

**PENGEMBANGAN SISTEM *POINT OF SALE* BERBASIS WEB
PADA EDD'S WAFFLE UNTUK Mendukung PENJUALAN
*RESTAURANT***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Said Atharillah Alifka Alhabsyi

NIM: 155150207111008



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019**

PENGESAHAN

PENGEMBANGAN SISTEM *POINT OF SALE* BERBASIS WEB PADA EDDS WAFFLE
UNTUK Mendukung Penjualan *RESTAURANT*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh:
Said Atharillah Alifka Alhabsyi
NIM: 155150207111008

Skripsi ini telah diuji dan dinyatakan lulus pada
12 Maret 2019

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Fajar Pradana, S.ST., M.Eng.
NIP: 19871121 201504 1 004

Dr.Eng. Fitra A. Bachtiar, S.T, M.Eng
NIK: 2012018406281001

Mengetahui
Ketua Jurusan Teknik Informatika

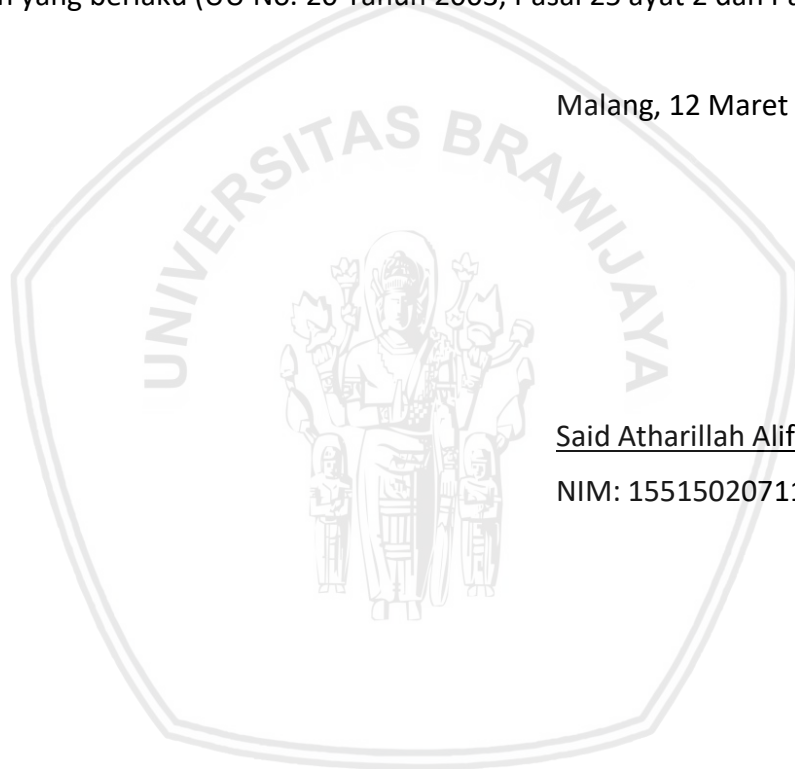
Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

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 referensi.

Apabila ternyata di dalam naskah 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, 12 Maret 2019



Said Atharillah Alifka Alhabsyi

NIM: 155150207111008

PRAKATA

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul “Pengembangan Sistem *Point Of Sale* Berbasis Web Pada Edd’s Waffle Untuk Mendukung Penjualan *Restaurant*” ini dapat terselesaikan.

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. Kedua orang tua dan keluarga besar atas doa dan dukungan yang diberikan selama proses studi hingga saat ini.
2. Bapak Bapak Fajar Pradana, S.ST, M.Eng dan Bapak Dr.Eng. Fitra A. Bachtiar, S.T, M.Eng selaku Pembimbing skripsi yang telah bersedia membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika dan Bapak Agus Wahyu Widodo, S.T, M.Cs. selaku Ketua Program Studi Teknik Informatika.
4. Gledies Putri Irza, B.Bus atas seluruh dukungan selama proses penelitian dan bantuan dalam hal menyediakan data-data yang digunakan pada skripsi ini.
5. Teman-teman keminatan Rekayasa Perangkat Lunak yang sudah bersedia meluangkan waktu untuk berdiskusi dan memberikan dukungan dalam proses pengerjaan skripsi ini.
6. Teman-teman teknik informatika 2015 atas bantuan, motivasi dan memberikan waktu untuk berdiskusi selama ini.
7. Seluruh civitas akademika Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Informatika Universitas Brawijaya dan selama penyelesaian 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, 12 Maret 2019

Penulis
alifka.atharillah@gmail.com

ABSTRAK

Said Atharillah Alifka Alhabsyi, Pengembangan Sistem *Point Of Sale* Berbasis Web Pada Edd's Waffle Untuk Mendukung Penjualan *Restaurant*

