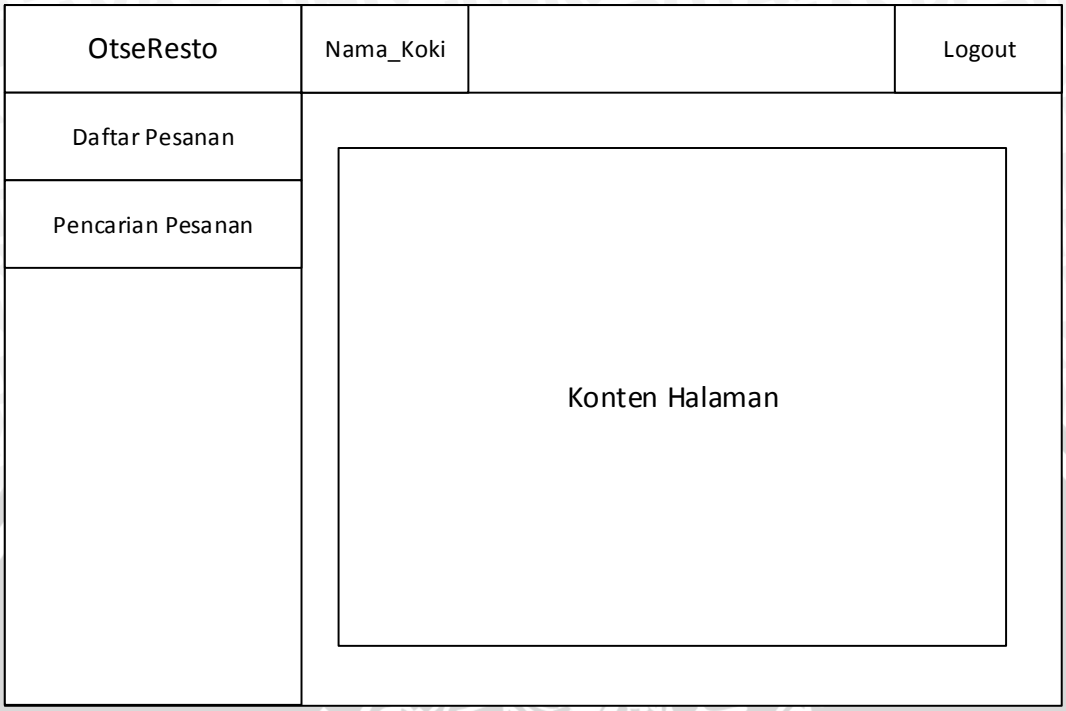
2. Perancangan Antarmuka Kasir

Gambar 4.55 merepresentasikan rancangan antarmuka untuk aktor kasir.



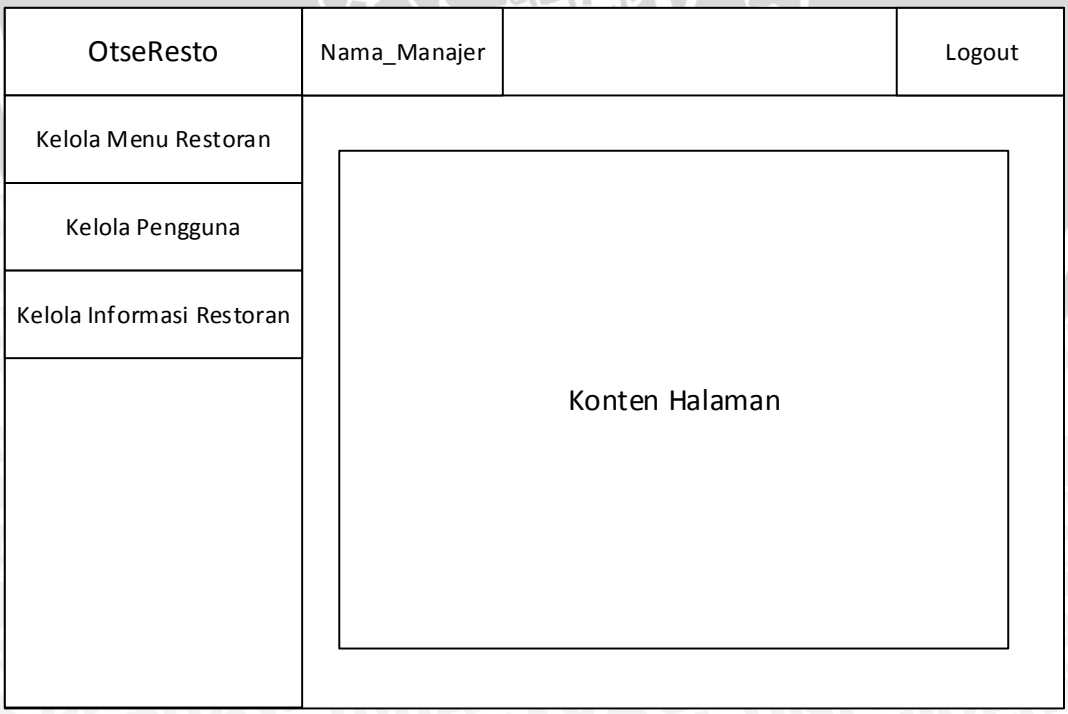
Gambar 4.55 Perancangan Antarmuka Kasir

3. Perancangan Antarmuka Koki
   Gambar 4.56 merepresentasikan rancangan antarmuka dari aktor koki.



Gambar 4.56 Perancangan Antarmuka Koki

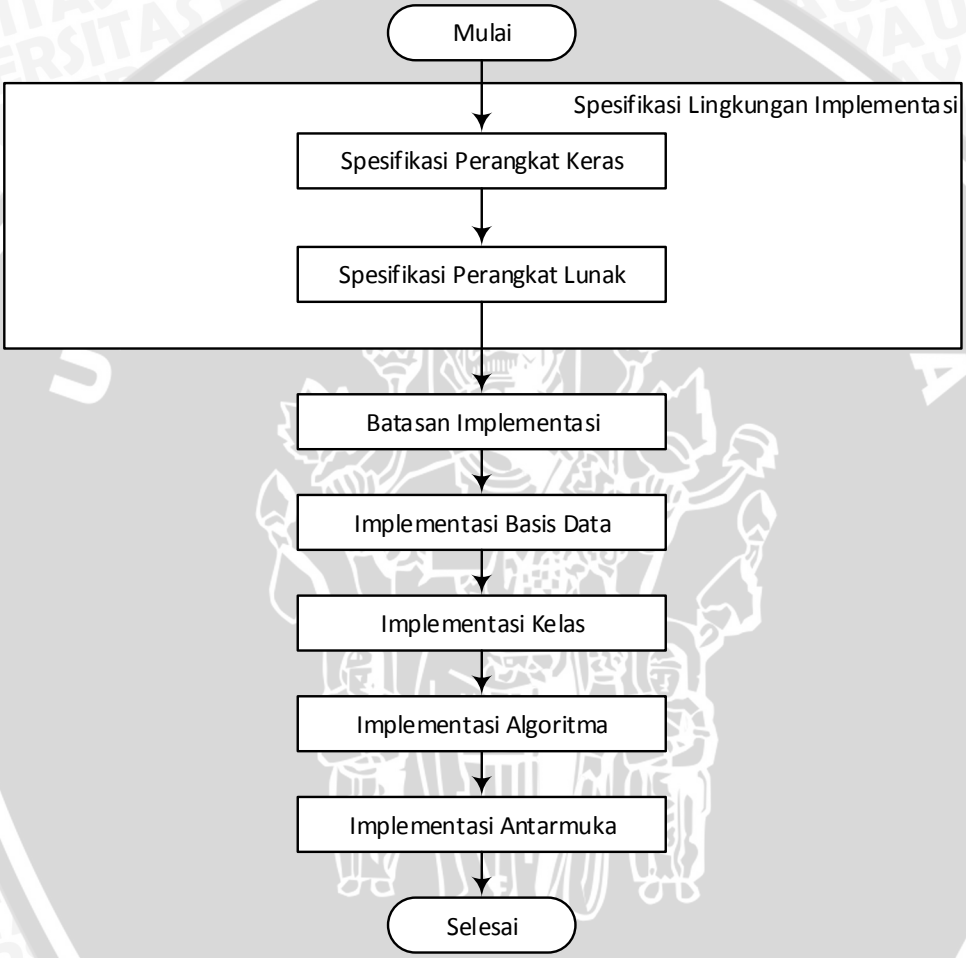
4. Perancangan Antarmuka Manajer
   Gambar 4.57 merepresentasikan rancangan antarmuka untuk aktor manajer.



Gambar 4.57 Perancangan Antarmuka Manajer

4.4 Implementasi

Bagian ini memaparkan tahapan implementasi sistem pengelolaan restoran terintegrasi (OtseResto) yang mengacu kepada tahap perancangan sistem. Bahasan pada sub bab ini terdiri dari penjelasan terkait spesifikasi lingkungan implementasi, batasan dalam implementasi, implementasi basis data, implementasi kelas, implementasi algoritma dan implementasi antarmuka. Gambar 4.58 menjelaskan alur dari implementasi sistem ini.



Gambar 4.58 Diagram Alir Implementasi

4.4.1 Spesifikasi Lingkungan Implementasi

Sistem pengelolaan restoran terintegrasi (OtseResto) ini dikembangkan dalam lingkungan implementasi yang terdiri dari perangkat keras dan perangkat lunak dengan bahasa pemrograman tertentu.

4.4.1.1 *Spesifikasi Perangkat Keras*

Spesifikasi perangkat keras yang digunakan dalam proses implementasi OtseResto dapat dilihat pada Tabel 4.19.

Tabel 4.19 Spesifikasi Perangkat Keras

Notebook Asus X450JV	
Processor	Intel Quad Core i7-4700HQ 2,4 GHz
Memory (RAM)	4 GB
Harddisk	1 TB
VGA	Intel HD Graphics 4600 Nvidia GeForce 745M
Resolusi Layar	1366x768 Pixel

4.4.1.2 *Spesifikasi Perangkat Lunak*

Spesifikasi perangkat lunak yang digunakan dalam proses implementasi OtseResto dapat dilihat pada Tabel 4.20.

Tabel 4.20 Spesifikasi Perangkat Lunak

Notebook Asus X450JV	
Sistem Operasi	Microsoft Windows 8.1 Pro with Media Center x64
Bahasa Pemrograman	PHP, HTML, CSS, Javascript
Perkakas Bantu	XAMPP 1.8.3 Sublime Text 3 Adobe Illustrator CS6 Enterprise Architect 11 Microsoft Visio 2013 Microsoft Word 2013

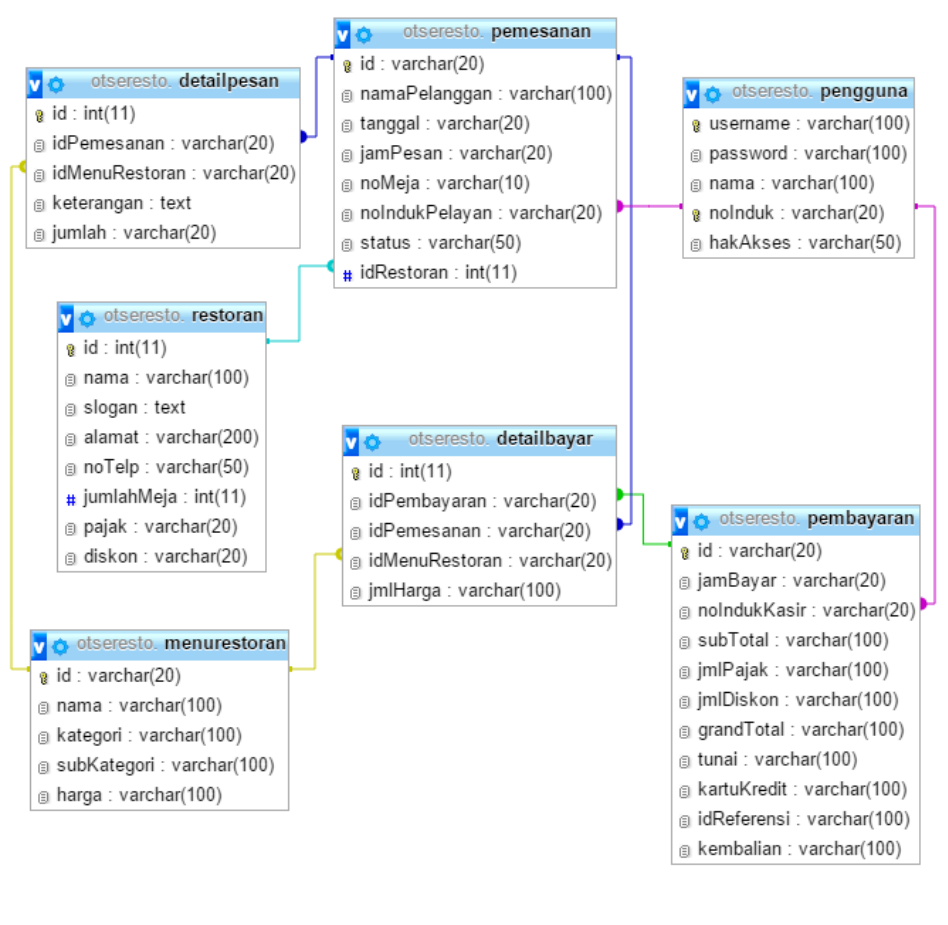
4.4.2 Batasan Implementasi

Beberapa batasan dalam implementasi OtseResto diantaranya:

1. Sistem yang dibangun mencakup empat bagian yang ada di restoran yaitu pelayan, kasir, koki, dan manajer.
2. *Database Management System* yang digunakan adalah MySQL versi 5.6.20.
3. *Framework* CodeIgniter yang digunakan adalah versi 2.1.0.
4. Simulasi pengoperasian sistem menggunakan koneksi Ad-Hoc.

4.4.3 Implementasi Basis Data

Implementasi basis data dilakukan berdasarkan perancangan sistem dengan menggunakan perkakas bantu PHPMyAdmin 4.2.7.1. Skema basis data yang telah diimplementasikan dapat dilihat pada Gambar 4.59.



Gambar 4.59 Skema Basis Data OtseResto

4.4.4 Implementasi Kelas

Setiap kelas yang telah dirancang pada tahap perancangan sistem direalisasikan pada sebuah *file* sistem dengan ekstensi *.php. Penjelasan terkait kelas dan *file* sistem yang digunakan pada sistem akan dijelaskan pada Tabel 4.21.

Tabel 4.21 Implementasi Kelas

No	*Package*	Nama Kelas	Nama *File* Sistem
1	*controllers*	Autentikasi	autentikasi.php
2	*controllers*	KelolaDapur	pengelolaanDapur.php
3	*controllers*	KelolaInformasiRestoran	pengelolaanInformasiRestoran.php
4	*controllers*	KelolaMenuRestoran	pengelolaanMenuRestoran.php
5	*controllers*	KelolaPembayaran	pengelolaanPembayaran.php
6	*controllers*	KelolaPemesanan	pengelolaanPemesanan.php
7	*controllers*	KelolaPengguna	pengelolaanPengguna.php
8	*models*	Dapur	dapur.php
9	*models*	DetailBayar	detailBayar.php
10	*models*	DetailPesan	detailPesan.php

11	*models*	MenuRestoran	menuRestoran.php
12	*models*	ModAutentikasi	modAutentikasi.php
13	*models*	Pembayaran	pembayaran.php
14	*models*	Pemesanan	pemesanan.php
15	*models*	Pengguna	pengguna.php
16	*models*	Restoran	restoran.php
17	*views*	*File* HTML	login.php
18	*views*/dapur	*File* HTML	pencarianPesanan.php
19	*views*/dapur	*File* HTML	homeDapur.php
20	*views*/dapur	*File* HTML	penglihatanDaftarPesanan.php
21	*views*/dapur	*File* HTML	strukLembarPesanan.php
22	*views*/manajerial	*File* HTML	pembuatanMenuRestoran.php
23	*views*/manajerial	*File* HTML	pembuatanPengguna.php
24	*views*/manajerial	*File* HTML	homeManajerial.php
25	*views*/manajerial	*File* HTML	penglihatanHistoriTransaksi.php
26	*views*/manajerial	*File* HTML	penglihatanMenuRestoran.php
27	*views*/manajerial	*File* HTML	penglihatanPengguna.php
28	*views*/manajerial	*File* HTML	pengubahanInformasiRestoran.php
29	*views*/manajerial	*File* HTML	pengubahanMenuRestoran.php
30	*views*/manajerial	*File* HTML	pengubahanPengguna.php
31	*views*/posPembayaran	*File* HTML	pengubahanTagihanPembayaran.php
32	*views*/posPembayaran	*File* HTML	homePOSPembayaran.php
33	*views*/posPembayaran	*File* HTML	penglihatanTagihanPembayaran.php
34	*views*/posPembayaran	*File* HTML	strukTagihanPembayaran.php
35	*views*/posPemesanan	*File* HTML	pembuatanPemesananMenu.php
36	*views*/posPemesanan	*File* HTML	homePOSPemesanan.php
37	*views*/posPemesanan	*File* HTML	penglihatanStatusPemesananMenu.php

4.4.5 Implementasi Algoritma

Implementasi algoritma yang dicantumkan dalam laporan skripsi ini merupakan lima kebutuhan fungsional yang memiliki prioritas tinggi. Di mana kelima algoritma tersebut memiliki nilai kompleksitas lebih dari sama dengan lima.