Pembimbing: Fajar Pradana, S.ST, M.Eng dan Dr.Eng. Fitra A. Bachtiar, S.T, M.Eng

Bisnis kuliner merupakan bisnis yang tersedia di setiap wilayah baik dalam skala kecil dan besar. Edd's Waffle, merupakan salah satu *restaurant* yang terletak di kota Pekanbaru dengan variasi harga makanan dan minuman yang bervariasi dan digemari oleh remaja. Saat ini, Edd's Waffle menggunakan mesin kasir standar yakni Cash Register Sharp XE-A102 untuk mendukung transaksi penjualan. Dengan mesin kasir standar ini, meninggalkan permasalahan dalam hal perkembangan penjualan dikarenakan kemampuannya yang hanya mampu mencetak resi penjualan dan menyediakan laporan keuangan dalam wadah *X-reports* dan *Z-reports*. Untuk menyelesaikan masalah ini, dikembangkan sebuah sistem *point of sale* yang dapat mendukung Edd's Waffle baik dalam hal penjualan maupun optimalisasi informasi. Sistem *point of sale* ini menyediakan fitur utama seperti menambahkan transaksi, melihat riwayat transaksi penjualan serta grafik perkembangan penjualan, memperkirakan atau memprediksi pendapatan, hingga mengidentifikasi *trend* penjualan yang berisikan daftar-daftar produk dan kombinasi produk terlaris. Dalam mengoptimalkan data-data yang ada, sistem ini memanfaatkan 2 algoritme yang terdapat dalam teknik *data mining*. Algoritme pertama yakni *Holt Exponential Smoothing* untuk memprediksi pendapatan Edd's Waffle. Pemilihan algoritme ini dikarenakan pola pergerakan data yang ada bersifat dinamis yakni cenderung berubah dan tidak terikat variabel tertentu. Sedangkan algoritme kedua yakni *Apriori* yang digunakan untuk menganalisa *trend* pada produk-produk yang terjual sehingga menghasilkan informasi berupa kombinasi produk-produk yang potensial untuk disandingkan. Sistem ini dibangun pada *platform* web. Sistem ini telah diuji dengan menggunakan pengujian unit dan pengujian integrasi yang menggunakan metode *Whitebox* serta pengujian validasi yang menggunakan metode *Blackbox*. Selain itu, dilakukan pengujian usability yang diukur dengan menggunakan metode *System Usability Scale*, skor akhir yang diperoleh ialah 87.5 (*acceptable*).

Kata kunci: *Point Of Sale, Restaurant, Data Mining, Prediksi, Holt Exponential Smoothing, Apriori*

ABSTRACT

Said Atharillah Alifka Alhabsyi, *Web-Based Point of Sale System Development in Edd's Waffle to support restaurant sales.*

Adviser: Fajar Pradana, S.ST, M.Eng and Dr.Eng. Fitra A. Bachtiar, S.T, M.Eng

Culinary business is a business that is available in every region both on a small and large scale. Edd's Waffle, is a restaurant located in Pekanbaru with variant price of food and beverages that become popular among teenagers. Nowadays, Edd's Waffle uses a standard cash register, the Cash Register Sharp XE-A102 to support sales transactions. This standard cash register leaves a lot of problems in terms of sales development due to its ability to only be able to print sales receipts and provide financial reports in X-reports and Z-reports containers. To solve this problem, a point of sale system was developed to support Edd's Waffle both in terms of sales and information. This point of sale system provides key features such as adding transactions, observing sales transaction history and sales progress charts, estimating revenue predictions, and identifying sales trends that contain the lists of products and best-selling product combinations. In term to optimize current data, this system utilizes 2 algorithms that contained in data mining techniques. The first algorithm is Holt Exponential Smoothing to predict Edd's Waffle's income. The selection of this algorithm is because the existing pattern of data movement is dynamic, which tends to change and is not bound by certain variables. Whereas the second algorithm is Apriori, to analyze the trends in every sold product, so that it produces information to get the combination of potential products to be paired. This system is built on web platforms. This system has been tested using unit testing and integration testing using the Whitebox method and validation testing using the Blackbox method. Beside of that, usability testing is measured using the System Usability Scale method, which the final score is 87.5 (acceptable).

Keyword: *Point Of Sale, Restaurant, Data Mining, Prediction, Holt Exponential Smoothing, Apriori*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xix
BAB 1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan.....	3
1.4 Manfaat	4
1.5 Batasan masalah	4
1.6 Sistematika pembahasan	4
BAB 2 LANDASAN KEPUSTAKAAN.....	6
2.1 Kajian Pustaka.....	6
2.2 Rekayasa Perangkat Lunak	7
2.3 <i>Software Development Life Cycle</i>	7
2.3.1 <i>Model Waterfall</i>	8
2.4 Pendekatan Berorientasi Objek.....	9
2.5 Pemodelan Berorientasi Objek.....	9
2.6 Edd's Waffle.....	13
2.7 Teknologi Pengembangan Sistem.....	14
2.7.1 <i>ASP .NET MVC Framework</i>	14
2.7.2 <i>Entity Framework</i>	15
2.7.3 <i>Kendo UI</i>	15
2.8 <i>Point of Sale</i>	16
2.9 <i>Data Mining</i>	16
2.9.1 <i>Holt's Exponential Smoothing</i>	16
2.9.2 <i>Apriori</i>	17



2.10 Pengujian Perangkat Lunak	18
BAB 3 METODOLOGI	22
3.1 Studi Pustaka	22
3.2 Analisis Kebutuhan.....	23
3.3 Perancangan Sistem.....	24
3.4 Implementasi Sistem.....	24
3.5 Pengujian Sistem.....	25
3.6 Pengambilan Kesimpulan dan Saran.....	26
BAB 4 ANALISIS KEBUTUHAN	27
4.1 Gambaran Umum Sistem	27
4.2 Identifikasi Aktor.....	28
4.3 Kebutuhan Fungsional Sistem	29
4.4 Kebutuhan Non-Fungsional.....	37
4.5 Pemodelan Kebutuhan.....	37
4.5.1 Use Case Diagram.....	37
4.5.2 Use Case Scenario.....	39
BAB 5 PERANCANGAN DAN IMPLEMENTASI	76
5.1 Perancangan Sistem.....	76
5.1.1 Perancangan Arsitektur	76
5.1.2 Perancangan Data	82
5.1.3 Perancangan Komponen.....	83
5.1.4 Perancangan Antarmuka	85
5.2 Implementasi Sistem.....	89
5.2.1 Spesifikasi Sistem.....	89
5.2.2 Implementasi Kode Program	90
5.2.3 Implementasi Basis Data.....	92
5.2.4 Implementasi Antarmuka	96
BAB 6 PENGUJIAN.....	99
6.1 Pengujian Unit	99
6.1.1 Pengujian Unit <i>Method InsertTableCart</i>	99
6.1.2 Pengujian Unit <i>Method Prediction_Data</i>	101
6.1.3 Pengujian Unit <i>Method Recommendation_Data</i>	103

6.2 Pengujian Integrasi	105
6.3 Pengujian Validasi	107
6.3.1 Login.....	107
6.3.2 Logout	108
6.3.3 Tambah Transaksi.....	109
6.3.4 Hapus <i>Item</i> Transaksi.....	112
6.3.5 <i>Reset</i> Transaksi	112
6.3.6 Lihat Referensi Produk.....	113
6.3.7 Cari Referensi Produk	113
6.3.8 Lihat Total Transaksi Per Pekan.....	114
6.3.9 Lihat Total Transaksi Per Bulan	115
6.3.10 Lihat Total Pendapatan Transaksi Per Pekan	115
6.3.11 Lihat Total Pendapatan Transaksi Per Bulan.....	116
6.3.12 Lihat Pendapatan Transaksi Tertinggi Per Pekan	116
6.3.13 Lihat Pendapatan Transaksi Tertinggi Per Bulan.....	117
6.3.14 Lihat Pendapatan Transaksi Terendah Per Pekan	118
6.3.15 Lihat Pendapatan Transaksi Terendah Per Bulan.....	118
6.3.16 Lihat Rata-Rata Pendapatan Transaksi Per Pekan.....	119
6.3.17 Lihat Rata-Rata Pendapatan Transaksi Per Bulan	119
6.3.18 Lihat Prediksi Pendapatan.....	120
6.3.19 Lihat 10 Hari Dengan Pendapatan Tertinggi	120
6.3.20 Lihat 10 Hari Dengan Pendapatan Terendah	121
6.3.21 Lihat Riwayat Transaksi.....	122
6.3.22 Lihat Rincian Riwayat Transaksi	122
6.3.23 Hapus Transaksi.....	123
6.3.24 Cari Transaksi	124
6.3.25 Urut Transaksi	124
6.3.26 Lihat Kombinasi Produk	125
6.3.27 Cari Kombinasi Produk.....	126
6.3.28 Urut Kombinasi Produk.....	126
6.3.29 Lihat Jumlah Produk Terjual.....	127
6.3.30 Cari Jumlah Produk Terjual	128



6.3.31 Urut Jumlah Produk Terjual	128
6.3.32 Lihat Daftar Produk.....	129
6.3.33 Cari Produk.....	129
6.3.34 Tambah Produk	130
6.3.35 Hapus Produk	132
6.3.36 Lihat Daftar Kategori.....	133
6.3.37 Cari Kategori.....	133
6.3.38 Tambah Kategori	133
6.3.39 Hapus Kategori	135
6.3.40 Lihat Daftar Akun.....	136
6.3.41 Cari Akun	136
6.3.42 Tambah Akun	136
6.3.43 Hapus Akun	138
6.3.44 Atur Nilai Parameter Prediksi.....	139
6.3.45 Lihat Informasi Prediksi Pendapatan.....	140
6.3.46 Atur Nilai Parameter Produk Potensial.....	140
6.3.47 Lihat Daftar Produk Potensial	142
6.3.48 Cari Produk Potensial.....	143
6.3.49 Urut Produk Potensial.....	143
6.3.50 Lihat Informasi Produk Potensial	144
6.4 Pengujian Usabilitas.....	145
6.4.1 Prosedur Pengujian Usabilitas.....	145
6.4.2 Analisis dan Hasil Pengujian Usabilitas.....	146
BAB 7 PENUTUP	148
7.1 Kesimpulan	148
7.2 Saran.....	148
DAFTAR REFERENSI.....	150