4.4.5.1 *Algoritma Membuat Pemesanan Menu*

Algoritma membuat pemesanan menu ini merupakan algoritma untuk operasi modMembuatPemesananMenu dari kelas pemesanan pada *package models*. Algoritma ini berfungsi untuk melakukan pemrosesan *input* pemesanan menu makanan dan minuman pada restoran yang dioperasikan oleh pelayan. Gambar 4.60 menjelaskan algoritma membuat pemesanan menu.

```
1  procedure membuatPemesananMenu;
2        SET waktu lokal;
3        SET namaPelanggan, tanggal, jamPesan, noMeja, noIndukPelayan,
   status, idRestoran, cekID, qty, jumlahPesanan;
4        IF noMeja != NULL
5              THEN cekID = DB SELECT pemesanan;
6                    IF cekID.nilai >= 1
7                          THEN buat ID Pesan untuk tamu restoran
   yang ingin pesan kembali;
8                          ELSE buat ID Pesan untuk tamu restoran
   yang baru pertama pesan;
9                    ENDIF
10                   FOR i = 1 TO jumlahPesanan;
11                         IF IDMenuRestoran.i == NULL || qty.i <= 0
12                               THEN return FALSE;
13                         ENDIF
14                         FOR j = jumlahPesanan to i;
15                               IF       IDMenuRestoran.j       ==
   IDMenuRestoran.i
16                                     THEN return FALSE;
17                               ENDIF
18                         ENDFOR
19                   ENDFOR
20                   DB INSERT pemesanan;
21                   return TRUE;
22             ELSE return FALSE;
23       ENDIF
24 END membuatPemesananMenu
```

Gambar 4.60 Algoritma Membuat Pemesanan Menu

4.4.5.2 *Algoritma Melihat Status Pemesanan Menu*

Algoritma melihat status pemesanan menu ini merupakan algoritma untuk operasi melihatStatusPemesananMenu dari kelas kelolaPemesanan pada *package controllers*. Algoritma ini berfungsi untuk melakukan pemantauan status secara *real-time* terhadap menu makanan dan minuman yang telah disiapkan oleh koki untuk segera diantarkan pelayan. Gambar 4.61 menjelaskan algoritma melihat status pemesanan menu.

```
1  procedure melihatStatusPemesananMenu;
2        IF statusAutentikasi == Pelayan
3              THEN SET id, refresh;
4                    IF refresh IS SET
5                          THEN load statusPemesananMenu;
6                               SET result = statusPemesananMenu;
7                               SET array json_response;
8                               IF result != NULL
9                                     THEN FOREACH
10                                                pindahkan
   variabel result ke dalam array json_response;
11                                      ENDFOREACH
```

repository.ub.ac.id

```
                                        print atribut file JSON
                                    ELSE json_response == NULL;
                                ENDIF
                            ELSEIF id IS SET
                            THEN ubah statusPemesananMenu;
                            ELSEIF badge IS SET
                            THEN badge = cek record statusPemesananMenu terbaru;
                                IF badge.nilai > 0
                                    THEN print badge;
                                ENDIF
                            ELSE load halaman statusPemesananMenu;
                        ENDIF
                    ELSE keluar sistem;
            ENDIF
END melihatStatusPemesananMenu
```

Gambar 4.61 Algoritma Melihat Status Pemesanan Menu

4.4.5.3 *Algoritma Membuat Tagihan Pembayaran*

Algoritma membuat tagihan pembayaran ini merupakan algoritma untuk operasi modMembuaTagihanPembayaran dari kelas pembayaran pada *package models*. Algoritma ini berfungsi untuk melakukan melakukan pemrosesan *input* dari tagihan pembayaran makanan dan minuman pada restoran yang dioperasikan oleh kasir. Gambar 4.62 menjelaskan algoritma membuat tagihan pembayaran.

```
procedure membuatTagihanPembayaran;
        SET waktu lokal;
        SET jamBayar, noIndukKasir, subTotal, jmlPajak, jmlDiskon, grandTotal;
        IF noIndukKasir != NULL && subTotal != NULL
                THEN IF metodePembayaran == TUNAI && jmlTunai != NULL && kembalian != NULL
                                THEN lakukan metode pembayaran TUNAI;
                                ELSEIF metodePembayaran == KARTUKREDIT && idReferensi != NULL
                                THEN lakukan metodePembayaran KARTUKREDIT;
                                ELSE return FALSE;
                        ENDIF
                        SET IDPembayaran;
                        DB INSERT pembayaran;
                        DB UPDATE pemesanan;
                        return TRUE;
                ELSE return FALSE;
        ENDIF
END membuatTagihanPembayaran
```

Gambar 4.62 Algoritma Membuat Tagihan Pembayaran

UNIVERSITAS BRAWIJAYA

4.4.5.4 *Algoritma Notifikasi Pesanan*

Algoritma memberikan notifikasi pesanan ini merupakan algoritma untuk operasi memberikanNotifikasiPesanan dari kelas kelolaDapur pada *package controllers*. Algoritma ini berfungsi untuk memberikan notifikasi peringatan pada pesanan yang tidak disiapkan dalam batas waktu yang telah ditentukan kepada koki secara *real-time*. Gambar 4.63 menjelaskan algoritma notifikasi pesanan.

```
1  procedure memberikanNotifikasiPesanan;
2        IF statusAutentikasi == Koki
3              THEN SET waktu lokal;
4                    SET waktuPesan, sekarang, diff;
5                    diff = waktuPesan - sekarang;
6                    IF diff > 10 && diff <= 15
7                          THEN notifikasi = warna kuning;
8                          ELSEIF diff > 15 && diff <= 20
9                          THEN notifikasi = warna orange;
10                         ELSEIF diff > 20
11                         THEN notifikasi = warna merah;
12                         ELSE notifikasi = NULL;
13                   ENDIF
14                   return notifikasi;
15             ELSE redirect login;
16       ENDIF
17 END memberikanNotifikasiPesanan
```

Gambar 4.63 Algoritma Notifikasi Pesanan

4.4.5.5 *Algoritma Melihat Daftar Pesanan*

Algoritma melihat daftar pesanan ini merupakan algoritma untuk operasi melihatDaftarPesanan dari kelas kelolaDapur pada *package controllers*. Algoritma ini berfungsi untuk melihat daftar pesanan untuk segera disiapkan oleh koki secara *real-time*. Gambar 4.64 menjelaskan algoritma melihat daftar pesanan.

```
1  procedure melihatDaftarPesanan;
2        IF statusAutentikasi == Koki
3              THEN SET badgeStat;
4                    IF refresh IS SET
5                          THEN load daftarPesanan;
6                                SET result = daftarPesanan;
7                                SET array json_response;
8                                IF result != NULL
9                                      THEN FOREACH
10                                                pindahkan
   variabel result ke dalam array json_response;
11                                     ENDFOREACH
12                                     print   atribut   file
   JSON
13                               ELSE json_response == NULL;
14                               ENDIF
15                         ELSEIF badge IS SET
```

```
16                              THEN      badge      =      cek      record
statusPemesananMenu terbaru;
17                                  IF badge.nilai > 0
18                                        THEN print badge;
19                                  ENDIF
20                              ELSE load halaman daftarPesanan;
21                        ENDIF
22                ELSE redirect login;
23          ENDIF
24  END melihatDaftarPesanan
```

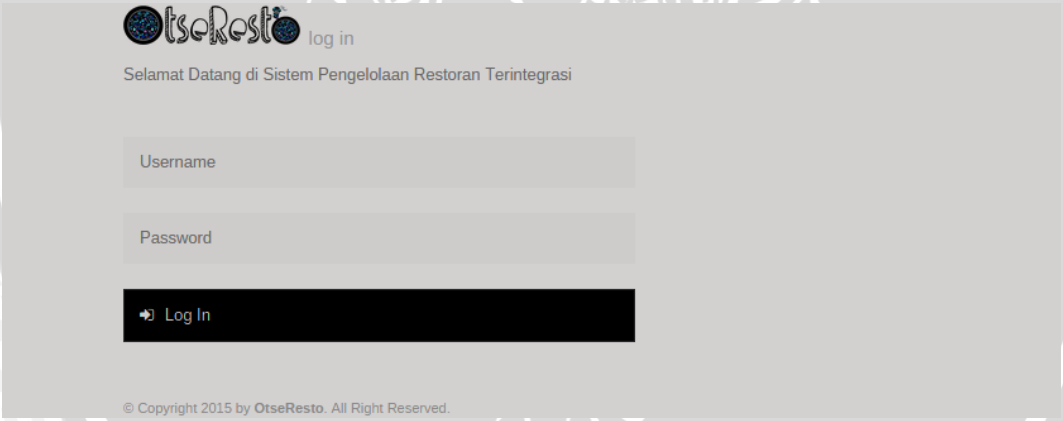
Gambar 4.64 Algoritma Melihat Daftar Pesanan

4.4.6 Implementasi Antarmuka

Antarmuka OtseResto digunakan oleh pengguna agar dapat berinteraksi dengan komponen-komponen sistem yang telah dikembangkan. Antarmuka utama dibagi menjadi empat bagian yaitu antarmuka untuk pelayan, kasir, koki, dan manajer.

4.4.6.1 *Implementasi Antarmuka Login*

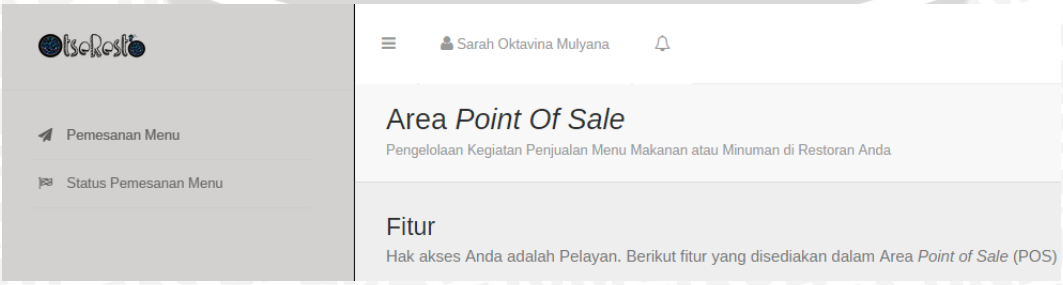
Gambar 4.65 menggambarkan hasil implementasi dari halaman login. Halaman ini digunakan untuk melakukan autentikasi terhadap pengguna yang ingin mengakses fitur-fitur OtseResto.



Gambar 4.65 Implementasi Antarmuka *Login*

4.4.6.2 *Implementasi Antarmuka Pelayan*

Gambar 4.66 menggambarkan hasil implementasi dari antarmuka pelayan.



Gambar 4.66 Implementasi Antarmuka Pelayan

Antarmuka yang digambarkan tersebut merupakan halaman utama atau *dashboard* dari pelayan. Di mana terdapat dua pilihan utama yaitu:

1. Pemesanan Menu

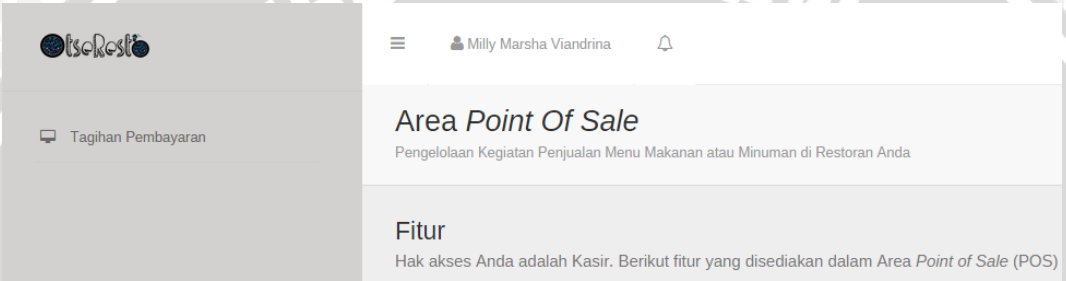
   Digunakan untuk melakukan pemesanan menu makanan atau minuman yang disediakan restoran.

2. Status Pemesanan Menu

   Digunakan untuk melihat status dari menu makanan atau minuman yang telah dipesan.

4.4.6.3 *Implementasi Antarmuka Kasir*

Gambar 4.67 menggambarkan hasil implementasi dari antarmuka kasir.

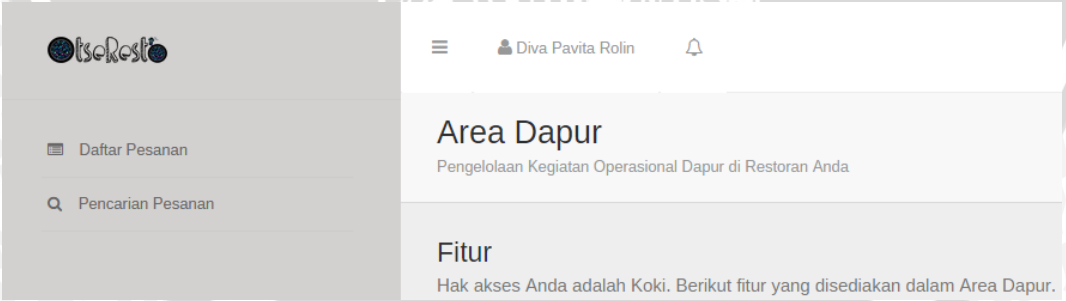


Gambar 4.67 Implementasi Antarmuka Kasir

Antarmuka yang digambarkan tersebut merupakan halaman utama atau *dashboard* dari kasir. Di mana terdapat satu pilihan utama yaitu Tagihan Pembayaran yang disediakan untuk melakukan fungsi pemrosesan tagihan makanan dan minuman yang telah dikonsumsi tamu restoran.

4.4.6.4 *Implementasi Antarmuka Koki*

Gambar 4.68 menggambarkan hasil implementasi dari antarmuka koki.



Gambar 4.68 Implementasi Antarmuka Koki

Antarmuka yang digambarkan tersebut merupakan halaman utama atau *dashboard* dari koki. Di mana terdapat dua pilihan utama yaitu:

1. Daftar Pesanan

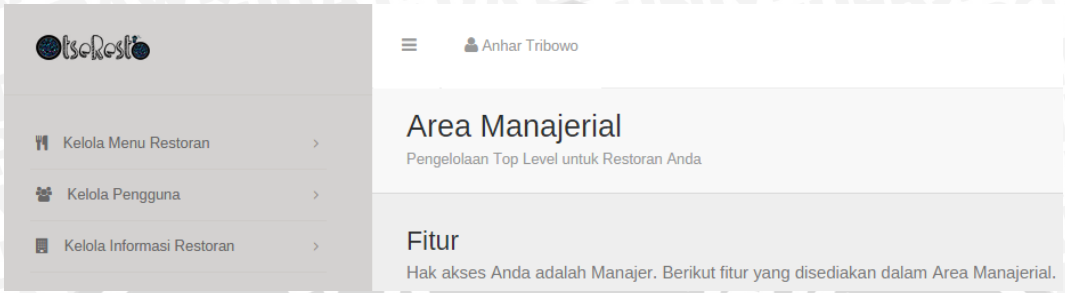
   Digunakan untuk memantau dan melihat daftar pesanan yang harus disiapkan oleh koki.

2. Pencarian Pesanan

Digunakan untuk mencari lembar pesanan jika lembar pesanan yang sebelumnya dicetak telah hilang.

4.4.6.5 *Implementasi Antarmuka Manajer*

Gambar 4.69 menggambarkan hasil implementasi dari antarmuka manajer.



Gambar 4.69 Implementasi Antarmuka Manajer

Antarmuka yang digambarkan tersebut merupakan halaman utama atau *dashboard* dari manajer. Di mana terdapat tiga pilihan utama yaitu:

1. Kelola Menu Restoran

   Digunakan untuk melakukan pengelolaan terhadap menu makanan dan minuman yang disediakan oleh restoran.

2. Kelola Pengguna

   Digunakan untuk melakukan pengelolaan terhadap pengguna yang dapat mengakses sistem sesuai dengan hak aksesnya.

3. Kelola Informasi Restoran

   Digunakan untuk melakukan pengelolaan informasi restoran. Informasi restoran yang dimaksud berupa penentuan nama, slogan, alamat, besaran pajak, dan lainnya. Di samping itu, pada pilihan ini terdapat fitur untuk melihat histori transaksi yang dilakukan oleh kasir.

BAB V
PENGUJIAN DAN ANALISIS

Pada bab ini akan dilakukan tahap pengujian dan analisis dari sistem pengelolaan restoran terintegrasi. Pengujian yang dilakukan melalui dua tahap besar yaitu pengujian dengan paradigma rekayasa perangkat lunak yang kemudian dilanjutkan dengan pengujian berbasis model kualitas ISO 9126. Pengujian dengan paradigma rekayasa perangkat lunak dilakukan dengan pengujian unit, pengujian integrasi dan pengujian validasi. Sedangkan pengujian dengan model kualitas ISO 9126 hanya dilakukan pada karakteristik utama yaitu *functionality, reliability, usability, efficiency, maintainability,* dan *portability*. Seluruh pengujian disertakan analisis dari hasil pengujian tersebut dengan mengacu kepada dasar teori yang bertujuan untuk memastikan sistem telah memenuhi kebutuhan yang didefinisikan pada tahap analisis kebutuhan.

5.1 Pengujian Unit

5.1.1 Tujuan

Tujuan dilakukannya pengujian unit adalah untuk memastikan beberapa algoritma yang memiliki prioritas tinggi telah diimplementasikan sesuai dengan yang diharapkan.

5.1.2 Mekanisme

Pengujian unit dilakukan dengan menggunakan teknik *white box* dengan jenis pengujian *basis path*. Algoritma yang diujikan pada pengujian ini merupakan algoritma melihatStatusPemesananMenu, memberikanNotifikasiPesanan dan melihatDaftarPesanan. Pengujian unit dilakukan pada ketiga algoritma tersebut sebab merupakan domain permasalahan dari sistem yang telah dibangun.

5.1.3 Hasil Pengujian

1. Algoritma melihatStatusPemesananMenu.

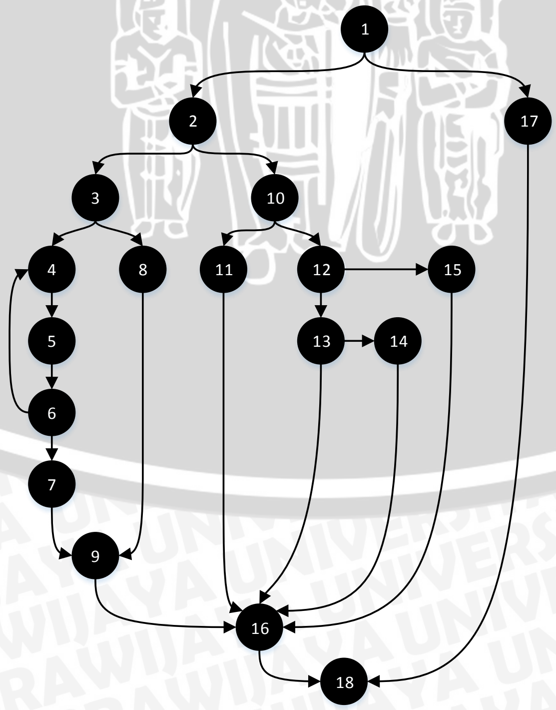
Gambar 5.1 memaparkan algoritma melihatStatusPemesananMenu beserta *node flowgraph.*

```
1  procedure melihatStatusPemesananMenu;
2          IF statusAutentikasi == Pelayan
3              THEN SET id, refresh;
4                   IF refresh IS SET
5                       THEN load statusPemesananMenu;
6                            SET result = statusPemesananMenu;
7                            SET array json_response;
8                            IF result != NULL
9                                 THEN FOREACH
```

```
                    5  pindahkan
variabel result ke dalam array json_response;
                  6  ENDFOREACH
                7  print atribut file
JSON
              8  ELSE json_response == NULL;
          9  ENDIF
          ELSEIF id IS SET  10
          THEN ubah statusPemesananMenu;  11
          ELSEIF badge IS SET  12
          THEN badge = cek record
statusPemesananMenu terbaru;
                                      13
              IF badge.nilai > 0
                  THEN print badge;  14
          15    ENDIF
                ELSE load halaman statusPemesananMenu;
      16  ENDIF
   17  ELSE keluar sistem;
18  ENDIF
END melihatStatusPemesananMenu
```

Gambar 5.1 Pembentukan *Node* Algoritma melihatStatusPemesananMenu

Berdasarkan Gambar 5.1 maka dapat dibentuk *flowgraph* yang terdiri dari 18 *node*. Gambar 5.2 memperlihatkan *flowgraph* dari algoritma melihatStatusPemesananMenu.



Gambar 5.2 *Flowgraph* Algoritma melihatStatusPemesananMenu

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Berdasarkan *flowgraph* algoritma melihatStatusPemesananMenu yang dapat dilihat pada Gambar 5.2, maka dapat dihitung nilai *cyclomatic complexity*.

V(G) = E – N + 2

= 24 – 18 + 2

= 8

V(G) = P + 1

= 7 + 1

= 8

Sehingga berdasarkan hasil persamaan *cyclomatic complexity* maka didapatkan 8 jalur independen yaitu:

1. Jalur 1 = 1 – 17 – 18
2. Jalur 2 = 1 – 2 – 10 – 11 – 16 – 18
3. Jalur 3 = 1 – 2 – 10 – 12 – 15 – 16 – 18
4. Jalur 4 = 1 – 2 – 10 – 12 – 13 – 16 – 18
5. Jalur 5 = 1 – 2 – 10 – 12 – 13 – 14 – 16 – 18
6. Jalur 6 = 1 – 2 – 3 – 8 – 9 – 16 – 18
7. Jalur 7 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 9 – 16 – 18
8. Jalur 8 = 1 – 2 – 3 – 4 – 5 – 6 – 4 – 5 – 6 – 7 – 9 – 16 – 18

Berdasarkan 8 jalur independen yang telah didefinisikan tersebut maka dapat dibentuk kasus ujinya. Tabel 5.1 memaparkan kasus uji dari algoritma melihatStatusPemesananMenu.