DAFTAR TABEL

Tabel 2.1 Notasi <i>Use Case Diagram</i>	10
Tabel 2.2 Notasi <i>Sequence Diagram</i>	11
Tabel 2.3 Notasi <i>Sequence Diagram</i> (Lanjutan).....	12
Tabel 2.4 Notasi <i>Class Diagram</i>	12
Tabel 2.5 Notasi <i>Class Diagram</i> (Lanjutan)	13
Tabel 2.6 Keterangan skor skala <i>likert</i> tiap pernyataan SUS	20
Tabel 4.1 Identifikasi Aktor	28
Tabel 4.2 Daftar Kebutuhan Fungsional	29
Tabel 4.3 Daftar Kebutuhan Fungsional (Lanjutan).....	30
Tabel 4.4 Daftar Kebutuhan Fungsional (Lanjutan).....	31
Tabel 4.5 Daftar Kebutuhan Fungsional (Lanjutan).....	32
Tabel 4.6 Daftar Kebutuhan Fungsional (Lanjutan).....	33
Tabel 4.7 Daftar Kebutuhan Fungsional (Lanjutan).....	34
Tabel 4.8 Daftar Kebutuhan Fungsional (Lanjutan).....	35
Tabel 4.9 Daftar Kebutuhan Fungsional (Lanjutan).....	36
Tabel 4.10 Daftar Kebutuhan Fungsional (Lanjutan).....	37
Tabel 4.11 Daftar Kebutuhan Non-Fungsional.....	37
Tabel 4.12 <i>Use Case Scenario Login</i>	39
Tabel 4.13 <i>Use Case Scenario Logout</i>	40
Tabel 4.14 <i>Use Case Scenario</i> Tambah Transaksi.....	40
Tabel 4.15 <i>Use Case Scenario</i> Tambah Transaksi (Lanjutan)	41
Tabel 4.16 <i>Use Case Scenario</i> Tambah Transaksi (Lanjutan)	42
Tabel 4.17 <i>Use Case Scenario</i> Tambah Transaksi (Lanjutan)	43
Tabel 4.18 <i>Use Case Scenario</i> Hapus <i>Item</i> Transaksi.....	43
Tabel 4.19 <i>Use Case Scenario</i> <i>Reset</i> Transaksi.....	44
Tabel 4.20 <i>Use Case Scenario</i> Lihat Referensi Produk.....	44
Tabel 4.21 <i>Use Case Scenario</i> Lihat Referensi Produk (Lanjutan).....	45
Tabel 4.22 <i>Use Case Scenario</i> Cari Referensi Produk	45
Tabel 4.23 <i>Use Case Scenario</i> Lihat Total Transaksi Per Pekan.....	46
Tabel 4.24 <i>Use Case Scenario</i> Lihat Total Transaksi Per Bulan	46
Tabel 4.25 <i>Use Case Scenario</i> Lihat Total Pendapatan Transaksi Per Pekan.....	47

Tabel 4.26 <i>Use Case Scenario</i> Lihat Total Pendapatan Transaksi Per Bulan	47
Tabel 4.27 <i>Use Case Scenario</i> Lihat Total Pendapatan Transaksi Per Bulan (Lanjutan)	48
Tabel 4.28 <i>Use Case Scenario</i> Lihat Pendapatan Transaksi Tertinggi Per Pekan..	48
Tabel 4.29 <i>Use Case Scenario</i> Lihat Pendapatan Transaksi Tertinggi Per Bulan...	49
Tabel 4.30 <i>Use Case Scenario</i> Lihat Pendapatan Transaksi Terendah Per Pekan.	49
Tabel 4.31 <i>Use Case Scenario</i> Lihat Pendapatan Transaksi Terendah Per Pekan (Lanjutan)	50
Tabel 4.32 <i>Use Case Scenario</i> Lihat Pendapatan Transaksi Terendah Per Bulan .	50
Tabel 4.33 <i>Use Case Scenario</i> Lihat Rata-Rata Pendapatan Transaksi Per Pekan.	51
Tabel 4.34 <i>Use Case Scenario</i> Lihat Rata-Rata Pendapatan Transaksi Per Bulan .	51
Tabel 4.35 <i>Use Case Scenario</i> Lihat Rata-Rata Pendapatan Transaksi Per Bulan (Lanjutan)	52
Tabel 4.36 <i>Use Case Scenario</i> Lihat Prediksi Pendapatan	52
Tabel 4.37 <i>Use Case Scenario</i> Lihat 10 Hari Dengan Pendapatan Tertinggi	53
Tabel 4.38 <i>Use Case Scenario</i> Lihat 10 Hari Dengan Pendapatan Terendah.....	53
Tabel 4.39 <i>Use Case Scenario</i> Lihat 10 Hari Dengan Pendapatan Terendah (Lanjutan)	54
Tabel 4.40 <i>Use Case Scenario</i> Lihat Riwayat Transaksi	54
Tabel 4.41 <i>Use Case Scenario</i> Lihat Rincian Riwayat Transaksi	55
Tabel 4.42 <i>Use Case Scenario</i> Hapus Transaksi.....	55
Tabel 4.43 <i>Use Case Scenario</i> Hapus Transaksi (Lanjutan).....	56
Tabel 4.44 <i>Use Case Scenario</i> Cari Transaksi	56
Tabel 4.45 <i>Use Case Scenario</i> Cari Transaksi (Lanjutan).....	57
Tabel 4.46 <i>Use Case Scenario</i> Urut Transaksi	57
Tabel 4.47 <i>Use Case Scenario</i> Lihat Kombinasi Produk	58
Tabel 4.48 <i>Use Case Scenario</i> Cari Kombinasi Produk.....	58
Tabel 4.49 <i>Use Case Scenario</i> Cari Kombinasi Produk (Lanjutan).....	59
Tabel 4.50 <i>Use Case Scenario</i> Urut Kombinasi Produk	59
Tabel 4.51 <i>Use Case Scenario</i> Lihat Jumlah Produk Terjual.....	60
Tabel 4.52 <i>Use Case Scenario</i> Cari Jumlah Produk Terjual	60
Tabel 4.53 <i>Use Case Scenario</i> Urut Jumlah Produk Terjual	61
Tabel 4.54 <i>Use Case Scenario</i> Lihat Daftar Produk	61