Tabel 5.1 Kasus Uji Algoritma melihatStatusPemesananMenu

No	Jalur	Kelas	*Method*	Data *Input*	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
1	1 – 17 – 18	KelolaPemesanan	melihatStatusPemesananMenu()	ID Pemesanan, Refresh Status, Badge Status, ID Session	Sistem mengeluarkan pengguna jika ID Session NULL	Sistem mengeluarkan pengguna jika ID Session NULL	Valid
2	1 – 2 – 10 – 11 – 16 – 18	KelolaPemesanan	melihatStatusPemesananMenu()	ID Pemesanan, Refresh Status, Badge Status, ID Session	Sistem mengubah status pemesanan menu	Sistem mengubah status pemesanan menu	Valid
3	1 – 2 – 10 – 12 – 15 – 16 – 18	KelolaPemesanan	melihatStatusPemesananMenu()	ID Pemesanan, Refresh Status, Badge	Sistem hanya menampilkan halaman status	Sistem hanya menampilkan halaman status	Valid

				Status, ID Session	pemesanan menu	pemesanan menu	
4	1 – 2 – 10 – 12 – 13 – 16 – 18	KelolaPemesanan	melihatStatusPemesanMenu()	ID Pemesanan, Refresh Status, Badge Status, ID Session	Sistem memperbaharui notifikasi status pemesanan menu	Sistem memperbaharui notifikasi status pemesanan menu	Valid
5	1 – 2 – 10 – 12 – 13 – 14 – 16 – 18	KelolaPemesanan	melihatStatusPemesanMenu()	ID Pemesanan, Refresh Status, Badge Status, ID Session	Sistem memperbaharui dan memberikan notifikasi status pemesanan menu	Sistem memperbaharui dan memberikan notifikasi status pemesanan menu	Valid
6	1 – 2 – 3 – 8 – 9 – 16 – 18	KelolaPemesanan	melihatStatusPemesanMenu()	ID Pemesanan, Refresh Status, Badge Status, ID Session	Sistem memperbaharui tabel status pemesanan menu namun dengan nilai JSON kosong	Sistem memperbaharui tabel status pemesanan menu namun dengan nilai JSON kosong	Valid
7	1 – 2 – 3 – 4 – 5 – 6 – 7 – 9 – 16 – 18	KelolaPemesanan	melihatStatusPemesanMenu()	ID Pemesanan, Refresh Status, Badge Status, ID Session	Sistem memperbaharui tabel status pemesanan menu namun hanya dengan satu *record* JSON	Sistem memperbaharui tabel status pemesanan menu namun dengan hanya satu *record* JSON	Valid
8	1 – 2 – 3 – 4 – 5 – 6 – 4 – 5 – 6 – 7 – 9 – 16 – 18	KelolaPemesanan	melihatStatusPemesanMenu()	ID Pemesanan, Refresh Status, Badge Status, ID Session	Sistem memperbaharui tabel status pemesanan menu dengan *record* JSON lebih dari satu	Sistem memperbaharui tabel status pemesanan menu dengan *record* JSON lebih dari satu	Valid

2. Algoritma memberikanNotifikasiPesanan.

Gambar 5.3 menggambarkan algoritma memberikanNotifikasiPesanan beserta *node flowgraph*.

```
1  procedure memberikanNotifikasiPesanan;
2        IF statusAutentikasi == Koki
3              THEN SET waktu lokal;
4                   SET waktuPesan, sekarang, diff;
5                   diff = waktuPesan - sekarang;
```

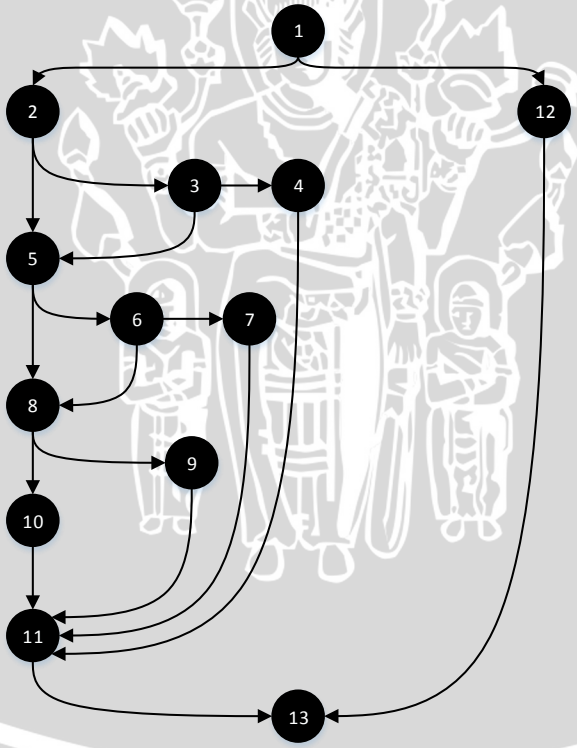
1

2

```
6                    IF diff > 10 && diff <= 15
7                        THEN notifikasi = warna kuning;
8                    ELSEIF diff > 15 && diff <= 20
9                        THEN notifikasi = warna orange;
10                   ELSEIF diff > 20
11                       THEN notifikasi = warna merah;
12                   ELSE notifikasi = NULL;
13               ENDIF
14               return notifikasi;
15           ELSE redirect login;
16       ENDIF
17   END memberikanNotifikasiPesanan
```

Gambar 5.3 Pembentukan *Node* Algoritma memberikanNotifikasiPesanan

Berdasarkan Gambar 5.3 maka dapat dibentuk *flowgraph* yang terdiri dari 13 *node*. Gambar 5.4 memperlihatkan *flowgraph* dari algoritma memberikanNotifikasiPesanan.



Gambar 5.4 *Flowgraph* Algoritma memberikanNotifikasiPesanan

Berdasarkan *flowgraph* algoritma memberikanNotifikasiPesanan yang dapat dilihat pada Gambar 5.4, maka dapat dihitung nilai *cyclomatic complexity*.

V(G) = E – N + 2

= 18 – 13 + 2

= 7

V(G) = P + 1

= 6 + 1

= 7

Sehingga berdasarkan hasil persamaan *cyclomatic complexity* maka didapatkan 7 jalur independen yaitu:

1. Jalur 1 = 1 – 12 – 13
2. Jalur 2 = 1 – 2 – 5 – 8 – 10 – 11 – 13
3. Jalur 3 = 1 – 2 – 3 – 5 – 8 – 10 – 11 – 13
4. Jalur 4 = 1 – 2 – 3 – 4 – 11 – 13
5. Jalur 5 = 1 – 2 – 5 – 6 – 8 – 10 – 11 – 13
6. Jalur 6 = 1 – 2 – 5 – 6 – 7 – 11 – 13
7. Jalur 7 = 1 – 2 – 5 – 8 – 9 – 11 – 13

Berdasarkan 7 jalur independen yang telah didefinisikan tersebut maka dapat dibentuk kasus ujinya. Tabel 5.2 memaparkan kasus uji dari algoritma memberikanNotifikasiPesanan.

Tabel 5.2 Kasus Uji Algoritma memberikanNotifikasiPesanan

No	Jalur	Kelas	*Method*	Data *Input*	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
1	1 – 12 – 13	KelolaDapur	memberikanNotifikasiPesanan( )	ID Pemesanan, Jam Pesan, ID Session	Sistem mengeluarkan pengguna jika ID Session NULL	Sistem mengeluarkan pengguna jika ID Session NULL	Valid
2	1 – 2 – 5 – 8 – 10 – 11 – 13	KelolaDapur	memberikanNotifikasiPesanan( )	ID Pemesanan, Jam Pesan, ID Session	Sistem memperbaharui halaman namun tanpa notifikasi	Sistem memperbaharui halaman namun tanpa notifikasi	Valid
3	1 – 2 – 3 – 5 – 8 – 10 – 11 – 13	KelolaDapur	memberikanNotifikasiPesanan( )	ID Pemesanan, Jam Pesan, ID Session	Sistem memperbaharui halaman namun tanpa notifikasi	Sistem memperbaharui halaman namun tanpa notifikasi	Valid
4	1 – 2 – 3 – 4 – 11 – 13	KelolaDapur	memberikanNotifikasiPesanan( )	ID Pemesanan, Jam Pesan, ID Session	Sistem memperbaharui halaman dengan notifikasi kuning	Sistem memperbaharui halaman dengan notifikasi kuning	Valid

repository.ub.ac.id

5	1 – 2 – 5 – 6 – 8 – 10 – 11 – 13	KelolaDapur	memberikanNotifikasiPesanan()	ID Pemesanan, Jam Pesan, ID Session	Sistem memperbaharui halaman namun tanpa notifikasi	Sistem memperbaharui halaman namun tanpa notifikasi	Valid
6	1 – 2 – 5 – 6 – 7 – 11 – 13	KelolaDapur	memberikanNotifikasiPesanan()	ID Pemesanan, Jam Pesan, ID Session	Sistem memperbaharui halaman dengan notifikasi oranye	Sistem memperbaharui halaman dengan notifikasi oranye	Valid
7	1 – 2 – 5 – 8 – 9 – 11 – 13	KelolaDapur	memberikanNotifikasiPesanan()	ID Pemesanan, Jam Pesan, ID Session	Sistem memperbaharui halaman dengan notifikasi merah	Sistem memperbaharui halaman dengan notifikasi merah	Valid

3. Algoritma melihatDaftarPesanan

Gambar 5.5 menggambarkan algoritma melihatDaftarPesanan beserta *node flowgraph*.

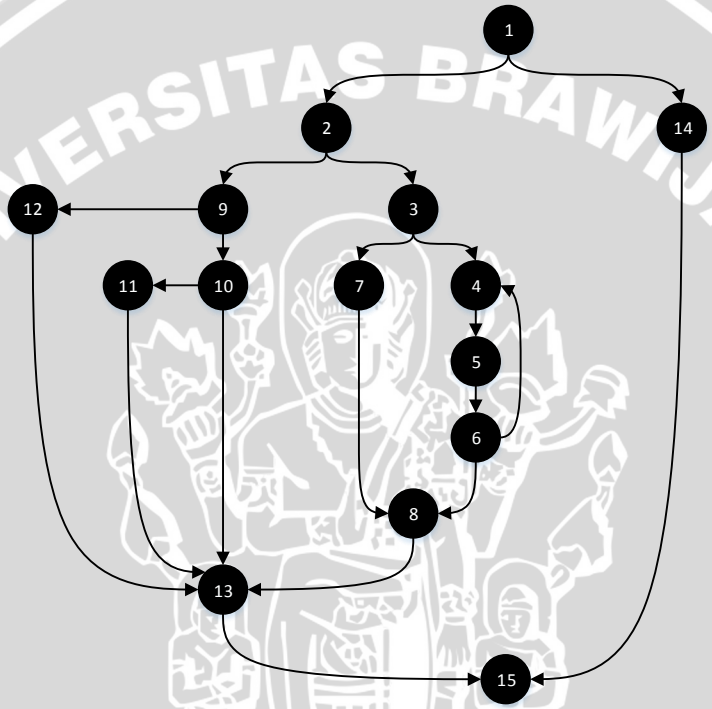
```
procedure melihatDaftarPesanan;
        IF statusAutentikasi == Koki
                THEN SET badgeStat;
                        IF refresh IS SET
                                THEN load daftarPesanan;
                                        SET result = daftarPesanan;
                                        SET array json_response;
                                        IF result != NULL
                                                THEN FOREACH
                                                        pindahkan variabel result ke dalam array json_response;
                                                        ENDFOREACH
                                                        print atribut file JSON
                                                ELSE json_response == NULL;
                                        ENDIF
                                ELSEIF badge IS SET
                                THEN badge = cek record statusPemesananMenu terbaru;
                                        IF badge.nilai > 0
                                                THEN print badge
                                        ENDIF
```


UNIVERSITAS BRAWIJAYA

```
20                          ELSE load halaman daftarPesanan;
21                      ENDIF
22                  ELSE redirect login;
23              ENDIF
24  END melihatDaftarPesanan
```

Gambar 5.5 Pembentukan *Node* Algoritma melihatDaftarPesanan

Berdasarkan Gambar 5.5 maka dapat dibentuk *flowgraph* yang terdiri dari 15 *node*. Gambar 5.6 memperlihatkan *flowgraph* dari algoritma melihatDaftarPesanan.



Gambar 5.6 *Flowgraph* Algoritma melihatDaftarPesanan

Berdasarkan *flowgraph* algoritma melihatDaftarPesanan yang dapat dilihat pada Gambar 5.6, maka dapat dihitung nilai *cyclomatic complexity*.

V(G) = E – N + 2

= 20 – 15 + 2

= 7

V(G) = P + 1

= 6 + 1

= 7

Sehingga berdasarkan hasil persamaan *cyclomatic complexity* maka didapatkan 7 jalur independen yaitu:

1. Jalur 1 = 1 – 14 – 15

2. Jalur 2 = 1 – 2 – 12 – 13 – 15
3. Jalur 3 = 1 – 2 – 9 – 10 – 13 – 15
4. Jalur 4 = 1 – 2 – 9 – 10 – 11 – 13 – 15
5. Jalur 5 = 1 – 2 – 3 – 7 – 8 – 13 – 15
6. Jalur 6 = 1 – 2 – 3 – 4 – 5 – 6 – 8 – 13 – 15
7. Jalur 7 = 1 – 2 – 3 – 4 – 5 – 6 – 4 – 5 – 6 – 8 – 13 – 15

Berdasarkan 7 jalur independen yang telah didefinisikan tersebut maka dapat dibentuk kasus ujinya. Tabel 5.3 memaparkan kasus uji dari algoritma melihatDaftarPesanan.

Tabel 5.3 Kasus Uji Algoritma melihatDaftarPesanan

No	Jalur	Kelas	*Method*	Data *Input*	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
1	1 – 14 – 15	KelolaDapur	melihatDaftarPesanan()	Refresh Status, Badge Status, ID Session	Sistem mengeluarkan pengguna jika ID Session NULL	Sistem mengeluarkan pengguna jika ID Session NULL	Valid
2	1 – 2 – 12 – 13 – 15	KelolaDapur	melihatDaftarPesanan()	Refresh Status, Badge Status, ID Session	Sistem hanya menampilkan halaman daftar pesanan	Sistem hanya menampilkan halaman daftar pesanan	Valid
3	1 – 2 – 9 – 10 – 13 – 15	KelolaDapur	melihatDaftarPesanan()	Refresh Status, Badge Status, ID Session	Sistem memperbahar ui notifikasi daftar pesanan	Sistem memperbahar ui notifikasi daftar pesanan	Valid
4	1 – 2 – 9 – 10 – 11 – 13 – 15	KelolaDapur	melihatDaftarPesanan()	Refresh Status, Badge Status, ID Session	Sistem memperbahar ui dan memberikan notifikasi daftar pesanan	Sistem memperbahar ui dan memberikan notifikasi daftar pesanan	Valid
5	1 – 2 – 3 – 7 – 8 – 13 – 15	KelolaDapur	melihatDaftarPesanan()	Refresh Status, Badge Status, ID Session	Sistem memperbahar ui tabel daftar pesanan namun dengan nilai JSON kosong	Sistem memperbahar ui tabel daftar pesanan namun dengan nilai JSON kosong	Valid
6	1 – 2 – 3 – 4 – 5 – 6 – 8 – 13 – 15	KelolaDapur	melihatDaftarPesanan()	Refresh Status, Badge	Sistem memperbahar ui tabel daftar pesanan	Sistem memperbahar ui tabel daftar pesanan	Valid

				Status, ID Session	namun hanya dengan satu *record* JSON	namun hanya dengan satu *record* JSON	
7	1 – 2 – 3 – 4 – 5 – 6 – 4 – 5 – 6 – 8 – 13 – 15	KelolaDapur	melihatDaftarPesanan()	Refresh Status, Badge Status, ID Session	Sistem memperbaharui tabel daftar pesanan dengan *record* JSON lebih dari satu	Sistem memperbaharui tabel daftar pesanan dengan *record* JSON lebih dari satu	Valid

5.1.4 Analisis

Berdasarkan hasil pengujian yang telah dilakukan pada algoritma melihatStatusPemesananMenu, memberikanNotifikasiPesanan dan melihatDaftarPesanan bahwa seluruh kasus uji memiliki hasil yang telah sesuai dengan yang diharapkan atau bernilai valid.

5.2 Pengujian Integrasi

5.2.1 Tujuan

Tujuan dilakukannya pengujian integrasi adalah untuk memastikan proses yang melibatkan interaksi antar kelas telah terintegrasi sesuai dengan yang diharapkan.

5.2.2 Mekanisme

Pengujian integrasi dilakukan pada proses yang melibatkan interaksi antar kelas. Di mana teknik yang digunakan pada pengujian ini menggunakan *black box*. Sistem memenuhi aspek pengujian integrasi apabila seluruh hasil yang diharapkan (*expected result*) sesuai dengan hasil yang diperoleh (*actual result*). Namun jika terdapat hasil yang tidak sesuai yang diharapkan maka proses integrasi antar kelas harus diperbaiki. Berikut ini merupakan kasus uji dari pengujian integrasi.

Tabel 5.4 menjelaskan kasus uji integrasi antara unit PengelolaanPemesanan dengan Pemesanan.

Tabel 5.4 Kasus Uji Integrasi Unit PengelolaanPemesanan dan Pemesanan

Nomor Kasus Uji	INT_001
Nama Kasus Uji	Kasus uji integrasi unit PengelolaanPemesanan dan Pemesanan
Unit yang Terlibat	PengelolaanPemesanan dan Pemesanan
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa proses yang melibatkan interaksi antara unit PengelolaanPemesanan dan Pemesanan telah terintegrasi dengan baik

Prosedur Uji	1. Penguji menyiapkan kelas Tester untuk mengaktifkan unit PengelolaanPemesanan 2. Penguji menyiapkan *input-input* yang diperlukan unit PengelolaanPemesanan dan unit Pemesanan 3. Penguji mengeksekusi kelas Tester tersebut 4. Penguji melihat kembalian dari hasil eksekusi bernilai *true* atau *fase*
Hasil yang Diharapkan	Nilai kembalian bernilai *true* jika seluruh *input* valid dan nilai kembalian bernilai *false* jika terdapat *input* yang tidak valid

Tabel 5.5 menjelaskan kasus uji integrasi antara unit PengelolaanPemesanan dengan ModAutentikasi.

Tabel 5.5 Kasus Uji Integrasi Unit PengelolaanPemesanan dan ModAutentikasi

Nomor Kasus Uji	INT_002
Nama Kasus Uji	Kasus uji integrasi unit PengelolaanPemesanan dan ModAutentikasi
Unit yang Terlibat	PengelolaanPemesanan dan ModAutentikasi
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa proses yang melibatkan interaksi antara unit PengelolaanPemesanan dan ModAutentikasi telah terintegrasi dengan baik
Prosedur Uji	1. Penguji menyiapkan kelas Tester untuk mengaktifkan unit PengelolaanPemesanan 2. Penguji menyiapkan variabel *session* yang diperlukan unit PengelolaanPemesanan dan ModAutentikasi 3. Penguji mengeksekusi kelas Tester tersebut 4. Penguji melihat kembalian dari hasil eksekusi bernilai 'Pelayan' atau NULL
Hasil yang Diharapkan	Nilai kembalian bernilai 'Pelayan' jika variabel *session* valid dan nilai kembalian NULL jika variabel *session* tidak valid

Tabel 5.6 menjelaskan kasus uji integrasi antara unit PengelolaanPemesanan dengan MenuRestoran.

Tabel 5.6 Kasus Uji Integrasi Unit PengelolaanPemesanan dan MenuRestoran

Nomor Kasus Uji	INT_003
Nama Kasus Uji	Kasus uji integrasi unit PengelolaanPemesanan dan MenuRestoran
Unit yang Terlibat	PengelolaanPemesanan dan MenuRestoran

Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa proses yang melibatkan interaksi antara unit PengelolaanPemesanan dan MenuRestoran telah terintegrasi dengan baik
Prosedur Uji	1. Penguji menyiapkan kelas Tester untuk mengaktifkan unit PengelolaanPemesanan 2. Penguji menyiapkan *input-input* yang diperlukan unit PengelolaanPemesanan 3. Penguji mengeksekusi kelas Tester tersebut 4. Penguji melihat variabel *array* menuRestoran dari hasil eksekusi memiliki nilai atau NULL
Hasil yang Diharapkan	Variabel *array* menuRestoran memiliki nilai jika terdapat *record* menu restoran dan NULL jika tidak terdapat *record* menu restoran

Tabel 5.7 menjelaskan kasus uji integrasi antara unit PengelolaanPemesanan dengan Restoran.