Tabel 4.55 <i>Use Case Scenario</i> Lihat Daftar Produk (Lanjutan).....	62
Tabel 4.56 <i>Use Case Scenario</i> Cari Produk.....	62
Tabel 4.57 <i>Use Case Scenario</i> Tambah Produk	63
Tabel 4.58 <i>Use Case Scenario</i> Hapus Produk	63
Tabel 4.59 <i>Use Case Scenario</i> Hapus Produk (Lanjutan)	64
Tabel 4.60 <i>Use Case Scenario</i> Lihat Daftar Kategori.....	64
Tabel 4.61 <i>Use Case Scenario</i> Lihat Daftar Kategori (Lanjutan).....	65
Tabel 4.62 <i>Use Case Scenario</i> Cari Kategori.....	65
Tabel 4.63 <i>Use Case Scenario</i> Tambah Kategori	65
Tabel 4.64 <i>Use Case Scenario</i> Tambah Kategori (Lanjutan)	66
Tabel 4.65 <i>Use Case Scenario</i> Hapus Kategori	66
Tabel 4.66 <i>Use Case Scenario</i> Hapus Kategori (Lanjutan)	67
Tabel 4.67 <i>Use Case Scenario</i> Lihat Daftar Akun.....	67
Tabel 4.68 <i>Use Case Scenario</i> Cari Akun.....	68
Tabel 4.69 <i>Use Case Scenario</i> Tambah Akun	68
Tabel 4.70 <i>Use Case Scenario</i> Tambah Akun (Lanjutan).....	69
Tabel 4.71 <i>Use Case Scenario</i> Hapus Akun	69
Tabel 4.72 <i>Use Case Scenario</i> Hapus Akun (Lanjutan).....	70
Tabel 4.73 <i>Use Case Scenario</i> Atur Nilai Parameter Prediksi.....	70
Tabel 4.74 <i>Use Case Scenario</i> Lihat Informasi Prediksi Pendapatan.....	71
Tabel 4.75 <i>Use Case Scenario</i> Atur Nilai Parameter Produk Potensial.....	71
Tabel 4.76 <i>Use Case Scenario</i> Atur Nilai Parameter Produk Potensial (Lanjutan)	72
Tabel 4.77 <i>Use Case Scenario</i> Lihat Daftar Produk Potensial	72
Tabel 4.78 <i>Use Case Scenario</i> Lihat Daftar Produk Potensial (Lanjutan)	73
Tabel 4.79 <i>Use Case Scenario</i> Cari Produk Potensial	73
Tabel 4.80 <i>Use Case Scenario</i> Urut Produk Potensial.....	73
Tabel 4.81 <i>Use Case Scenario</i> Urut Produk Potensial (Lanjutan).....	74
Tabel 4.82 <i>Use Case Scenario</i> Lihat Informasi Produk Potensial	74
Tabel 4.83 <i>Use Case Scenario</i> Lihat Informasi Produk Potensial (Lanjutan)	75
Tabel 5.1 <i>Pseudocode</i> algoritme method <i>InsertTableCart</i>	83
Tabel 5.2 <i>Pseudocode</i> algoritme method <i>InsertTableCart</i> (Lanjutan)	84
Tabel 5.3 <i>Pseudocode</i> algoritme method <i>Prediction_Data</i>	84