Tabel 5.7 Kasus Uji Integrasi Unit PengelolaanPemesanan dan Restoran

Nomor Kasus Uji	INT_004
Nama Kasus Uji	Kasus uji integrasi unit PengelolaanPemesanan dan Restoran
Unit yang Terlibat	PengelolaanPemesanan dan Restoran
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa proses yang melibatkan interaksi antara unit PengelolaanPemesanan dan Restoran telah terintegrasi dengan baik
Prosedur Uji	1. Penguji menyiapkan kelas Tester untuk mengaktifkan unit PengelolaanPemesanan 2. Penguji menyiapkan *input-input* yang diperlukan unit PengelolaanPemesanan 3. Penguji mengeksekusi kelas Tester tersebut 4. Penguji melihat variabel *array* restoran dari hasil eksekusi memiliki nilai atau NULL
Hasil yang Diharapkan	Variabel *array* menuRestoran memiliki nilai jika terdapat *record* restoran dan NULL jika tidak terdapat *record* restoran

Tabel 5.8 menjelaskan kasus uji integrasi antara kelas Pemesanan dengan DetailPesan.

Tabel 5.8 Kasus Uji Integrasi Unit Pemesanan dan DetailPesan

Nomor Kasus Uji	INT_005
Nama Kasus Uji	Kasus uji integrasi unit Pemesanan dan DetailPesan

Unit yang Terlibat	Pemesanan dan DetailPesan
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa proses yang melibatkan interaksi antara unit Pemesanan dan DetailPesan telah terintegrasi dengan baik
Prosedur Uji	1. Penguji menyiapkan kelas Tester untuk mengaktifkan unit Pemesanan 2. Penguji menyiapkan *input-input* yang diperlukan unit Pemesanan dan DetailPesan 3. Penguji mengeksekusi kelas Tester tersebut 4. Penguji mengecek *record* detail pesan memiliki nilai atau NULL
Hasil yang Diharapkan	*Record* detail pesan memiliki nilai jika *input* valid dan NULL jika *input* tidak valid

Tabel 5.9 menjelaskan kasus uji integrasi antara unit PengelolaanPembayaran dengan Pembayaran.

Tabel 5.9 Kasus Uji Integrasi Unit PengelolaanPembayaran dan Pembayaran

Nomor Kasus Uji	INT_006
Nama Kasus Uji	Kasus uji integrasi unit PengelolaanPembayaran dan Pembayaran
Unit yang Terlibat	PengelolaanPembayaran dan Pembayaran
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa proses yang melibatkan interaksi antara unit PengelolaanPembayaran dan Pembayaran telah terintegrasi dengan baik
Prosedur Uji	1. Penguji menyiapkan kelas Tester untuk mengaktifkan unit PengelolaanPembayaran 2. Penguji menyiapkan *input-input* yang diperlukan unit PengelolaanPembayaran dan Pembayaran 3. Penguji mengeksekusi kelas Tester tersebut 4. Penguji melihat kembalian dari hasil eksekusi bernilai *true* atau *fase*
Hasil yang Diharapkan	Nilai kembalian bernilai *true* jika seluruh *input* valid dan nilai kembalian bernilai *false* jika terdapat *input* yang tidak valid

Tabel 5.10 menjelaskan kasus uji integrasi antara unit PengelolaanPembayaran dengan ModAutentikasi.

Tabel 5.10 Kasus Uji Integrasi Unit PengelolaanPembayaran dan ModAutentikasi

Nomor Kasus Uji	INT_007

Nama Kasus Uji	Kasus uji integrasi unit PengelolaanPembayaran dan ModAutentikasi
Unit yang Terlibat	PengelolaanPembayaran dan ModAutentikasi
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa proses yang melibatkan interaksi antara unit PengelolaanPembayaran dan ModAutentikasi telah terintegrasi dengan baik
Prosedur Uji	1. Penguji menyiapkan kelas Tester untuk mengaktifkan unit PengelolaanPembayaran 2. Penguji menyiapkan variabel *session* yang diperlukan unit PengelolaanPembayaran dan ModAutentikasi 3. Penguji mengeksekusi kelas Tester tersebut 4. Penguji melihat kembalian dari hasil eksekusi bernilai 'Kasir' atau NULL
Hasil yang Diharapkan	Nilai kembalian bernilai 'Kasir' jika variabel *session* valid dan nilai kembalian NULL jika variabel *session* tidak valid

Tabel 5.11 menjelaskan kasus uji integrasi antara unit PengelolaanPembayaran dengan Pembayaran.

Tabel 5.11 Kasus Uji Integrasi Unit PengelolaanPembayaran dan Pembayaran

Nomor Kasus Uji	INT_008
Nama Kasus Uji	Kasus uji integrasi unit PengelolaanPembayaran dan Pembayaran
Unit yang Terlibat	PengelolaanPembayaran dan Pembayaran
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa proses yang melibatkan interaksi antara unit PengelolaanPembayaran dan Pembayaran telah terintegrasi dengan baik
Prosedur Uji	1. Penguji menyiapkan kelas Tester untuk mengaktifkan unit PengelolaanPembayaran 2. Penguji menyiapkan *input-input* yang diperlukan unit PengelolaanPembayaran 3. Penguji mengeksekusi kelas Tester tersebut 4. Penguji melihat variabel *array* tagihanPembayaran dari hasil eksekusi memiliki nilai atau NULL
Hasil yang Diharapkan	Variabel *array* tagihanPembayaran memiliki nilai jika terdapat *record* tagihan pembayaran dan NULL jika tidak terdapat *record* tagihan pembayaran

Tabel 5.12 menjelaskan kasus uji integrasi antara unit Pembayaran dengan DetailBayar.

Tabel 5.12 Kasus Uji Integrasi Unit Pembayaran dan DetailBayar

Nomor Kasus Uji	INT_009
Nama Kasus Uji	Kasus uji integrasi unit Pembayaran dan DetailBayar
Unit yang Terlibat	Pembayaran dan DetailBayar
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa proses yang melibatkan interaksi antara unit Pembayaran dan DetailBayar telah terintegrasi dengan baik
Prosedur Uji	1. Penguji menyiapkan kelas Tester untuk mengaktifkan unit Pembayaran 2. Penguji menyiapkan *input-input* yang diperlukan unit Pembayaran dan DetailBayar 3. Penguji mengeksekusi kelas Tester tersebut 4. Penguji mengecek *record* detail bayar memiliki nilai atau NULL
Hasil yang Diharapkan	*Record* detail bayar memiliki nilai jika *input* valid dan NULL jika *input* tidak valid

5.2.3 Hasil Pengujian

Tabel 5.13 menjelaskan hasil pengujian integrasi yang telah dilakukan.

Tabel 5.13 Hasil Pengujian Integrasi

No. Kasus Uji	Nama Kasus Uji	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
INT_001	Kasus uji integrasi unit Pengelolaan Pemesanan dan Pemesanan	Nilai kembalian bernilai *true* jika seluruh *input* valid dan nilai kembalian bernilai *false* jika terdapat *input* yang tidak valid	Nilai kembalian bernilai *true* jika seluruh *input* valid dan nilai kembalian bernilai *false* jika terdapat *input* yang tidak valid	Valid
INT_002	Kasus uji integrasi unit Pengelolaan Pemesanan dan ModAutentikasi	Nilai kembalian bernilai 'Pelayan' jika variabel *session* valid dan nilai kembalian NULL jika variabel *session* tidak valid	Nilai kembalian bernilai 'Pelayan' jika variabel *session* valid dan nilai kembalian NULL jika variabel *session* tidak valid	Valid

INT_003	Kasus uji integrasi unit Pengelolaan Pemesanan dan MenuRestoran	Variabel *array* menuRestoran memiliki nilai jika terdapat *record* menu restoran dan NULL jika tidak terdapat *record* menu restoran	Variabel *array* menuRestoran memiliki nilai jika terdapat *record* menu restoran dan NULL jika tidak terdapat *record* menu restoran	Valid
INT_004	Kasus uji integrasi unit Pengelolaan Pemesanan dan Restoran	Variabel *array* menuRestoran memiliki nilai jika terdapat *record* restoran dan NULL jika tidak terdapat *record* restoran	Variabel *array* menuRestoran memiliki nilai jika terdapat *record* restoran dan NULL jika tidak terdapat *record* restoran	Valid
INT_005	Kasus uji integrasi unit Pemesanan dan DetailPesan	*Record* detail pesan memiliki nilai jika *input* valid dan NULL jika *input* tidak valid	*Record* detail pesan memiliki nilai jika *input* valid dan NULL jika *input* tidak valid	Valid
INT_006	Kasus uji integrasi unit Pengelolaan Pembayaran dan Pembayaran	Nilai kembalian bernilai *true* jika seluruh *input* valid dan nilai kembalian bernilai *false* jika terdapat *input* yang tidak valid	Nilai kembalian bernilai *true* jika seluruh *input* valid dan nilai kembalian bernilai *false* jika terdapat *input* yang tidak valid	Valid
INT_007	Kasus uji integrasi unit Pengelolaan Pembayaran dan ModAutentikasi	Nilai kembalian bernilai 'Kasir' jika variabel *session* valid dan nilai kembalian NULL jika variabel *session* tidak valid	Nilai kembalian bernilai 'Kasir' jika variabel *session* valid dan nilai kembalian NULL jika variabel *session* tidak valid	Valid
INT_008	Kasus uji integrasi unit Pengelolaan Pembayaran dan	Variabel *array* tagihanPembayaran memiliki nilai jika terdapat *record* tagihan pembayaran dan NULL jika tidak	Variabel *array* tagihanPembayaran memiliki nilai jika terdapat *record* tagihan pembayaran dan NULL jika tidak	Valid

	Pembayaran	terdapat *record* tagihan pembayaran	terdapat *record* tagihan pembayaran	
INT_009	Kasus uji integrasi unit Pembayaran dan DetailBayar	*Record* detail bayar memiliki nilai jika *input* valid dan NULL jika *input* tidak valid	*Record* detail bayar memiliki nilai jika *input* valid dan NULL jika *input* tidak valid	Valid

5.2.4 Analisis

Berdasarkan hasil pengujian integrasi, proses inti yang melibatkan interaksi antar kelas pada sistem telah terintegrasi dengan baik. Sehingga dapat dikatakan sistem pengelolaan restoran terintegrasi (OtseResto) lolos uji integrasi. Maka dari itu tidak diperlukan adanya perbaikan proses integrasi antar kelas sebab seluruh kasus uji integrasi telah valid.

5.3 Pengujian Validasi

5.3.1 Tujuan

Tujuan dilakukannya pengujian validasi adalah untuk memastikan setiap spesifikasi kebutuhan perangkat lunak yang didefinisikan telah sesuai dengan yang diharapkan.

5.3.2 Mekanisme

Setiap spesifikasi kebutuhan perangkat lunak yang didefinisikan pada tahap analisis kebutuhan akan diujikan pada pengujian ini dengan cara mendefinisikan kasus uji terhadap setiap kebutuhan tersebut lalu membandingkannya dengan hasil yang diperoleh.

Tabel 5.14 menjelaskan kasus uji dari pengujian validasi pada fungsi membuat pemesanan menu.

Tabel 5.14 Kasus Uji Validasi Membuat Pemesanan Menu

Nomor Kasus Uji	VAL_001
Nama Kasus Uji	Kasus uji validasi membuat pemesanan menu
Nomor SKPL	SKPL_01_1
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan pemesanan menu makanan dan minuman yang tersedia pada restoran
Prosedur Uji	1. Penguji masuk ke halaman utama pelayan 2. Penguji menekan tombol pemesanan menu 3. Penguji mengisi *form* dengan variasi *input* valid dan tidak valid kemudian mengeklik tombol Buat

Hasil yang Diharapkan	Pesanan menu makanan dan minuman disimpan ke dalam sistem

Tabel 5.15 menjelaskan kasus uji dari pengujian validasi pada fungsi melihat status pemesanan menu.

Tabel 5.15 Kasus Uji Validasi Melihat Status Pemesanan Menu

Nomor Kasus Uji	VAL_002
Nama Kasus Uji	Kasus uji validasi melihat status pemesanan menu
Nomor SKPL	SKPL_01_2
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melihat status dari pemesanan menu makanan dan minuman
Prosedur Uji	1. Penguji masuk ke halaman utama pelayan 2. Penguji menekan tombol status pemesanan menu
Hasil yang Diharapkan	Sistem menampilkan status pemesanan menu makanan dan minuman yang aktif secara *real-time*

Tabel 5.16 menjelaskan kasus uji dari pengujian validasi pada fungsi membuat tagihan pembayaran.

Tabel 5.16 Kasus Uji Validasi Membuat Tagihan Pembayaran

Nomor Kasus Uji	VAL_003
Nama Kasus Uji	Kasus uji validasi membuat tagihan pembayaran
Nomor SKPL	SKPL_02_1
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat membuat tagihan pembayaran dari tagihan-tagihan yang aktif
Prosedur Uji	1. Penguji menekan tombol proses pada tagihan yang ingin diproses di halaman tagihan pembayaran 2. Penguji mengisi *form* dengan variasi *input* valid dan tidak valid lalu mengeklik tombol Cetak
Hasil yang Diharapkan	Pesanan menu makanan atau minuman disimpan ke dalam sistem

Tabel 5.17 menjelaskan kasus uji dari pengujian validasi pada fungsi melihat tagihan pembayaran.

Tabel 5.17 Kasus Uji Validasi Melihat Tagihan Pembayaran

Nomor Kasus Uji	VAL_004
Nama Kasus Uji	Kasus uji validasi melihat tagihan pembayaran
Nomor SKPL	SKPL_02_2
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melihat daftar tagihan pembayaran yang aktif

Prosedur Uji	1. Penguji masuk ke halaman utama kasir 2. Penguji menekan tombol tagihan pembayaran
Hasil yang Diharapkan	Sistem dapat menampilkan daftar tagihan pembayaran secara *real-time*

Tabel 5.18 menjelaskan kasus uji dari pengujian validasi pada fungsi melihat daftar pesanan.

Tabel 5.18 Kasus Uji Validasi Melihat Daftar Pesanan

Nomor Kasus Uji	VAL_005
Nama Kasus Uji	Kasus uji validasi melihat daftar pesanan
Nomor SKPL	SKPL_03
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melihat daftar pesanan yang harus disiapkan oleh koki
Prosedur Uji	1. Penguji masuk ke halaman utama koki 2. Penguji menekan tombol daftar pesanan
Hasil yang Diharapkan	Sistem dapat menampilkan daftar pesanan secara *real-time*

Tabel 5.19 menjelaskan kasus uji dari pengujian validasi pada fungsi notifikasi pesanan.

Tabel 5.19 Kasus Uji Validasi Notifikasi Pesanan

Nomor Kasus Uji	VAL_006
Nama Kasus Uji	Kasus uji validasi notifikasi pesanan
Nomor SKPL	SKPL_04
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memberikan notifikasi pesanan kepada pengguna
Prosedur Uji	1. Penguji masuk ke halaman daftar pesanan 2. Penguji menunggu notifikasi pesanan yang muncul di halaman daftar pesanan
Hasil yang Diharapkan	Sistem memberikan notifikasi pesanan berupa perubahan warna daftar pesanan

Tabel 5.20 menjelaskan kasus uji dari pengujian validasi pada fungsi mencetak tagihan pembayaran.

Tabel 5.20 Kasus Uji Validasi Mencetak Tagihan Pembayaran

Nomor Kasus Uji	VAL_007
Nama Kasus Uji	Kasus uji validasi mencetak tagihan pembayaran
Nomor SKPL	SKPL_02_3

Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat mencetak tagihan pembayaran yang telah diproses
Prosedur Uji	1. Penguji masuk ke halaman pembuatan tagihan pembayaran 2. Penguji menekan tombol cetak pada tagihan yang diproses
Hasil yang Diharapkan	Sistem mencetak tagihan pembayaran dalam bentuk PDF

Tabel 5.21 menjelaskan kasus uji dari pengujian validasi pada fungsi mencari pesanan.

Tabel 5.21 Kasus Uji Validasi Mencari Pesanan

Nomor Kasus Uji	VAL_008
Nama Kasus Uji	Kasus uji validasi mencari pesanan
Nomor SKPL	SKPL_05
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat mencari pesanan yang belum diantarkan pelayan
Prosedur Uji	1. Penguji masuk ke halaman utama koki 2. Penguji menekan tombol pencarian pesanan 3. Penguji mengisi *form* dengan variasi *input* valid dan tidak valid lalu mengeklik tombol bergambar ikon *zoom*
Hasil yang Diharapkan	Sistem menampilkan hasil pencarian pesanan

Tabel 5.22 menjelaskan kasus uji dari pengujian validasi pada fungsi memvalidasi pesanan.

Tabel 5.22 Kasus Uji Validasi Memvalidasi Pesanan

Nomor Kasus Uji	VAL_009
Nama Kasus Uji	Kasus uji validasi memvalidasi pesanan
Nomor SKPL	SKPL_06_1
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memvalidasi pesanan yang telah disiapkan oleh koki
Prosedur Uji	1. Penguji masuk ke halaman daftar pesanan 2. Penguji mengeklik tombol validasi pada pesanan yang telah disiapkan oleh koki
Hasil yang Diharapkan	Sistem mengubah status pesanan sekaligus mencetak lembar pesanan

Tabel 5.23 menjelaskan kasus uji dari pengujian validasi pada fungsi mencetak pesanan.

Tabel 5.23 Kasus Uji Validasi Mencetak Pesanan

Nomor Kasus Uji	VAL_010
Nama Kasus Uji	Kasus uji validasi mencetak pesanan
Nomor SKPL	SKPL_06_2
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat mencetak lembar pesanan
Prosedur Uji	1. Penguji melihat pesanan yang ingin dicetak pada halaman pencarian pesanan 2. Penguji mengeklik tombol cetak
Hasil yang Diharapkan	Sistem mencetak lembar pesanan

Tabel 5.24 menjelaskan kasus uji dari pengujian validasi pada fungsi kelola menu restoran.

Tabel 5.24 Kasus Uji Validasi Kelola Menu Restoran

Nomor Kasus Uji	VAL_011
Nama Kasus Uji	Kasus uji validasi kelola menu restoran
Nomor SKPL	SKPL_07
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan pengelolaan menu makanan dan minuman yang tersedia pada restoran
Prosedur Uji	1. Penguji masuk ke halaman utama manajer 2. Penguji masuk ke halaman tambah menu restoran jika ingin menambah menu makanan atau minuman 3. Penguji masuk ke halaman daftar menu restoran jika ingin melihat daftar menu makanan dan minuman yang tersedia 4. Penguji masuk ke halaman ubah menu restoran dengan mengeklik tombol ubah pada halaman daftar menu restoran jika ingin mengubah menu makanan atau minuman 5. Penguji mengeklik tombol hapus pada halaman daftar menu restoran jika ingin menghapus menu makanan atau minuman
Hasil yang Diharapkan	Sistem dapat membuat, melihat, mengubah, dan menghapus menu makanan dan minuman pada sistem

Tabel 5.25 menjelaskan kasus uji dari pengujian validasi pada fungsi kelola pengguna.

Tabel 5.25 Kasus Uji Validasi Kelola Pengguna

Nomor Kasus Uji	VAL_012
Nama Kasus Uji	Kasus uji validasi kelola pengguna
Nomor SKPL	SKPL_08_1
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan pengelolaan pengguna yang dapat mengakses sistem
Prosedur Uji	1. Penguji masuk ke halaman utama manajer 2. Penguji masuk ke halaman tambah pengguna jika ingin menambah pengguna 3. Penguji masuk ke halaman daftar pengguna jika ingin melihat daftar pengguna 4. Penguji masuk ke halaman ubah pengguna dengan mengeklik tombol ubah pada halaman daftar pengguna jika ingin mengubah pengguna 5. Penguji mengeklik tombol hapus pada halaman daftar pengguna jika ingin menghapus pengguna
Hasil yang Diharapkan	Sistem dapat membuat, melihat, mengubah, dan menghapus data pengguna pada sistem

Tabel 5.26 menjelaskan kasus uji dari pengujian validasi pada fungsi mengubah informasi restoran.

Tabel 5.26 Kasus Uji Validasi Mengubah Informasi Restoran

Nomor Kasus Uji	VAL_013
Nama Kasus Uji	Kasus uji validasi mengubah informasi restoran
Nomor SKPL	SKPL_08_2
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan pengubahan informasi restoran
Prosedur Uji	1. Penguji masuk ke halaman utama manajer 2. Penguji mengeklik ubah informasi restoran 3. Penguji mengubah isi *form* dengan variasi *input* valid dan tidak valid lalu mengeklik tombol ubah
Hasil yang Diharapkan	Data informasi restoran disimpan oleh sistem

Tabel 5.27 menjelaskan kasus uji dari pengujian validasi pada fungsi melihat histori transaksi.

Tabel 5.27 Kasus Uji Validasi Melihat Histori Transaksi

Nomor Kasus Uji	VAL_014
Nama Kasus Uji	Kasus uji validasi melihat histori transaksi
Nomor SKPL	SKPL_09

Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melihat seluruh histori transaksi kasir
Prosedur Uji	1. Penguji masuk ke halaman utama manajer 2. Penguji mengeklik tombol daftar histori transaksi
Hasil yang Diharapkan	Sistem menampilkan seluruh histori transaksi

Tabel 5.28 menjelaskan kasus uji dari pengujian validasi pada proses login.

Tabel 5.28 Kasus Uji Validasi Login

Nomor Kasus Uji	VAL_015
Nama Kasus Uji	Kasus uji validasi login
Nomor SKPL	SKPL_10
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan autentikasi pengguna yaitu proses login
Prosedur Uji	1. Penguji masuk ke halaman login sistem 2. Penguji memasukkan *input* yang valid dan tidak valid 3. Penguji mengeklik tombol Log In
Hasil yang Diharapkan	Sistem melakukan proses autentikasi dengan benar

Tabel 5.29 menjelaskan kasus uji dari pengujian validasi pada kebutuhan non fungsional *reliability*.

Tabel 5.29 Kasus Uji Validasi *Reliability*

Nomor Kasus Uji	VAL_016
Nama Kasus Uji	Kasus uji validasi *reliability*
Nomor SKPL	SKPL_11
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melayani perintah pengguna dengan batas toleransi *error* hingga 5%.
Prosedur Uji	Prosedur uji mengacu kepada mekanisme pengujian ISO 9126 pada karakteristik *reliability*
Hasil yang Diharapkan	Sistem memiliki ketersediaan lebih dari sama dengan 95%

Tabel 5.30 menjelaskan kasus uji dari pengujian validasi pada kebutuhan non fungsional *usability*.

Tabel 5.30 Kasus Uji Validasi *Usability*

Nomor Kasus Uji	VAL_017
Nama Kasus Uji	Kasus uji validasi *usability*
Nomor SKPL	SKPL_12

Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat digunakan dengan mudah oleh pengguna.
Prosedur Uji	Prosedur uji mengacu kepada mekanisme pengujian ISO 9126 pada karakteristik *usability*
Hasil yang Diharapkan	Sistem masuk ke dalam kriteria mudah digunakan

Tabel 5.31 menjelaskan kasus uji dari pengujian validasi pada kebutuhan non fungsional *efficiency*.

Tabel 5.31 Kasus Uji Validasi *Efficiency*

Nomor Kasus Uji	VAL_018
Nama Kasus Uji	Kasus uji validasi *efficiency*
Nomor SKPL	SKPL_13
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melayani perintah pengguna dengan waktu respon yang relatif cepat
Prosedur Uji	Prosedur uji mengacu kepada mekanisme pengujian ISO 9126 pada karakteristik *efficiency*
Hasil yang Diharapkan	Sistem memiliki waktu respon di bawah 10 detik

Tabel 5.32 menjelaskan kasus uji dari pengujian validasi pada kebutuhan non fungsional *maintainability*.

Tabel 5.32 Kasus Uji Validasi *Maintainability*

Nomor Kasus Uji	VAL_019
Nama Kasus Uji	Kasus uji validasi *Maintainability*
Nomor SKPL	SKPL_14
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat dipelihara dengan mudah
Prosedur Uji	Prosedur uji mengacu kepada mekanisme pengujian ISO 9126 pada karakteristik *maintainability*
Hasil yang Diharapkan	Sistem memiliki nilai MI lebih dari sama dengan 65

Tabel 5.33 menjelaskan kasus uji dari pengujian validasi pada kebutuhan non fungsional *portability*.

Tabel 5.33 Kasus Uji Validasi *Portability*

Nomor Kasus Uji	VAL_020
Nama Kasus Uji	Kasus uji validasi *portability*
Nomor SKPL	SKPL_15
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat dipindahkan atau dioperasikan pada lingkungan yang berbeda.

Prosedur Uji	Prosedur uji mengacu kepada mekanisme pengujian ISO 9126 pada karakteristik *portability*
Hasil yang Diharapkan	Sistem dapat dijalankan di berbagai *web browser* yang berbeda

5.3.3 Hasil Pengujian

Tabel 5.34 memaparkan hasil pengujian validasi yang telah dilakukan terhadap setiap kasus uji yang telah didefinisikan pada tahap sebelumnya.

Tabel 5.34 Hasil Pengujian Validasi

No. Kasus Uji	Nama Kasus Uji	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
VAL_001	Kasus uji validasi membuat pemesanan menu	Pesanan menu makanan dan minuman disimpan ke dalam sistem	Pesanan menu makanan dan minuman disimpan ke dalam sistem	Valid
VAL_002	Kasus uji validasi melihat status pemesanan menu	Sistem menampilkan status pemesanan menu makanan dan minuman yang aktif secara *real-time*	Sistem menampilkan status pemesanan menu makanan dan minuman yang aktif secara *real-time*	Valid
VAL_003	Kasus uji validasi membuat tagihan pembayaran	Pesanan menu makanan atau minuman disimpan ke dalam sistem	Pesanan menu makanan atau minuman disimpan ke dalam sistem	Valid
VAL_004	Kasus uji validasi melihat tagihan pembayaran	Sistem dapat menampilkan daftar tagihan pembayaran secara *real-time*	Sistem dapat menampilkan daftar tagihan pembayaran secara *real-time*	Valid
VAL_005	Kasus uji validasi melihat daftar pesanan	Sistem dapat menampilkan daftar pesanan secara *real-time*	Sistem dapat menampilkan daftar pesanan secara *real-time*	Valid
VAL_006	Kasus uji validasi notifikasi pesanan	Sistem memberikan notifikasi pesanan berupa perubahan warna daftar pesanan	Sistem memberikan notifikasi pesanan berupa perubahan warna daftar pesanan	Valid

VAL_007	Kasus uji validasi mencetak tagihan pembayaran	Sistem mencetak tagihan pembayaran dalam bentuk PDF	Sistem mencetak tagihan pembayaran dalam bentuk PDF	Valid
VAL_008	Kasus uji validasi mencari pesanan	Sistem menampilkan hasil pencarian pesanan	Sistem menampilkan hasil pencarian pesanan	Valid
VAL_009	Kasus uji validasi memvalidasi pesanan	Sistem mengubah status pesanan sekaligus mencetak lembar pesanan	Sistem mengubah status pesanan sekaligus mencetak lembar pesanan	Valid
VAL_010	Kasus uji validasi mencetak pesanan	Sistem mencetak lembar pesanan	Sistem mencetak lembar pesanan	Valid
VAL_011	Kasus uji validasi kelola menu restoran	Sistem dapat membuat, melihat, mengubah, dan menghapus menu makanan dan minuman pada sistem	Sistem dapat membuat, melihat, mengubah, dan menghapus menu makanan dan minuman pada sistem	Valid
VAL_012	Kasus uji validasi kelola pengguna	Sistem dapat membuat, melihat, mengubah, dan menghapus data pengguna pada sistem	Sistem dapat membuat, melihat, mengubah, dan menghapus data pengguna pada sistem	Valid
VAL_013	Kasus uji validasi mengubah informasi restoran	Data informasi restoran disimpan oleh sistem	Data informasi restoran disimpan oleh sistem	Valid
VAL_014	Kasus uji validasi melihat histori transaksi	Sistem menampilkan seluruh histori transaksi	Sistem menampilkan seluruh histori transaksi	Valid
VAL_015	Kasus uji validasi login	Sistem melakukan proses autentikasi dengan benar	Sistem melakukan proses autentikasi dengan benar	Valid
VAL_016	Kasus uji validasi *reliability*	Sistem memiliki ketersediaan lebih dari sama dengan 95%	Sistem memiliki ketersediaan sebesar 100%	Valid

VAL_017	Kasus uji validasi *usability*	Sistem masuk ke dalam kriteria mudah digunakan	Sistem masuk ke dalam kriteria mudah digunakan	Valid
VAL_018	Kasus uji validasi *efficiency*	Sistem memiliki waktu respon di bawah 10 detik	Sistem memiliki waktu respon sebesar 0,78 detik	Valid
VAL_019	Kasus uji validasi *maintainability*	Sistem memiliki nilai MI lebih dari sama dengan 65	Sistem memiliki nilai MI sebesar 66,62597	Valid
VAL_020	Kasus uji validasi *portability*	Sistem dapat dijalankan di berbagai *web browser* yang berbeda	Sistem dapat dijalankan pada *web browser* yang berbeda tanpa ditemukannya *error* maupun *warning*	Valid

5.3.4 Analisis

Berdasarkan hasil pengujian validasi, seluruh kasus uji yang didefinisikan telah valid atau berarti seluruh spesifikasi kebutuhan perangkat lunak telah terpenuhi. Sehingga dapat dikatakan sistem pengelolaan restoran terintegrasi (OtseResto) lolos uji validasi.

5.4 Pengujian Berbasis ISO 9126

Pengujian ini dilakukan pada enam karakteristik utama yang terdapat pada model kualitas ISO 9126 yaitu *functionality, reliability, usability, efficiency, maintainability,* dan *portability*.

5.4.1 *Functionality*

5.4.1.1 *Tujuan*

Tujuan dilakukannya pengujian pada karakteristik *functionality* adalah untuk memastikan kebutuhan fungsional sistem telah diimplementasikan sesuai dengan yang diharapkan.

5.4.1.2 *Mekanisme*

Karakteristik *functionality* pada sistem akan dinilai dengan melakukan pengujian validasi. Pengujian dilakukan dengan memvalidasi setiap kebutuhan fungsional untuk memastikan fungsi-fungsi telah diimplementasikan sesuai dengan yang diharapkan. Jika semua kebutuhan fungsional telah dinilai validitasnya maka akan dihitung dengan menggunakan Persamaan 5.1 untuk mendapatkan tingkat *functionality* dari sistem.

$$X = 1 - \frac{A}{B} \quad (5.1)$$

Dengan:

X = Tingkat *functionality*.

A = Jumlah fungsi yang bermasalah.

B = Jumlah fungsi yang diuji keseluruhan.

Jika tingkat *functionality* dari sistem telah mendekati angka 1 maka sistem memenuhi karakteristik *functionality*. Namun jika tingkat *functionality* masih jauh dari angka 1 maka fungsi-fungsi yang bermasalah harus diperbaiki. Berikut ini merupakan kasus uji dari pengujian validasi.

5.4.1.3 *Hasil Pengujian*

Berdasarkan hasil pengujian validasi pada Tabel 5.27, tidak ditemukan fungsi yang bermasalah dari seluruh fungsi yang diujikan. Kemudian dapat dimasukkan ke dalam persamaan untuk mencari tingkat *functionality*.

$$X = 1 - \frac{0}{15} = 1 - 0 = 1$$

Dari hasil tersebut maka didapatkan tingkat *functionality* sebesar 1.

5.4.1.4 *Analisis*

Berdasarkan hasil pengujian pada karakteristik *functionality,* didapatkan tingkat *functionality* sebesar 1. Dan untuk dapat memenuhi standar *functionality*, diperlukan tingkat *functionality* yang mendekati atau bernilai 1. Sehingga dari hasil pengujian tersebut maka sistem pengelolaan restoran terintegrasi (OtseResto) telah memenuhi standar untuk karakteristik *functionality*.

5.4.2 *Reliability*

5.4.2.1 *Tujuan*

Tujuan dilakukannya pengujian pada karakteristik *reliability* adalah untuk memastikan sistem telah memenuhi kebutuhan non fungsional pada parameter *reliability*.

5.4.2.2 *Mekanisme*

Karakteristik *reliability* pada sistem akan dinilai dengan melakukan pengujian terhadap kriteria *sessions, pages,* dan *hits*. *Sessions* merupakan skenario uji pengguna dari melakukan *request* hingga mendapatkan respons dari sistem, sedangkan *pages* merupakan halaman yang sukses diunduh, sedangkan *hits* merupakan *request* yang sukses diterima sistem.

Jumlah *sessions* yang diujikan berjumlah 300 buah dengan memanfaatkan *tool* WAPT (*Web Application Performance Tools*). Setiap *sessions* tersebut melakukan *request* terhadap enam belas kelas *controller* yang diakses oleh 80 *virtual user* secara bersamaan. Jika kriteria yang diuji telah didapatkan hasilnya maka akan

dihitung dengan menggunakan Persamaan 5.2 untuk mendapatkan nilai *reliability* dari sistem.

$$X = \frac{A}{B} \quad (5.2)$$

Dengan:

X = Nilai *reliability*.

A = Jumlah berhasil dalam pengujian.

B = Jumlah pengujian keseluruhan.

Jika nilai *reliability* dari sistem lebih dari sama dengan 95% maka sistem memenuhi karakteristik *reliability*. Namun jika nilai *reliability* sistem masih kurang dari 95% maka *request* dari pengguna harus dibatasi.

5.4.2.3 *Hasil Pengujian*

Hasil pengujian *reliability* dapat dilihat secara lengkap pada Lampiran 10. Tabel 5.35 menjelaskan ringkasan hasil pengujian *reliability*.

Tabel 5.35 Ringkasan Hasil Pengujian *Reliability*

Kriteria	Berhasil	Gagal	Total
Sessions	300	0	300
Pages	8700	0	8700
Hits	8700	0	8700

Berdasarkan ringkasan hasil pengujian *reliability* pada Tabel 5.35, seluruh kriteria yang diujikan tidak memiliki kegagalan. Kemudian dapat dimasukkan ke persamaan untuk pencarian nilai *reliability*.

a. *Sessions*

$$X = \frac{300}{300} \text{x } 100\% = 100\%$$

b. *Pages*

$$X = \frac{8700}{8700} \text{ x } 100\% = 100\%$$

c. *Hits*

$$X = \frac{8700}{8700} \text{x } 100\% = 100\%$$

Dari hasil tersebut didapatkan ketiga kriteria yaitu *sessions, pages,* dan *hits* memiliki nilai 100%.

5.4.2.4 *Analisis*

Berdasarkan hasil pengujian pada karakteristik *reliability,* didapatkan nilai *sessions, pages,* dan *hits* sebesar 100%. Dan untuk memenuhi standar *reliability*, diperlukan nilai *reliability* lebih dari sama dengan 95%. Dengan kata lain, sistem

pengelolaan restoran terintegrasi (OtseResto) telah memenuhi standar untuk karakteristik *reliability*.

5.4.3 *Usability*

5.4.3.1 *Tujuan*

Tujuan dilakukannya pengujian pada karakteristik *usability* adalah untuk memastikan sistem telah memenuhi kebutuhan non fungsional pada parameter *usability*.

5.4.3.2 *Mekanisme*

Karakteristik *usability* pada sistem akan dinilai dengan melakukan penyebaran kuesioner terhadap responden. Kuesioner yang disebar menggunakan model kuesioner SUS (*System Usability Scale*). Responden yang dipilih berjumlah lima belas orang. Jika kelima belas kuesioner tersebut telah diisi oleh responden maka akan dihitung dengan persamaan untuk model kuesioner SUS.

Poin ganjil dihitung dengan Persamaan 5.3:

$$Skor\ Ganjil = Bobot\ Jawaban - 1 \quad (5.3)$$

Poin genap dihitung dengan Persamaan 5.4:

$$Skor\ Genap = 5 - Bobot\ Jawaban \quad (5.4)$$

Skor akhir dihitung dengan Persamaan 5.5:

$$Skor\ Akhir = Jumlah\ Skor\ x\ 2{,}5 \quad (5.5)$$

Skor akhir dari model kuesioner SUS memiliki rentang nilai 0-100 dengan menggunakan Skala Likert. Yang mana jika ditranslasikan dalam bentuk interval yaitu:

1. 0 – 19,99 = Sangat Buruk
2. 20 – 39,99 = Buruk
3. 40 – 59,99 = Cukup
4. 60 – 79,99 = Baik
5. 80 – 100 = Sangat Baik

Jika sistem memiliki nilai kuesioner rata-rata lebih dari sama dengan 60 maka sistem memenuhi karakteristik *usability*. Namun jika sistem memiliki nilai kuesioner rata-rata di bawah 60 maka antarmuka sistem harus disederhanakan.

5.4.3.3 *Hasil Pengujian*

Poin-poin kuesioner SUS yang digunakan dalam pengujian *usability* ini dapat dilihat pada Lampiran 9. Hasil pengujian yang didapatkan dari lima belas responden dapat dilihat pada Tabel 5.36.

Tabel 5.36 Hasil Pengujian *Usability*

Responden	Skor Ganjil	Skor Genap	Skor Akhir
Responden 1	14	12	65
Responden 2	11	14	62,5
Responden 3	12	12	60
Responden 4	13	15	70
Responden 5	17	13	75
Responden 6	16	9	62,5
Responden 7	16	14	75
Responden 8	13	11	60
Responden 9	14	10	60
Responden 10	16	11	67,5
Responden 11	17	13	75
Responden 12	15	17	80
Responden 13	17	14	77,5
Responden 14	13	10	57,5
Responden 15	18	14	80
Rata-rata Skor Akhir			68,5

Berdasarkan hasil pengujian *usability* pada Tabel 5.36, rata-rata skor akhir dari pengujian pada karakteristik *usability* memiliki nilai 68,5.

5.4.3.4 *Analisis*

Berdasarkan hasil pengujian pada karakteristik *usability,* didapatkan rata-rata skor akhir kuesioner SUS sebesar 68,5. Sehingga jika hasil tersebut dimasukkan ke dalam interval nilai 0-100 Skala Likert maka sistem masuk ke dalam kategori baik. Dan untuk dapat memenuhi standar *usability,* diperlukan nilai lebih dari sama dengan 60. Maka dari itu dapat disimpulkan bahwa sistem pengelolaan restoran terintegrasi (OtseResto) memenuhi karakteristik *usability*.

5.4.4 *Efficiency*

5.4.4.1 *Tujuan*

Tujuan dilakukannya pengujian pada karakteristik *efficiency* adalah untuk memastikan sistem telah memenuhi kebutuhan non fungsional pada parameter *efficiency*.

5.4.4.2 *Mekanisme*

Karakteristik *efficiency* pada sistem akan dinilai dengan melakukan pengujian terhadap kriteria waktu respon (*response time*). Pengujian pada karakteristik ini menggunakan *tool* WAPT (*Web Application Performance Tools*). Pengujian ini dieksekusi bersamaan dengan pengujian *reliability*. Beban sistem yang diujikan sama dengan pengujian *reliability* yaitu sebesar 80 *virtual user*. Jika kriteria yang diuji telah didapatkan nilainya maka akan dibandingkan dengan standar waktu

respon Jakob Nielsen. Jika sistem memiliki waktu respon kurang dari 10 detik maka sistem memenuhi karakteristik *efficiency*. Namun jika sistem memiliki waktu respons lebih dari sama dengan 10 detik maka proses atau antarmuka pada sistem harus disederhanakan.