Tabel 5.4 <i>Pseudocode</i> Algoritme <i>Recommendation_Data</i>	85
Tabel 5.5 Keterangan Perancangan Antarmuka Halaman Transaksi	86
Tabel 5.6 Keterangan Perancangan Antarmuka Halaman Transaksi (Lanjutan) ..	87
Tabel 5.7 Keterangan Perancangan Antarmuka Prediksi Pendapatan.....	88
Tabel 5.8 Keterangan Perancangan Antarmuka Kombinasi Produk	88
Tabel 5.9 Keterangan Perancangan Antarmuka Kombinasi Produk (Lanjutan)....	89
Tabel 5.10 Spesifikasi Perangkat Keras.....	89
Tabel 5.11 Spesifikasi Perangkat Lunak.....	89
Tabel 5.12 Spesifikasi Perangkat Lunak (Lanjutan)	90
Tabel 5.13 kode program <i>method InsertTableCart</i>	90
Tabel 5.14 kode program <i>method InsertTableCart</i> (Lanjutan)	91
Tabel 5.15 kode program <i>method Prediction_Data</i>	91
Tabel 5.16 kode program <i>method Recommendation_Data</i>	92
Tabel 5.17 Implementasi Tabel <i>Category</i>	92
Tabel 5.18 Implementasi Tabel <i>ItemSold</i>	93
Tabel 5.19 Implementasi Tabel <i>LowIncome</i>	93
Tabel 5.20 Implementasi Tabel <i>OrderDetails</i>	93
Tabel 5.21 Implementasi Tabel <i>PotentialProduct</i>	94
Tabel 5.22 Implementasi Tabel <i>Prediction</i>	94
Tabel 5.23 Implementasi Tabel <i>PredictParameter</i>	94
Tabel 5.24 Implementasi Tabel <i>Product</i>	95
Tabel 5.25 Implementasi Tabel <i>Recommendation</i>	95
Tabel 5.26 Implementasi Tabel <i>TopIncome</i>	95
Tabel 5.27 Implementasi Tabel <i>TransactionHeader</i>	96
Tabel 5.28 Implementasi Tabel <i>TrendsParameter</i>	96
Tabel 5.29 Implementasi Tabel <i>User</i>	96
Tabel 6.1 <i>Pseudocode</i> Algoritme <i>InsertTableCart</i>	99
Tabel 6.2 Hasil pengujian unit <i>method InsertTableCart</i>	101
Tabel 6.3 <i>Pseudocode</i> Algoritme <i>method Prediction_Data</i>	101
Tabel 6.4 <i>Pseudocode</i> Algoritme <i>method Prediction_Data</i> (Lanjutan).....	102
Tabel 6.5 Hasil pengujian unit <i>method Prediction_Data</i>	103
Tabel 6.6 <i>Pseudocode</i> Algoritme <i>method Recommendation_Data</i>	103

Tabel 6.7 Hasil pengujian unit <i>method Recommendation_Data</i>	104
Tabel 6.8 Hasil pengujian unit <i>method Recommendation_Data</i> (Lanjutan)	105
Tabel 6.9 <i>Pseudocode</i> Algoritme <i>method Prediction_Data</i>	105
Tabel 6.10 Hasil pengujian integrasi.....	107
Tabel 6.11 Pengujian validasi <i>Login</i>	108
Tabel 6.12 Pengujian validasi <i>Login</i> - Alternatif.....	108
Tabel 6.13 Pengujian validasi <i>Logout</i>	108
Tabel 6.14 Pengujian validasi <i>Logout</i> (Lanjutan).....	109
Tabel 6.15 Pengujian validasi Tambah Transaksi	109
Tabel 6.16 Pengujian validasi Tambah Transaksi – Alternatif 1.....	109
Tabel 6.17 Pengujian validasi Tambah Transaksi – Alternatif 1 (Lanjutan)	110
Tabel 6.18 Pengujian validasi Tambah Transaksi – Alternatif 2.....	110
Tabel 6.19 Pengujian validasi Tambah Transaksi – Alternatif 3.....	110
Tabel 6.20 Pengujian validasi Tambah Transaksi – Alternatif 4.....	111
Tabel 6.21 Pengujian validasi Tambah Transaksi – Alternatif 5.....	111
Tabel 6.22 Pengujian validasi Tambah Transaksi – Alternatif 6.....	111
Tabel 6.23 Pengujian validasi Tambah Transaksi – Alternatif 6 (Lanjutan)	112
Tabel 6.24 Pengujian validasi Hapus <i>Item</i> Transaksi	112
Tabel 6.25 Pengujian validasi <i>Reset</i> Transaksi	113
Tabel 6.26 Pengujian validasi Lihat Referensi Produk.....	113
Tabel 6.27 Pengujian validasi Cari Referensi Produk	114
Tabel 6.28 Pengujian validasi Lihat Total Transaksi Per Pekan.....	114
Tabel 6.29 Pengujian validasi Lihat Total Transaksi Per Bulan	115
Tabel 6.30 Pengujian validasi Lihat Total Pendapatan Transaksi Per Pekan	115
Tabel 6.31 Pengujian validasi Lihat Total Pendapatan Transaksi Per Pekan (Lanjutan)	116
Tabel 6.32 Pengujian validasi Lihat Total Pendapatan Transaksi Per Bulan.....	116
Tabel 6.33 Pengujian validasi Lihat Pendapatan Transaksi Tertinggi Per Pekan	117
Tabel 6.34 Pengujian validasi Lihat Pendapatan Transaksi Tertinggi Per Bulan .	117
Tabel 6.35 Pengujian validasi Lihat Pendapatan Transaksi Terendah Per Pekan	118
Tabel 6.36 Pengujian validasi Lihat Pendapatan Transaksi Terendah Per Bulan	118
Tabel 6.37 Pengujian validasi Lihat Pendapatan Transaksi Terendah Per Bulan (Lanjutan)	119



Tabel 6.38 Pengujian validasi Lihat Rata-Rata Pendapatan Transaksi Per Pekan	119
Tabel 6.39 Pengujian validasi Lihat Rata-Rata Pendapatan Transaksi Per Bulan	120
Tabel 6.40 Pengujian validasi Lihat Prediksi Pendapatan.....	120
Tabel 6.41 Pengujian validasi Lihat 10 Hari Dengan Pendapatan Tertinggi	121
Tabel 6.42 Pengujian validasi Lihat 10 Hari Dengan Pendapatan Terendah	121
Tabel 6.43 Pengujian validasi Lihat Riwayat Transaksi.....	122
Tabel 6.44 Pengujian validasi Lihat Riwayat Transaksi.....	122
Tabel 6.45 Pengujian validasi Lihat Riwayat Transaksi (Lanjutan)	123
Tabel 6.46 Pengujian validasi Hapus Transaksi.....	123
Tabel 6.47 Pengujian validasi Hapus Transaksi – Alternatif 1.....	123
Tabel 6.48 Pengujian validasi Hapus Transaksi – Alternatif 1 (Lanjutan)	124
Tabel 6.49 Pengujian validasi Cari Transaksi.....	124
Tabel 6.50 Pengujian validasi Urut Transaksi.....	124
Tabel 6.51 Pengujian validasi Urut Transaksi (Lanjutan).....	125
Tabel 6.52 Pengujian validasi Urut Transaksi – Alternatif 1	125
Tabel 6.53 Pengujian validasi Lihat Kombinasi Produk	125
Tabel 6.54 Pengujian validasi Lihat Kombinasi Produk (Lanjutan).....	126
Tabel 6.55 Pengujian validasi Cari Kombinasi Produk	126
Tabel 6.56 Pengujian validasi Urut Kombinasi Produk.....	126
Tabel 6.57 Pengujian validasi Urut Kombinasi Produk (Lanjutan)	127
Tabel 6.58 Pengujian validasi Urut Kombinasi Produk – alternatif 1	127
Tabel 6.59 Pengujian validasi Lihat Jumlah Produk Terjual	127
Tabel 6.60 Pengujian validasi Cari Jumlah Produk Terjual	128
Tabel 6.61 Pengujian validasi Urut Jumlah Produk Terjual	128
Tabel 6.62 Pengujian validasi Urut Jumlah Produk Terjual – Alternatif 1	129
Tabel 6.63 Pengujian validasi Lihat Daftar Produk.....	129
Tabel 6.64 Pengujian validasi Cari Produk	130
Tabel 6.65 Pengujian validasi Tambah Produk	130
Tabel 6.66 Pengujian validasi Tambah Produk – Alternatif 1	130
Tabel 6.67 Pengujian validasi Tambah Produk – Alternatif 1 (Lanjutan)	131
Tabel 6.68 Pengujian validasi Tambah Produk – Alternatif 2	131
Tabel 6.69 Pengujian validasi Tambah Produk – Alternatif 3	131

Tabel 6.70 Pengujian validasi Tambah Produk – Alternatif 3 (Lanjutan)	132
Tabel 6.71 Pengujian validasi Hapus Produk	132
Tabel 6.72 Pengujian validasi Hapus Produk – Alternatif 1	132
Tabel 6.73 Pengujian validasi Lihat Daftar Kategori.....	133
Tabel 6.74 Pengujian validasi Cari Kategori	133
Tabel 6.75 Pengujian validasi Tambah Kategori.....	134
Tabel 6.76 Pengujian validasi Tambah Kategori – Alternatif 1	134
Tabel 6.77 Pengujian validasi Tambah Produk – Alternatif 2	134
Tabel 6.78 Pengujian validasi Tambah Produk – Alternatif 2 (Lanjutan)	135
Tabel 6.79 Pengujian validasi Hapus Kategori	135
Tabel 6.80 Pengujian validasi Hapus Produk – Alternatif 1	135
Tabel 6.81 Pengujian validasi Lihat Daftar Akun.....	136
Tabel 6.82 Pengujian validasi Cari Akun	136
Tabel 6.83 Pengujian validasi Tambah Akun.....	137
Tabel 6.84 Pengujian validasi Tambah Akun – Alternatif 1	137
Tabel 6.85 Pengujian validasi Tambah Produk – Alternatif 2	137
Tabel 6.86 Pengujian validasi Tambah Produk – Alternatif 2 (Lanjutan)	138
Tabel 6.87 Pengujian validasi Hapus Akun	138
Tabel 6.88 Pengujian validasi Hapus Akun – Alternatif 1	138
Tabel 6.89 Pengujian validasi Hapus Akun – Alternatif 1 (Lanjutan)	139
Tabel 6.90 Pengujian validasi Atur Nilai Parameter Prediksi.....	139
Tabel 6.91 Pengujian validasi Atur Nilai Parameter Prediksi - Alternatif 1	139
Tabel 6.92 Pengujian validasi Atur Nilai Parameter Prediksi - Alternatif 2	140
Tabel 6.93 Pengujian validasi Lihat Informasi Prediksi Pendapatan	140
Tabel 6.94 Pengujian validasi Atur Nilai Parameter Produk Potensial.....	141
Tabel 6.95 Pengujian validasi Atur Nilai Parameter Produk Potensial.....	141
Tabel 6.96 Pengujian validasi Atur Nilai Parameter Produk Potensial.....	141
Tabel 6.97 Pengujian validasi Atur Nilai Parameter Produk Potensial.....	142
Tabel 6.98 Pengujian validasi Atur Nilai Parameter Produk Potensial.....	142
Tabel 6.99 Pengujian validasi Lihat Daftar Produk Potensial.....	142
Tabel 6.100 Pengujian validasi Lihat Daftar Produk Potensial (Lanjutan).....	143
Tabel 6.101 Pengujian validasi Cari Produk Potensial.....	143