5.4.4.3 *Hasil Pengujian*

Hasil pengujian *efficiency* dapat dilihat secara lengkap pada Lampiran 11. Tabel 5.37 menjelaskan ringkasan hasil pengujian *efficiency*.

Tabel 5.37 Ringkasan Hasil Pengujian *Efficiency*

URL	Rata-rata Hasil (Detik)
http://localhost/otseresto/autentikasi	0,34
http://localhost/otseresto/autentikasi/login	1,04
http://localhost/otseresto/autentikasi/logout	0,72
http://localhost/otseresto/kelolaPengguna/	1,08
http://localhost/otseresto/kelolaPengguna/membuatPengguna	0,74
http://localhost/otseresto/kelolaPengguna/melihatPengguna	0,74
http://localhost/otseresto/kelolaPengguna/mengubahPengguna	0,78
http://localhost/otseresto/kelolaPengguna/menghapusPengguna	0,77
http://localhost/otseresto/kelolaMenuRestoran/	0,73
http://localhost/otseresto/kelolaMenuRestoran/membuatMenuRestoran	0,73
http://localhost/otseresto/kelolaMenuRestoran/melihatMenuRestoran	0,73
http://localhost/otseresto/kelolaMenuRestoran/mengubahMenuRestoran	0,75
http://localhost/otseresto/kelolaMenuRestoran/menghapusMenuRestoran	0,76
http://localhost/otseresto/kelolaInformasiRestoran	1,11
http://localhost/otseresto/kelolaInformasiRestoran/mengubahInformasiRestoran	0,75
http://localhost/otseresto/kelolaPemesanan	0,76
http://localhost/otseresto/kelolaPemesanan/membuatPemesananMenu	0,77
http://localhost/otseresto/kelolaPemesanan/melihatStatusPemesananMenu	0,75
http://localhost/otseresto/kelolaDapur	0,76
http://localhost/otseresto/kelolaDapur/melihatDaftarPesanan	0,77
http://localhost/otseresto/kelolaDapur/memberikanNotifikasiPesanan	0,77
http://localhost/otseresto/kelolaDapur/memvalidasiPesanan	0,79
http://localhost/otseresto/kelolaDapur/mencariPesanan	0,76
http://localhost/otseresto/kelolaDapur/mencetakPesanan	0,78
http://localhost/otseresto/kelolaPembayaran	0,75

http://localhost/otseresto/kelolaPembayaran/membuatTagihanPembayaran	0,77
http://localhost/otseresto/kelolaPembayaran/melihatTagihanPembayaran	0,78
http://localhost/otseresto/kelolaPembayaran/mencetakTagihanPembayaran	0,81
http://localhost/otseresto/kelolaPembayaran/melihatHistoriTransaksi	0,79
Rata-Rata Seluruh URL (Detik)	0,78

Berdasarkan ringkasan hasil pengujian *efficiency* pada Tabel 5.37, nilai rata-rata waktu repon sistem seluruh URL adalah 0,78 detik.

5.4.4.4 *Analisis*

Berdasarkan hasil pengujian pada karakteristik *efficiency,* didapatkan rata-rata waktu respon sistem sebesar 0,78 detik. Sehingga jika hasil tersebut dibandingkan dengan standar Jakob Nielsen, maka dapat dikatakan pengguna merasakan adanya jeda namun masih fokus dalam sistem. Dan untuk dapat memenuhi standar *efficiency*, dibutuhkan nilai kurang dari 10 detik. Maka dari itu dapat disimpulkan bahwa sistem pengelolaan restoran terintegrasi (OtseResto) memenuhi karakteristik *usability*.

5.4.5 *Maintainability*

5.4.5.1 *Tujuan*

Tujuan dilakukannya pengujian pada karakteristik *maintainability* adalah untuk memastikan sistem telah memenuhi kebutuhan non fungsional pada parameter *maintainability*.

5.4.5.2 *Mekanisme*

Karakteristik *maintainability* pada sistem akan dinilai dengan menggunakan persamaan MI (*Maintainability Index*). Perhitungan MI dapat dilakukan dengan menggunakan Persamaan 5.6.

$$MI = 171 - 5{,}2\ x\ \ln(aveV) - 0{,}23\ x\ aveV(g) - 16{,}2\ x\ \ln(aveLOC) \quad (5.6)$$

Dengan:

MI = Maintainability Index.

aveV = Rata-rata *Halstead Volume*.

aveV(g) = Rata-rata *Cyclomatic Complexity*.

aveLOC = Rata-rata *Line of Code*.

Pada pengujian *maintainability* ini digunakan *tool* DMSSearchEngine yang berfungsi untuk mencari *Halstead Volume* (aveV) dan *Line of Code* (aveLOC). Sedangkan untuk pencarian nilai *Cyclomatic Complexity* (aveV(g)) dilakukan dengan perhitungan persamaan V(g) terhadap setiap fungsi pada kode sistem yang telah diimplementasikan. Jika sistem memiliki nilai MI lebih dari sama dengan 65 maka sistem memenuhi karakteristik *maintainability*. Namun jika sistem memiliki nilai MI kurang dari 65 maka kode sistem harus disederhanakan.

5.4.5.3 *Hasil Pengujian*

Hasil pengujian *maintainability* dapat dilihat secara lengkap pada Lampiran 12. Tabel 5.38 menjelaskan ringkasan hasil pengujian *maintainability*.

Tabel 5.38 Ringkasan Hasil Pengujian *Maintainability*

Nama Kelas	*Halstead Volume*	*Lines of Code*	*Rata-Rata Cyclomatic Complexity*
Autentikasi	948,918	42	3,5
Dapur	1363,511	41	1,4
DetailBayar	559,092	18	2
DetailPesan	559,092	18	2
KelolaDapur	3392,859	111	3,857143
KelolaInformasiRestoran	540,484	29	2
KelolaMenuRestoran	1765,234	65	2,833333
KelolaPembayaran	3911,732	113	3,666667
KelolaPemesanan	1718,209	62	3,75
KelolaPengguna	1832,468	70	2,833333
MenuRestoran	2061,612	56	2
ModAutentikasi	371,385	16	2,5
Pembayaran	4262,312	94	2,6
Pemesanan	2700,053	60	5
Pengguna	1522,384	52	1,428571
Restoran	921,519	28	3
Rata - rata	1776,929	54,6875	2,773065

Berdasarkan ringkasan hasil pengujian *maintainability* pada Tabel 5.38, nilai rata-rata *Halstead Volume* (aveV) sebesar 1776,929, rata-rata *Lines of Code* (aveLOC) sebesar 54,6875, dan rata-rata seluruh *Cyclomatic Complexity* (aveV(g)) sebesar 2,773065. Kemudian dapat dimasukkan ke persamaan untuk pencarian nilai MI (*Maintainability Index*).

$$MI = 171 - 5{,}2\ x \ln(aveV) - 0{,}23\ x\ aveV(g) - 16{,}2\ x \ln(aveLOC)$$
$$= 171 - 5{,}2\ x \ln(1776{,}929) - 0{,}23\ x\ aveV(2{,}773065) - 16{,}2\ x \ln(54{,}6875)$$
$$= 66{,}62597$$

Dari hasil tersebut didapatkan nilai MI sebesar 66,62597.

5.4.5.4 *Analisis*

Berdasarkan hasil pengujian pada karakteristik *maintainability*, didapatkan nilai MI (*Maintainability Index*) sebesar 66,62597. Sehingga jika hasil tersebut dibandingkan dengan standar MI, maka sistem masuk ke dalam kategori *medium* MI dengan sifat *moderate maintainable*. Dan untuk dapat memenuhi standar *maintainability*, dibutuhkan nilai MI lebih dari sama dengan 65. Maka dari itu dapat disimpulkan bahwa sistem pengelolaan restoran terintegrasi (OtseResto) memenuhi karakteristik *maintainability*.

5.4.6 *Portability*

5.4.6.1 *Tujuan*

Tujuan dilakukannya pengujian pada karakteristik *portability* adalah untuk memastikan sistem telah memenuhi kebutuhan non fungsional pada parameter *portability*.

5.4.6.2 *Mekanisme*

Karakteristik *portability* pada sistem akan dinilai dengan melakukan pengujian pada kompatibilitas antar *browser*. Pengujian pada karakteristik ini menggunakan *tool* BrowseEmAll. Sistem diuji menggunakan tujuh *desktop browser* dan lima *mobile browser* secara virtual dengan BrowseEmAll. Setiap antarmuka sistem akan diujikan melalui *tools* tersebut. Jika hasil dari BrowseEmAll tidak menunjukkan adanya *error* dan *warning* maka sistem memenuhi karakteristik *portability*. Namun jika terdapat *error* ataupun *warning* maka *file view* harus diperbaiki.

5.4.6.3 *Hasil Pengujian*

Hasil pengujian *portability* dapat dilihat secara lengkap pada Lampiran 13 hingga Lampiran 39. Tabel 5.39 menjelaskan ringkasan hasil pengujian *portability*.

Tabel 5.39 Ringkasan Hasil Pengujian *Portability*

URL	Status
http://localhost/otseresto	Valid
http://localhost/otseresto/kelolaPemesanan	Valid
http://localhost/otseresto/kelolaPemesanan/membuatPemesananMenu	Valid
http://localhost/otseresto/kelolaPemesanan/melihatStatusPemesananMenu	Valid
http://localhost/otseresto/kelolaPembayaran	Valid
http://localhost/otseresto/kelolaPembayaran/membuatTagihanPembayaran	Valid
http://localhost/otseresto/kelolaPembayaran/melihatTagihanPembayaran	Valid
http://localhost/otseresto/kelolaDapur	Valid
http://localhost/otseresto/kelolaDapur/melihatDaftarPesanan	Valid
http://localhost/otseresto/kelolaDapur/mencariPesanan	Valid
http://localhost/otseresto/kelolaMenuRestoran	Valid
http://localhost/otseresto/kelolaMenuRestoran/membuatMenuRestoran	Valid
http://localhost/otseresto/kelolaMenuRestoran/melihatMenuRestoran	Valid
http://localhost/otseresto/kelolaMenuRestoran/mengubahMenuRestoran	Valid
http://localhost/otseresto/kelolaPengguna/membuatPengguna	Valid
http://localhost/otseresto/kelolaPengguna/melihatPengguna	Valid
http://localhost/otseresto/kelolaPengguna/mengubahPengguna	Valid
http://localhost/otseresto/kelolaInformasiRestoran/mengubahInformasiRestoran	Valid

http://localhost/otseresto/kelolaPembayaran/melihatHistoriTransaksi	Valid

Berdasarkan ringkasan hasil pengujian *portability* pada Tabel 5.39, seluruh halaman sistem tidak ditemukan adanya *error* maupun *warning*.

5.4.6.4 *Analisis*

Berdasarkan hasil pengujian pada karakteristik *portability,* sistem tidak memiliki *error* dan *warning* di tujuh *desktop browser* dan lima *mobile browser*. Dan untuk dapat memenuhi standar *portability*, diperlukan sistem yang tidak memiliki *error* dan *warning* jika dioperasikan di berbagai *browser.* Sehingga dari hasil pengujian tersebut maka sistem pengelolaan restoran terintegrasi (OtseResto) telah memenuhi standar untuk karakteristik *portability*.

BAB VI
PENUTUP

6.1 Kesimpulan

Berdasarkan hasil penelitian skripsi yang telah dilakukan maka dapat ditarik kesimpulan sebagai berikut:

1. Analisis sistem pengelolaan restoran terintegrasi (OtseResto) telah menghasilkan proses yang terintegrasi yaitu pemesanan dan pembayaran menu makanan. Proses pemesanan melibatkan interaksi antara area POS (*Point of Sales*) bagian pemesanan dengan area dapur sedangkan proses pembayaran melibatkan interaksi antara area POS bagian pemesanan dengan area POS bagian pembayaran.
2. Perancangan sistem pengelolaan restoran terintegrasi (OtseResto) telah menghasilkan desain dari proses pemesanan dan pembayaran menu makanan yang dibagi berdasarkan masing-masing area fungsional.
3. Implementasi desain dari proses terintegrasi pada sistem ini telah menghasilkan fitur pemesanan dan pembayaran menu makanan.
4. Berdasarkan hasil pengujian berbasis ISO 9126, sistem pengelolaan restoran terintegrasi (OtseResto) telah memenuhi standar dari karakteristik utama ISO 9126. Pada karakteristik *functionality* yang diujikan dengan pengujian validasi, didapatkan nilai 1 yang berarti seluruh fungsi berjalan dengan baik. Sedangkan pada karakteristik *reliability* yang diujikan dengan WAPT (*Web Application Performance Tools*), didapatkan nilai sebesar 100%. Kemudian pada karakteristik *usability* yang diujikan dengan menggunakan model kuesioner SUS (*System Usability Scale*), didapatkan nilai sebesar 68,5 yang berarti masuk ke dalam kategori baik. Lalu pada karakteristik *efficiency* yang diujikan dengan WAPT, didapatkan nilai rata-rata waktu respon sistem sebesar 0,78 detik. Kemudian pada karakteristik *maintainability* yang diujikan dengan persamaan MI (*Maintaintability Index*), didapatkan nilai sebesar 66,62597 yang masuk ke dalam kategori *moderate maintainable.* Dan pada karakteristik *portability* yang diujikan dengan BrowseEmAll, didapatkan hasil bahwa tidak ditemukan *error* maupun *warning* pada tujuh jenis *desktop browser* dan lima jenis *mobile browser*.

6.2 Saran

Saran yang dapat diberikan setelah menyelesaikan penelitian skripsi ini yaitu:

1. Untuk pengembangan lebih lanjut dapat dikembangkan fitur pada area fungsional yang sudah ada maupun yang belum tercakup dalam sistem sehingga setiap sumber daya yang ada pada restoran dapat dikelola lebih baik.

2. Untuk pengembangan lebih lanjut dapat dikembangkan aplikasi *mobile* untuk pelayan dan tamu restoran yang terhubung langsung ke sistem sehingga mobilitas sistem lebih baik.
3. Untuk mendapatkan hasil pengujian yang lebih detail dapat dilakukan dengan menambahkan sub karakteristik yang ada pada model kualitas ISO 9126 sebagai parameter tambahan.

DAFTAR PUSTAKA

Abeysinghe, S., 2009. *PHP Team Development: Easy and Effective Team Work using MVC, Agile Development, Source Control, Testing, Bug Tracking, and More.* Birmingham: PACKT Publishing.

Al-Safadi, L. A. & Garcia, R. A., 2012. ISO9126 Based Quality Model for Evaluating B2C e-Commerce Applications - A Saudi Market Perspective. *IJCIT,* 3(2), pp. 8-15.

Asthana, A. & Olivieri, J., 2009. *Quantifying Software Reliability and Readiness.* Naples, IEEE, pp. 1-6.

Atmodjo, M. W., 2005. *Restoran dan Segala Permasalahannya.* Yogyakarta: ANDI.

Broke, J., 1996. SUS - A Quick and Dirty Usability Scale. *Usability Evalution in Industry,* 189(194), pp. 4-7.

Coleman, D., Ash, D., Lowther, B. & Oman, P., 1994. Using Metrics to Evaluate Software System Maintainability. *Computer,* 27(8), pp. 44-49.

Fahmy, S., Haslinda, N., Roslina, W. & Fariha, Z., 2012. Evaluating the Quality of Software in e-Book Using the ISO 9126 Model. *International Journal of Control and Automation,* 5(2), pp. 115-122.

International Standards Office, 2000. *ISO/IEC FDIS 9126-1: Quality Model,* Geneva: International Organization for Standardization / International Electrotechnical Commission.

International Standards Office, 2002. *ISO/IEC TR 9126-2: External Metrics,* Geneva: International Organization for Standardization / International Electrotechnical Commission.

Jamwal, D., 2010. Analysis of Software Quality Models for Organizations. *International Journal of Latest Trends in Computing,* 1(2), pp. 19-23.

Laube, J., 2015. *How to Turn Your Good Restaurant into a Great Business.* [Online]
Available at: http://www.restaurantowner.com/public/How-to-Turn-Your-Good-Restauran-tinto-a-Great-Business.cfm
[Accessed 10 Maret 2015].

Ngai, E., Suk, F. & Lo, S., 2008. Development of an RFID-based Sushi Management System. *International Journal of Production Economics,* 112(2), pp. 630-645.

Nielsen, J., 1993. *Usability Engineering.* San Diego: Academic Press.

Orr, E. & Zadik, Y., 2013. *Programming with CodeIgniter MVC: Build Feature-rich Web Applications using The CodeIgniter MVC Framework.* Birmingham: PACKT Publishing.

Perdananugraha, G. M., 2010. Rancang Bangun dan Implementasi Sistem Informasi Terintegrasi (Integrated Information System) E-Government Berbasis Open Source. *Teknologi Indonesia,* 33(1), pp. 45-59.

Pieskä, S., Liuska, M., Jauhiainen, J. & Auno, A., 2013. *Intelligent Restaurant System Smartmenu.* Budapest, IEEE.

Rina & Tyagi, S., 2013. A Comparative Study of Performance Testing Tools. *International Journal of Advanced Research in Computer Science and Software Engineering,* 3(5), pp. 1300-1307.

Salonen, V., 2012. *Automatic Portability Testing. Master's Thesis.,* Jyväskylä: University Jyväskylä.

Saputra, S. T. & Harjanti, D., 2013. Pengelolaan dan Pengembangan Usaha pada Restoran Pondok Kemangi di Banjarmasin. *AGORA,* 1(1).

Setialana, P., 2014. Pengembangan dan Analisis Kualitas Sistem Informasi Jadwal Akademik Berbasis YII Framework. *Jurnal Pendidikan Teknik Informatika,* 3(7), pp. 1-10.

Shimmura, T., Takemaka, T. & Akamatsu, M., 2009. *Real-Time Process Management System in a Restaurant by Sharing Food Order Information.* Malacca, IEEE, pp. 703-706.

Sigala, M., 2003. Unravelling The Impact of Information and Communication Technologies (ICT) on Restaurant Productivity. *ECIS 2002 Proceedings,* p. 164.

Sukamto, R. A. & Shalahuddin, M., 2013. *Rekayasa Perangkat Lunak: Terstruktur dan Berorientasi Objek.* Bandung: Informatika.

Taiwar, G., 2013. *Integration Testing Tips.* [Online]
Available at: http://university.utest.com/integration-testing-tips/
[Accessed 23 Maret 2015].

Tan, T.-H. & Chang, C.-S., 2010. Development and Evaluation of an RFID-based e-Restaurant System for Customer-centric Service. *Expert Systems with Applications,* 37(9), pp. 6482-6492.

Tullis, T. S. & Stetson, J. N., 2004. A Comparison of Questionnaire for Assessing Website Usability. *Usability Proffessional Association Conference,* pp. 1-12.

Wahono, R. S., 2006. *Teknik Pengukuran Kualitas Perangkat Lunak.* [Online] Available at: http://romisatriawahono.net/2006/06/05/teknik-pengukuran-kualitas-perangkat-lunak/ [Accessed 5 Maret 2015].

LAMPIRAN

Yogurt & Mie Jogging

Jl.SoekarnoHatta 24 Malang
Tanggal : 09-03-2015
Jam : 14:56:48
Pesanan : anhar

No	Kode	Nama	Jumlah
1	MK03	Mie Banget	1.00
cb 8			
2	SUM11	Bola Udang Keju	1.00

Lampiran 1 Lembar Pesanan Restoran Mie Jogging Cabang Soekarno Hatta

Yogurt & Mie Jogging

Jl.SoekarnoHatta 24 Malang
No. : KSR03152357 09-03-2015
Kasir : Desy 14:57:03
Meja : SEMUA MEJA
Tamu :
Member: UMUM

Mie Banget			
1.00 x 13,000.00		0.00=	13,000.00
Bola Udang Keju			
1.00 x 12,000.00		0.00=	12,000.00
Sub total	:		25,000.00
Pajak	:		0.00
Diskon	:		0.00
Diskon Member	:		0.00
Deposit	:		0.00
Sisa	:		0.00
Grand total	:		25,000.00
Tunai	:		50,000.00
Kartu kredit	:		0.00
Kembalian	:		25,000.00

Terima Kasih

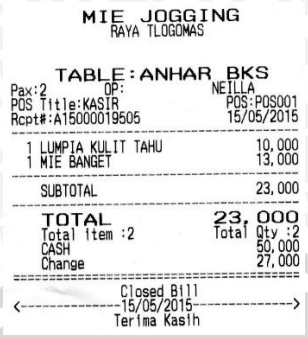
Lampiran 2 Struk Pembayaran Restoran Mie Jogging Cabang Soekarno Hatta

NE IN
ID : CHEN
BLE: ANHAR BKS

1 MIE BANGET
++BUNGKUS
++CABE 12
1 LUMPIA KULIT TAHU
++BUNGKUS

vers: 2
SIR OP: NEILLA
/05/2015 17:11
5000019505

Lampiran 3 Lembar Pesanan Restoran Mie Jogging Cabang Tlogomas



MIE JOGGING
RAYA TLOGOMAS

TABLE:ANHAR BKS
Pax:2 OP: NEILLA
POS Title:KASIR POS:POS001
Rcpt#:A15000019505 15/05/2015

1 LUMPIA KULIT TAHU	10,000
1 MIE BANGET	13,000
SUBTOTAL	23,000
TOTAL	23,000
Total item :2	Total Qty :2
CASH	50,000
Change	27,000

Closed Bill
15/05/2015
Terima Kasih

Lampiran 4 Struk Pembayaran Restoran Mie Jogging Cabang Tlogomas

Follow with us @koberbar

Kober Mie Setan

Cash Payment

Order No 88
Customer ANHAR
08571567XXXX
Lokasi Bawah

Menu	Qty	Price	Amount
Mie Iblis S (B)	1	9.500	9.500
Mie Cabe 5 (B)	1	8.500	8.500
Total			18,000
Tax 10%			1,800
			19,800
Discount			0
Grand Total			19.800
Cash			20.000
Cash Back			200

Thank You
MEMEY 27/02/2015 17:01:09

Lampiran 5 Struk Pembayaran Restoran Mie Setan

NOODLE INC.
Jl. Soekarno Hatta DR 1-2
Malang

Tanggal : 20-03-15
Jam : 18:51:41
Pelayan : Default
Kasir : hana
No.Meja : A-22

1 MIE TARIK SAPI	21,000
1 MIE TARIK TOFU	21,000
1 Mie Tarik Urat	21,000
1 BANANA COFFEE CREAM	17,000
1 HOT LEMON TEA	10,000
1 HOT ORANGE	10,000
1 LEMON SQUASH	15,000
1 UDANG MAYONAISE	26,000
Sub Total :	141,000
Total Bill :	141,000

Grand Total : 141,000

"TERIMA KASIH"
Kami Juga Melayani Delivery
Pesan ke 416919

200315117

Lampiran 6 Struk Pembayaran Restoran Noodle Inc

"Griya Bebek dan Ayam"
enaknya bikin nelongso
Jl.BendunganSutami No.3A MALANG
(0341) 5023603

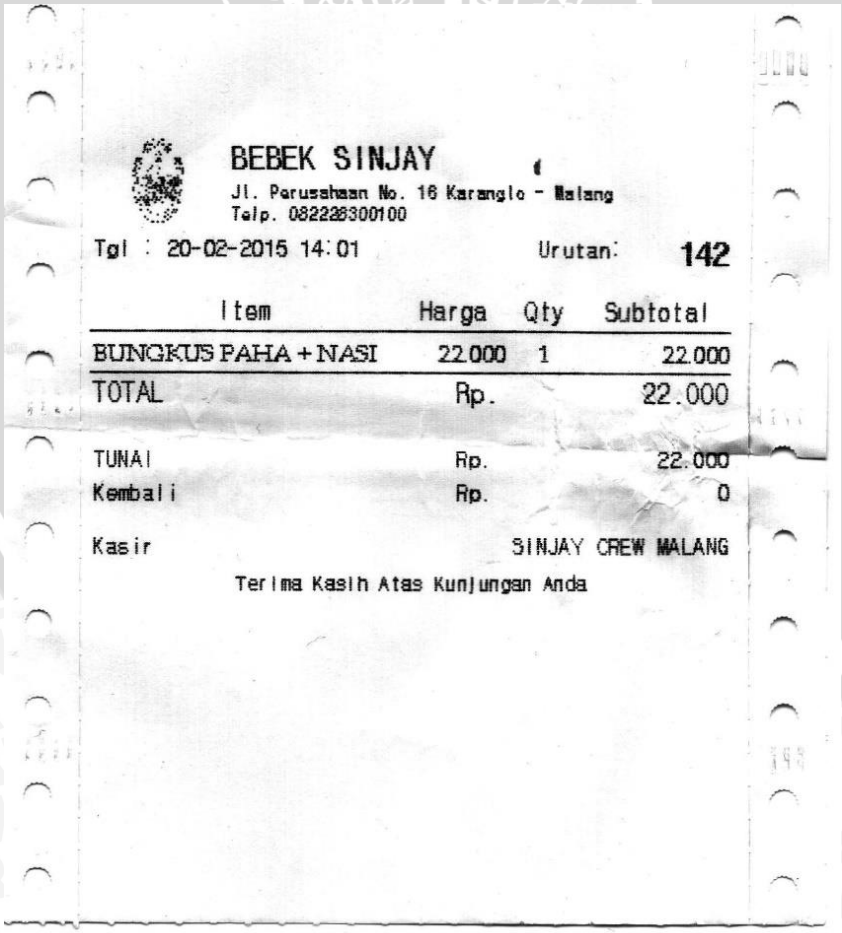
C032015-002099 05/03/2015

ES SODA GEMBIRA
1 7000 0,00% 7.000,00
AYAM GORENG KEPRUK
1 8000 0,00% 8.000,00

selvi2 Sub.TOTAL 15.000,00
#2,00 Diskon 0,00
22:14:10 TOTAL 15.000,00

Bayar 15.000,00
Kembali 0,00

Lampiran 7 Struk Pembayaran Restoran Griya Bebek dan Ayam Nelongso



BEBEK SINJAY
Jl. Perusahaan No. 16 Karanglo - Malang
Telp. 082228300100

Tgl : 20-02-2015 14:01 Urutan: 142

Item	Harga	Qty	Subtotal
BUNGKUS PAHA + NASI	22.000	1	22.000
TOTAL	Rp.		22.000
TUNAI	Rp.		22.000
Kembali	Rp.		0

Kasir SINJAY CREW MALANG

Terima Kasih Atas Kunjungan Anda

Lampiran 8 Struk Pembayaran Restoran Bebek Sinjay

KUESIONER

Evaluasi *Usability* Sistem Pengelolaan Restoran Terintegrasi (OtseResto) dengan Model Kuesioner SUS (*System Usability Scale*).

Nama : _____

Umur : _____

Pekerjaan : _____

Berilah tanda (X) pada lembar kuesioner yang Anda anggap sesuai!

STS(Sangat Tidak Setuju), TS(Tidak Setuju), N(Netral), S(Setuju), SS(Sangat Setuju)

No	Pertanyaan	Jawaban				
		STS	TS	N	S	SS
1	Saya berpikir bahwa saya akan sering menggunakan sistem ini					
2	Saya berpikir bahwa sistem ini terlalu kompleks					
3	Saya berpikir bahwa sistem ini mudah untuk digunakan					
4	Saya berpikir bahwa saya membutuhkan bantuan tenaga ahli untuk mengoperasikan sistem ini					
5	Saya menemukan berbagai fungsi dalam sistem ini terintegrasi dengan baik					
6	Saya berpikir banyak ketidakkonsistenan dalam sistem ini					
7	Saya membayangkan bahwa banyak orang akan dengan cepat mengerti tentang penggunaan sistem ini					
8	Saya merasakan bahwa sistem ini rumit (tidak praktis) untuk digunakan					
9	Saya merasa sangat percaya diri ketika menggunakan sistem ini					
10	Saya perlu belajar banyak hal sebelum mengoperasikan sistem ini					

Lampiran 9 Model Kuesioner SUS untuk Pengujian *Usability*

Summary

Profile	Successful sessions	Failed sessions	Successful pages	Failed pages	Successful hits	Failed hits	Total KBytes sent	Total KBytes received	Avg Response time, sec (with page elements)
otseresto	300	0	8700	0	8700	0	2727	73785	0.78(0.78)

Number of active users

Profile	0:00:00-0:00:10	0:00:10-0:00:20	0:00:20-0:00:30	0:00:30-0:00:38	0:00:38-0:00:48	0:00:48-0:00:58	0:00:58-0:01:06	0:01:06-0:01:16	0:01:16-0:01:26	0:01:26-0:01:36
otseresto	80	80	80	80	80	80	80	80	60	50
Total	80	80	80	80	80	80	80	80	60	50

Successful sessions (Failed sessions)

Profile	0:00:00-0:00:10	0:00:10-0:00:20	0:00:20-0:00:30	0:00:30-0:00:38	0:00:38-0:00:48	0:00:48-0:00:58	0:00:58-0:01:06	0:01:06-0:01:16	0:01:16-0:01:26	0:01:26-0:01:36	Total
otseresto	0(0)	21(0)	59(0)	10(0)	56(0)	15(0)	21(0)	58(0)	10(0)	50(0)	300(0)
Total	0(0)	21(0)	59(0)	10(0)	56(0)	15(0)	21(0)	58(0)	10(0)	50(0)	300(0)

Successful pages (Failed pages)

Profile	0:00:00-0:00:10	0:00:10-0:00:20	0:00:20-0:00:30	0:00:30-0:00:38	0:00:38-0:00:48	0:00:48-0:00:58	0:00:58-0:01:06	0:01:06-0:01:16	0:01:16-0:01:26	0:01:26-0:01:36	Total
otseresto	1040(0)	1064(0)	1060(0)	826(0)	1008(0)	962(0)	784(0)	731(0)	793(0)	432(0)	8700(0)
Total	1040(0)	1064(0)	1060(0)	826(0)	1008(0)	962(0)	784(0)	731(0)	793(0)	432(0)	8700(0)

Successful hits (Failed hits)

Profile	0:00:00-0:00:10	0:00:10-0:00:20	0:00:20-0:00:30	0:00:30-0:00:38	0:00:38-0:00:48	0:00:48-0:00:58	0:00:58-0:01:06	0:01:06-0:01:16	0:01:16-0:01:26	0:01:26-0:01:36	Total
otseresto	1040(0)	1064(0)	1060(0)	826(0)	1008(0)	962(0)	784(0)	731(0)	793(0)	432(0)	8700(0)
Total	1040(0)	1064(0)	1060(0)	826(0)	1008(0)	962(0)	784(0)	731(0)	793(0)	432(0)	8700(0)

Report generated at 20/05/2015 14:08:58 by WAPT Pro 3.1 © SoftLogica 2015

Lampiran 10 Hasil Pengujian *Reliability* dengan WAPT

Response time, sec (with page elements)

Name	Time	0:00:00-0:00:10	0:00:10-0:00:20	0:00:20-0:00:30	0:00:30-0:00:38	0:00:38-0:00:48	0:00:48-0:00:58	0:00:58-0:01:06	0:01:06-0:01:16	0:01:16-0:01:26	0:01:26-0:01:36	Total
otseresto.Autentikasi_index		80	21	59	8	53	20	20	39	0	0	300
	Avg	0.22(0.22)	0.36(0.36)	0.35(0.35)	0.40(0.40)	0.40(0.40)	0.39(0.39)	0.40(0.40)	0.41(0.41)	-	-	0.34(0.34)
otseresto.Autentikasi_login		80	17	63	5	44	31	18	42	0	0	300
	Avg	0.66(0.66)	1.09(1.09)	1.09(1.09)	1.12(1.12)	1.16(1.16)	1.19(1.19)	1.20(1.20)	1.34(1.34)	-	-	1.04(1.04)
otseresto.Autentikasi_logout		80	17	63	2	45	33	17	43	0	0	300
	Avg	0.45(0.45)	0.72(0.72)	0.72(0.72)	0.71(0.71)	0.76(0.76)	0.77(0.77)	0.78(0.78)	1.11(1.11)	-	-	0.72(0.72)
otseresto.KelolaPengguna_index		80	13	67	1	37	42	15	37	8	0	300
	Avg	0.73(0.73)	1.05(1.05)	1.08(1.08)	1.15(1.15)	1.11(1.11)	1.17(1.17)	1.18(1.18)	1.55(1.55)	1.68(1.68)	-	1.08(1.08)
otseresto.KelolaPengguna_membuatPengguna		80	12	65	3	33	47	9	37	14	0	300
	Avg	0.53(0.53)	0.74(0.74)	0.71(0.71)	0.73(0.73)	0.73(0.73)	0.76(0.76)	0.71(0.71)	1.11(1.11)	0.99(0.99)	-	0.74(0.74)
otseresto.KelolaPengguna_melihatPengguna		80	8	59	13	29	51	8	33	19	0	300
	Avg	0.55(0.55)	0.69(0.69)	0.72(0.72)	0.71(0.71)	0.77(0.77)	0.81(0.81)	0.86(0.86)	1.02(1.02)	0.83(0.83)	-	0.74(0.74)

otseresto.KelolaPengguna_mengubahPengguna		80	4	49	27	22	58	5	29	26	0	300
	Avg	0.61(0.61)	0.77(0.77)	0.74(0.74)	0.74(0.74)	0.80(0.80)	0.80(0.80)	0.77(0.77)	1.20(1.20)	0.87(0.87)	-	0.78(0.78)
otseresto.KelolaPengguna_menghapusPengguna		77	6	47	30	19	61	3	27	30	0	300
	Avg	0.61(0.61)	0.70(0.70)	0.74(0.74)	0.77(0.77)	0.80(0.80)	0.82(0.82)	0.81(0.81)	1.13(1.13)	0.78(0.78)	-	0.77(0.77)
otseresto.KelolaMenuRestoran_index		60	20	46	34	18	62	2	21	37	0	300
	Avg	0.57(0.57)	0.67(0.67)	0.73(0.73)	0.75(0.75)	0.77(0.77)	0.78(0.78)	0.76(0.76)	1.04(1.04)	0.72(0.72)	-	0.73(0.73)
otseresto.KelolaMenuRestoran_membuatMenuRestoran		50	30	42	38	17	62	2	19	40	0	300
	Avg	0.61(0.61)	0.72(0.72)	0.72(0.72)	0.70(0.70)	0.78(0.78)	0.78(0.78)	0.71(0.71)	0.92(0.92)	0.71(0.71)	-	0.73(0.73)
otseresto.KelolaMenuRestoran_melihatMenuRestoran		48	32	37	43	15	59	7	17	42	0	300
	Avg	0.61(0.61)	0.70(0.70)	0.72(0.72)	0.76(0.76)	0.76(0.76)	0.81(0.81)	0.80(0.80)	0.95(0.95)	0.66(0.66)	-	0.73(0.73)
otseresto.KelolaMenuRestoran_mengubahMenuRestoran		44	36	29	51	12	56	13	16	43	0	300
	Avg	0.62(0.62)	0.75(0.75)	0.74(0.74)	0.76(0.76)	0.80(0.80)	0.80(0.80)	0.75(0.75)	1.06(1.06)	0.67(0.67)	-	0.75(0.75)
otseresto.KelolaMenuRestoran_menghapusMenuRestoran		37	43	22	58	8	51	21	15	45	0	300

	Avg	0.65(0.65)	0.74(0.74)	0.75(0.75)	0.75(0.75)	0.75(0.75)	0.80(0.80)	0.76(0.76)	1.24(1.24)	0.65(0.65)	-	0.76(0.76)
otseresto.KelolaInformasiRestoran_index		31	49	21	59	6	43	31	12	48	0	300
	Avg	1.01(1.01)	1.10(1.10)	1.08(1.08)	1.08(1.08)	1.08(1.08)	1.16(1.16)	1.17(1.17)	1.59(1.59)	1.01(1.01)	-	1.11(1.11)
otseresto.KelolaInformasiRestoran_mengubahInformasiRestoran		23	57	18	61	4	41	36	9	51	0	300
	Avg	0.66(0.66)	0.71(0.71)	0.70(0.70)	0.75(0.75)	0.70(0.70)	0.77(0.77)	0.76(0.76)	1.29(1.29)	0.71(0.71)	-	0.75(0.75)
otseresto.KelolaPemesanan_index		21	59	17	52	12	36	43	8	51	1	300
	Avg	0.70(0.70)	0.74(0.74)	0.73(0.73)	0.75(0.75)	0.78(0.78)	0.81(0.81)	0.80(0.80)	1.20(1.20)	0.67(0.67)	1.29(1.29)	0.76(0.76)
otseresto.KelolaPemesanan_membuatPemesananMenu		19	61	15	40	26	32	47	4	50	6	300
	Avg	0.70(0.70)	0.73(0.73)	0.71(0.71)	0.74(0.74)	0.80(0.80)	0.78(0.78)	0.82(0.82)	1.29(1.29)	0.71(0.71)	1.24(1.24)	0.77(0.77)
otseresto.KelolaPemesanan_melihatStatusPemesananMenu		17	63	12	38	30	30	50	2	45	13	300
	Avg	0.69(0.69)	0.71(0.71)	0.73(0.73)	0.73(0.73)	0.75(0.75)	0.75(0.75)	0.78(0.78)	1.37(1.37)	0.70(0.70)	1.08(1.08)	0.75(0.75)
otseresto.KelolaDapur_index		16	64	9	39	32	21	59	2	40	18	300
	Avg	0.68(0.68)	0.72(0.72)	0.73(0.73)	0.76(0.76)	0.76(0.76)	0.78(0.78)	0.80(0.80)	0.76(0.76)	0.67(0.67)	0.95(0.95)	0.76(0.76)

otseresto.KelolaDapur_melihatDaftarPesanan		14	66	6	37	37	19	59	3	36	23	300
	Avg	0.66(0.66)	0.72(0.72)	0.71(0.71)	0.76(0.76)	0.77(0.77)	0.79(0.79)	0.80(0.80)	1.16(1.16)	0.69(0.69)	0.93(0.93)	0.77(0.77)
otseresto.KelolaDapur_memberikanNotifikasiPesanan		11	67	4	37	41	18	53	10	31	28	300
	Avg	0.65(0.65)	0.74(0.74)	0.76(0.76)	0.75(0.75)	0.82(0.82)	0.80(0.80)	0.80(0.80)	0.90(0.90)	0.68(0.68)	0.78(0.78)	0.77(0.77)
otseresto.KelolaDapur_memvalidasiPesanan		6	58	17	31	48	17	47	16	26	34	300
	Avg	0.74(0.74)	0.74(0.74)	0.79(0.79)	0.77(0.77)	0.79(0.79)	0.77(0.77)	0.82(0.82)	0.83(0.83)	0.70(0.70)	0.89(0.89)	0.79(0.79)
otseresto.KelolaDapur_mencariPesanan		4	47	29	23	57	16	40	24	21	39	300
	Avg	0.69(0.69)	0.72(0.72)	0.69(0.69)	0.73(0.73)	0.76(0.76)	0.83(0.83)	0.78(0.78)	0.79(0.79)	0.73(0.73)	0.84(0.84)	0.76(0.76)
otseresto.KelolaDapur_mencetakPesanan		1	48	31	21	59	13	36	31	19	41	300
	Avg	0.72(0.72)	0.72(0.72)	0.73(0.73)	0.78(0.78)	0.79(0.79)	0.79(0.79)	0.81(0.81)	0.83(0.83)	0.73(0.73)	0.83(0.83)	0.78(0.78)
otseresto.KelolaPembayaran_index		1	43	36	18	62	8	37	35	18	42	300
	Avg	0.69(0.69)	0.72(0.72)	0.73(0.73)	0.72(0.72)	0.75(0.75)	0.84(0.84)	0.74(0.74)	0.80(0.80)	0.67(0.67)	0.80(0.80)	0.75(0.75)
otseresto.KelolaPembayaran_membuatTagihanPembayaran		0	40	40	17	63	7	31	42	18	42	300

	Avg	-	0.74(0.74)	0.76(0.76)	0.74(0.74)	0.79(0.79)	0.81(0.81)	0.81(0.81)	0.83(0.83)	0.65(0.65)	0.73(0.73)	0.77(0.77)
otseresto.KelolaPembayaran_melihatTagihanPembayaran		0	35	45	17	63	5	28	47	14	46	300
	Avg	-	0.71(0.71)	0.70(0.70)	0.78(0.78)	0.79(0.79)	0.73(0.73)	0.78(0.78)	0.80(0.80)	0.69(0.69)	0.88(0.88)	0.78(0.78)
otseresto.KelolaPembayaran_mencetakTagihanPembayaran		0	27	53	13	60	8	26	53	11	49	300
	Avg	-	0.75(0.75)	0.74(0.74)	0.76(0.76)	0.81(0.81)	0.85(0.85)	0.80(0.80)	0.93(0.93)	0.70(0.70)	0.83(0.83)	0.81(0.81)
otseresto.KelolaPembayaran_melihatHistoriTransaksi		0	21	59	10	56	15	21	58	10	50	300
	Avg	-	0.72(0.72)	0.73(0.73)	0.69(0.69)	0.75(0.75)	0.81(0.81)	0.78(0.78)	0.98(0.98)	0.86(0.86)	0.72(0.72)	0.79(0.79)

Report generated at 20/05/2015 14:18:06 by WAPT Pro 3.1 © SoftLogica 2015

Lampiran 11 Hasil Pengujian *Efficiency* dengan WAPT

Source Lines	Code Lines	Comment Lines	Blank Lines	Halstead Volume	Filename
51	42	2	7	948,918	C:/Users/Anhar Tribowo/Desktop/maintainability/test2/source/autentikasi.php
50	41	2	7	1363,511	C:/Users/Anhar Tribowo/Desktop/maintainability/test2/source/dapur.php
24	18	2	4	559,092	C:/Users/Anhar Tribowo/Desktop/maintainability/test2/source/detailBayar.php
25	18	2	5	559,092	C:/Users/Anhar Tribowo/Desktop/maintainability/test2/source/detailPesan.php
130	111	2	17	3392,859	C:/Users/Anhar Tribowo/Desktop/maintainability/test2/source/kelolaDapur.php
36	29	2	5	540,484	C:/Users/Anhar Tribowo/Desktop/maintainability/test2/source/kelolaInformasiRestoran.php
77	65	2	10	1765,234	C:/Users/Anhar Tribowo/Desktop/maintainability/test2/source/kelolaMenuRestoran.php
124	113	2	9	3911,732	C:/Users/Anhar Tribowo/Desktop/maintainability/test2/source/kelolaPembayaran.php
71	62	2	7	1718,209	C:/Users/Anhar Tribowo/Desktop/maintainability/test2/source/kelolaPemesanan.php
81	70	2	9	1832,468	C:/Users/Anhar Tribowo/Desktop/maintainability/test2/source/kelolaPengguna.php
74	56	2	16	2061,612	C:/Users/Anhar Tribowo/Desktop/maintainability/test2/source/menuRestoran.php
25	16	2	7	371,385	C:/Users/Anhar Tribowo/Desktop/maintainability/test2/source/modAutentikasi.php
113	94	2	16	4262,312	C:/Users/Anhar Tribowo/Desktop/maintainability/test2/source/pembayaran.php
71	60	2	9	2700,053	C:/Users/Anhar Tribowo/Desktop/maintainability/test2/source/pemesanan.php
67	52	2	13	1522,384	C:/Users/Anhar Tribowo/Desktop/maintainability/test2/source/pengguna.php
37	28	2	7	921,519	C:/Users/Anhar Tribowo/Desktop/maintainability/test2/source/restoran.php

Lampiran 12 Hasil Pengujian *Maintainability* dengan DMSSearchEngine

Tested:	http://localhost/otseresto	Resolution:	1366x728

No errors found		
Internet Explorer 11	0 Errors, 0 warning(s)	Jump to Result
Firefox 33	0 Errors, 0 warning(s)	Jump to Result
Firefox 32	0 Errors, 0 warning(s)	Jump to Result
Safari 7	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 39	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 38	0 Errors, 0 warning(s)	Jump to Result
Opera 25	0 Errors, 0 warning(s)	Jump to Result
iPad 4 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPad Mini (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPhone 5 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
Android 4.3	0 Errors, 0 warning(s)	Jump to Result
Android 4.4 Tablet	0 Errors, 0 warning(s)	Jump to Result

Lampiran 13 Hasil Pengujian *Portability* Halaman Login

Tested:	http://localhost/otseresto/kelolaPemesanan	Resolution:	1366x728

No errors found		
Internet Explorer 11	0 Errors, 0 warning(s)	Jump to Result
Firefox 33	0 Errors, 0 warning(s)	Jump to Result
Firefox 32	0 Errors, 0 warning(s)	Jump to Result
Safari 7	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 39	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 38	0 Errors, 0 warning(s)	Jump to Result
Opera 25	0 Errors, 0 warning(s)	Jump to Result
iPad 4 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPad Mini (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPhone 5 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
Android 4.3	0 Errors, 0 warning(s)	Jump to Result
Android 4.4 Tablet	0 Errors, 0 warning(s)	Jump to Result

Lampiran 14 Hasil Pengujian *Portability* Halaman Utama Pelayan

Tested:	http://localhost/otseresto/kelolaPemesanan/membuatPemesananMenu	Resolution:	1366x728

No errors found		
Internet Explorer 11	0 Errors, 0 warning(s)	Jump to Result
Firefox 33	0 Errors, 0 warning(s)	Jump to Result
Firefox 32	0 Errors, 0 warning(s)	Jump to Result
Safari 7	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 39	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 38	0 Errors, 0 warning(s)	Jump to Result
Opera 25	0 Errors, 0 warning(s)	Jump to Result
iPad 4 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPad Mini (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPhone 5 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
Android 4.3	0 Errors, 0 warning(s)	Jump to Result
Android 4.4 Tablet	0 Errors, 0 warning(s)	Jump to Result

Lampiran 15 Hasil Pengujian *Portability* Halaman Buat Pemesanan

Tested:	http://localhost/otseresto/kelolaPemesanan/melihatStatusPemesananMenu	Resolution:	1366x728

No errors found		
Internet Explorer 11	0 Errors, 0 warning(s)	Jump to Result
Firefox 33	0 Errors, 0 warning(s)	Jump to Result
Firefox 32	0 Errors, 0 warning(s)	Jump to Result
Safari 7	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 39	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 38	0 Errors, 0 warning(s)	Jump to Result
Opera 25	0 Errors, 0 warning(s)	Jump to Result
iPad 4 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPad Mini (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPhone 5 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
Android 4.3	0 Errors, 0 warning(s)	Jump to Result
Android 4.4 Tablet	0 Errors, 0 warning(s)	Jump to Result

Lampiran 16 Hasil Pengujian *Portability* Lihat Status Pemesanan

Tested:	http://localhost/otseresto/kelolaPembayaran	Resolution:	1366x728
No errors found			
Internet Explorer 11	0 Errors, 0 warning(s)	Jump to Result	
Firefox 33	0 Errors, 0 warning(s)	Jump to Result	
Firefox 32	0 Errors, 0 warning(s)	Jump to Result	
Safari 7	0 Errors, 0 warning(s)	Jump to Result	
Google Chrome 39	0 Errors, 0 warning(s)	Jump to Result	
Google Chrome 38	0 Errors, 0 warning(s)	Jump to Result	
Opera 25	0 Errors, 0 warning(s)	Jump to Result	
iPad 4 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result	
iPad Mini (iOS 8)	0 Errors, 0 warning(s)	Jump to Result	
iPhone 5 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result	
Android 4.3	0 Errors, 0 warning(s)	Jump to Result	
Android 4.4 Tablet	0 Errors, 0 warning(s)	Jump to Result	

Lampiran 17 Hasil Pengujian *Portability* Halaman Utama Kasir

Tested:	http://localhost/otseresto/kelolaPembayaran/membuatTagihanPembayaran/PSN05201500017	Resolution:	1366x728
No errors found			
Internet Explorer 11	0 Errors, 0 warning(s)	Jump to Result	
Firefox 33	0 Errors, 0 warning(s)	Jump to Result	
Firefox 32	0 Errors, 0 warning(s)	Jump to Result	
Safari 7	0 Errors, 0 warning(s)	Jump to Result	
Google Chrome 39	0 Errors, 0 warning(s)	Jump to Result	
Google Chrome 38	0 Errors, 0 warning(s)	Jump to Result	
Opera 25	0 Errors, 0 warning(s)	Jump to Result	
iPad 4 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result	
iPad Mini (iOS 8)	0 Errors, 0 warning(s)	Jump to Result	
iPhone 5 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result	
Android 4.3	0 Errors, 0 warning(s)	Jump to Result	
Android 4.4 Tablet	0 Errors, 0 warning(s)	Jump to Result	

Lampiran 18 Hasil Pengujian *Portability* Halaman Buat Tagihan

Tested:	http://localhost/otseresto/kelolaPembayaran/melihatTagihanPembayaran	Resolution:	1366x728
No errors found			
Internet Explorer 11	0 Errors, 0 warning(s)	Jump to Result	
Firefox 33	0 Errors, 0 warning(s)	Jump to Result	
Firefox 32	0 Errors, 0 warning(s)	Jump to Result	
Safari 7	0 Errors, 0 warning(s)	Jump to Result	
Google Chrome 39	0 Errors, 0 warning(s)	Jump to Result	
Google Chrome 38	0 Errors, 0 warning(s)	Jump to Result	
Opera 25	0 Errors, 0 warning(s)	Jump to Result	
iPad 4 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result	
iPad Mini (iOS 8)	0 Errors, 0 warning(s)	Jump to Result	
iPhone 5 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result	
Android 4.3	0 Errors, 0 warning(s)	Jump to Result	
Android 4.4 Tablet	0 Errors, 0 warning(s)	Jump to Result	

Lampiran 19 Hasil Pengujian *Portability* Halaman Lihat Tagihan

Tested:	http://localhost/otseresto/kelolaDapur	Resolution:	1366x728
No errors found			
Internet Explorer 11	0 Errors, 0 warning(s)	Jump to Result	
Firefox 33	0 Errors, 0 warning(s)	Jump to Result	
Firefox 32	0 Errors, 0 warning(s)	Jump to Result	
Safari 7	0 Errors, 0 warning(s)	Jump to Result	
Google Chrome 39	0 Errors, 0 warning(s)	Jump to Result	
Google Chrome 38	0 Errors, 0 warning(s)	Jump to Result	
Opera 25	0 Errors, 0 warning(s)	Jump to Result	
iPad 4 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result	
iPad Mini (iOS 8)	0 Errors, 0 warning(s)	Jump to Result	
iPhone 5 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result	
Android 4.3	0 Errors, 0 warning(s)	Jump to Result	
Android 4.4 Tablet	0 Errors, 0 warning(s)	Jump to Result	

Lampiran 20 Hasil Pengujian *Portability* Halaman Utama Koki

Tested:	http://localhost/otseresto/kelolaDapur/melihatDaftarPesanan	Resolution:	1366x728

No errors found		
Internet Explorer 11	0 Errors, 0 warning(s)	Jump to Result
Firefox 33	0 Errors, 0 warning(s)	Jump to Result
Firefox 32	0 Errors, 0 warning(s)	Jump to Result
Safari 7	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 39	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 38	0 Errors, 0 warning(s)	Jump to Result
Opera 25	0 Errors, 0 warning(s)	Jump to Result
iPad 4 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPad Mini (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPhone 5 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
Android 4.3	0 Errors, 0 warning(s)	Jump to Result
Android 4.4 Tablet	0 Errors, 0 warning(s)	Jump to Result

Lampiran 21 Hasil Pengujian *Portability* Halaman Lihat Daftar Pesanan

Tested:	http://localhost/otseresto/kelolaDapur/mencariPesanan	Resolution:	1366x728

No errors found		
Internet Explorer 11	0 Errors, 0 warning(s)	Jump to Result
Firefox 33	0 Errors, 0 warning(s)	Jump to Result
Firefox 32	0 Errors, 0 warning(s)	Jump to Result
Safari 7	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 39	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 38	0 Errors, 0 warning(s)	Jump to Result
Opera 25	0 Errors, 0 warning(s)	Jump to Result
iPad 4 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPad Mini (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPhone 5 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
Android 4.3	0 Errors, 0 warning(s)	Jump to Result
Android 4.4 Tablet	0 Errors, 0 warning(s)	Jump to Result

Lampiran 22 Hasil Pengujian *Portability* Halaman Cari Pesanan

Tested:	http://localhost/otseresto/kelolaMenuRestoran	Resolution:	1366x728

No errors found		
Internet Explorer 11	0 Errors, 0 warning(s)	Jump to Result
Firefox 33	0 Errors, 0 warning(s)	Jump to Result
Firefox 32	0 Errors, 0 warning(s)	Jump to Result
Safari 7	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 39	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 38	0 Errors, 0 warning(s)	Jump to Result
Opera 25	0 Errors, 0 warning(s)	Jump to Result
iPad 4 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPad Mini (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPhone 5 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
Android 4.3	0 Errors, 0 warning(s)	Jump to Result
Android 4.4 Tablet	0 Errors, 0 warning(s)	Jump to Result

Lampiran 23 Hasil Pengujian *Portability* Halaman Utama Manajer

Tested:	http://localhost/otseresto/kelolaMenuRestoran/membuatMenuRestoran	Resolution:	1366x728

No errors found		
Internet Explorer 11	0 Errors, 0 warning(s)	Jump to Result
Firefox 33	0 Errors, 0 warning(s)	Jump to Result
Firefox 32	0 Errors, 0 warning(s)	Jump to Result
Safari 7	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 39	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 38	0 Errors, 0 warning(s)	Jump to Result
Opera 25	0 Errors, 0 warning(s)	Jump to Result
iPad 4 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPad Mini (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPhone 5 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
Android 4.3	0 Errors, 0 warning(s)	Jump to Result
Android 4.4 Tablet	0 Errors, 0 warning(s)	Jump to Result

Lampiran 24 Hasil Pengujian *Portability* Halaman Tambah Menu Restoran

Tested:	http://localhost/otseresto/kelolaMenuRestoran/melihatMenuRestoran	Resolution:	1366x728

No errors found		
Internet Explorer 11	0 Errors, 0 warning(s)	Jump to Result
Firefox 33	0 Errors, 0 warning(s)	Jump to Result
Firefox 32	0 Errors, 0 warning(s)	Jump to Result
Safari 7	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 39	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 38	0 Errors, 0 warning(s)	Jump to Result
Opera 25	0 Errors, 0 warning(s)	Jump to Result
iPad 4 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPad Mini (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPhone 5 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
Android 4.3	0 Errors, 0 warning(s)	Jump to Result
Android 4.4 Tablet	0 Errors, 0 warning(s)	Jump to Result

Lampiran 25 Hasil Pengujian *Portability* Halaman Lihat Menu Restoran

Tested:	http://localhost/otseresto/kelolaMenuRestoran/mengubahMenuRestoran/MAK10001	Resolution:	1366x728

No errors found		
Internet Explorer 11	0 Errors, 0 warning(s)	Jump to Result
Firefox 33	0 Errors, 0 warning(s)	Jump to Result
Firefox 32	0 Errors, 0 warning(s)	Jump to Result
Safari 7	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 39	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 38	0 Errors, 0 warning(s)	Jump to Result
Opera 25	0 Errors, 0 warning(s)	Jump to Result
iPad 4 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPad Mini (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPhone 5 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
Android 4.3	0 Errors, 0 warning(s)	Jump to Result
Android 4.4 Tablet	0 Errors, 0 warning(s)	Jump to Result

Lampiran 26 Hasil Pengujian *Portability* Halaman Ubah Menu Restoran

Tested:	http://localhost/otseresto/kelolaPengguna/membuatPengguna	Resolution:	1366x728

No errors found		
Internet Explorer 11	0 Errors, 0 warning(s)	Jump to Result
Firefox 33	0 Errors, 0 warning(s)	Jump to Result
Firefox 32	0 Errors, 0 warning(s)	Jump to Result
Safari 7	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 39	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 38	0 Errors, 0 warning(s)	Jump to Result
Opera 25	0 Errors, 0 warning(s)	Jump to Result
iPad 4 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPad Mini (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPhone 5 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
Android 4.3	0 Errors, 0 warning(s)	Jump to Result
Android 4.4 Tablet	0 Errors, 0 warning(s)	Jump to Result

Lampiran 27 Hasil Pengujian *Portability* Halaman Tambah Pengguna

Tested:	http://localhost/otseresto/kelolaPengguna/melihatPengguna	Resolution:	1366x728

No errors found		
Internet Explorer 11	0 Errors, 0 warning(s)	Jump to Result
Firefox 33	0 Errors, 0 warning(s)	Jump to Result
Firefox 32	0 Errors, 0 warning(s)	Jump to Result
Safari 7	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 39	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 38	0 Errors, 0 warning(s)	Jump to Result
Opera 25	0 Errors, 0 warning(s)	Jump to Result
iPad 4 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPad Mini (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPhone 5 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
Android 4.3	0 Errors, 0 warning(s)	Jump to Result
Android 4.4 Tablet	0 Errors, 0 warning(s)	Jump to Result

Lampiran 28 Hasil Pengujian *Portability* Halaman Lihat Pengguna

Tested:	http://localhost/otseresto/kelolaPengguna/mengubahPengguna/tribowoanhar	Resolution: 1366x728
No errors found		
Internet Explorer 11	0 Errors, 0 warning(s)	Jump to Result
Firefox 33	0 Errors, 0 warning(s)	Jump to Result
Firefox 32	0 Errors, 0 warning(s)	Jump to Result
Safari 7	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 39	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 38	0 Errors, 0 warning(s)	Jump to Result
Opera 25	0 Errors, 0 warning(s)	Jump to Result
iPad 4 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPad Mini (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPhone 5 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
Android 4.3	0 Errors, 0 warning(s)	Jump to Result
Android 4.4 Tablet	0 Errors, 0 warning(s)	Jump to Result

Lampiran 29 Hasil Pengujian *Portability* Halaman Ubah Pengguna

Tested:	http://localhost/otseresto/kelolaInformasiRestoran/mengubahInformasiRestoran	Resolution: 1366x728
No errors found		
Internet Explorer 11	0 Errors, 0 warning(s)	Jump to Result
Firefox 33	0 Errors, 0 warning(s)	Jump to Result
Firefox 32	0 Errors, 0 warning(s)	Jump to Result
Safari 7	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 39	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 38	0 Errors, 0 warning(s)	Jump to Result
Opera 25	0 Errors, 0 warning(s)	Jump to Result
iPad 4 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPad Mini (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPhone 5 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
Android 4.3	0 Errors, 0 warning(s)	Jump to Result
Android 4.4 Tablet	0 Errors, 0 warning(s)	Jump to Result

Lampiran 30 Hasil Pengujian *Portability* Halaman Ubah Informasi Restoran

Tested:	http://localhost/otseresto/kelolaPembayaran/melihatHistoriTransaksi	Resolution: 1366x728
No errors found		
Internet Explorer 11	0 Errors, 0 warning(s)	Jump to Result
Firefox 33	0 Errors, 0 warning(s)	Jump to Result
Firefox 32	0 Errors, 0 warning(s)	Jump to Result
Safari 7	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 39	0 Errors, 0 warning(s)	Jump to Result
Google Chrome 38	0 Errors, 0 warning(s)	Jump to Result
Opera 25	0 Errors, 0 warning(s)	Jump to Result
iPad 4 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPad Mini (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
iPhone 5 (iOS 8)	0 Errors, 0 warning(s)	Jump to Result
Android 4.3	0 Errors, 0 warning(s)	Jump to Result
Android 4.4 Tablet	0 Errors, 0 warning(s)	Jump to Result

Lampiran 31 Hasil Pengujian *Portability* Halaman Lihat Histori Transaksi