Tabel 6.102 Pengujian validasi Urut Produk Potensial..... 143

Tabel 6.103 Pengujian validasi Urut Produk Potensial (Lanjutan) 144

Tabel 6.104 Pengujian validasi Urut Produk Potensial – Alternatif 1..... 144

Tabel 6.105 Pengujian validasi Lihat Informasi Produk Potensial..... 144

Tabel 6.106 Pengujian validasi Lihat Informasi Produk Potensial (Lanjutan)..... 145

Tabel 6.107 Daftar Pertanyaan SUS..... 145

Tabel 6.108 Daftar Pertanyaan SUS (Lanjutan)..... 146

Tabel 6.109 Hasil Kuisiner Skor tiap pertanyaan SUS..... 146

Tabel 6.110 Hasil Konversi Skor tiap pertanyaan SUS..... 147

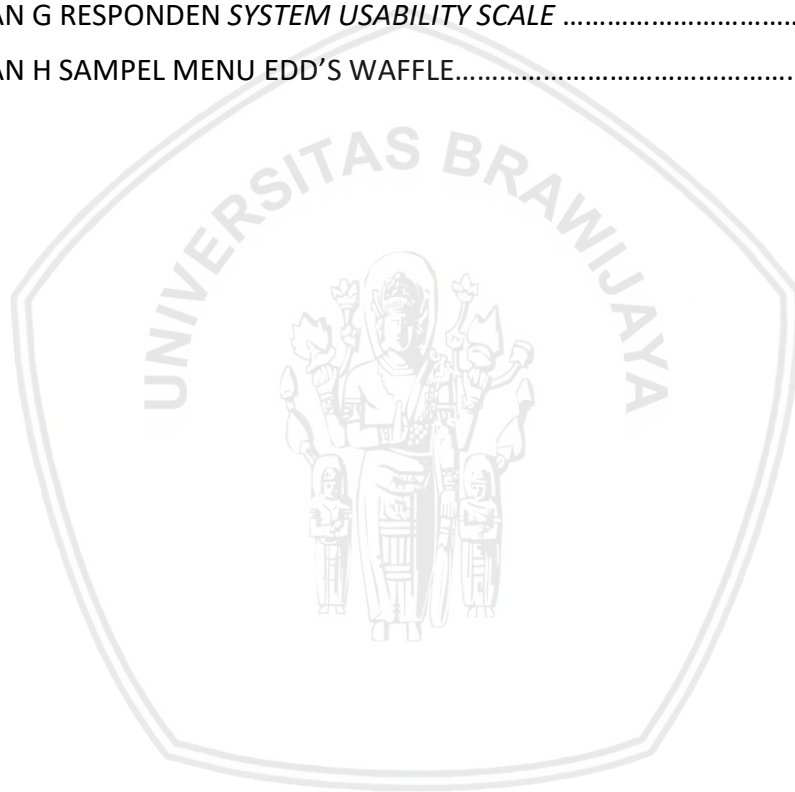


DAFTAR GAMBAR

Gambar 2.1 Model <i>Waterfall</i>	8
Gambar 2.2 Contoh <i>Class Diagram</i>	13
Gambar 2.3 Edd's Waffle	14
Gambar 2.4 Visualisasi Chart Kendo UI	15
Gambar 2.5 Rating dan skala konversi skor rata-rata SUS	21
Gambar 3.1 Diagram Alur Penelitian.....	22
Gambar 4.1 <i>Use Case Diagram Point of Sale</i> Edd's Waffle.....	38
Gambar 5.1 Sequence Diagram Tambah Transaksi.....	77
Gambar 5.2 <i>Sequence Diagram</i> Lihat Prediksi Pendapatan	78
Gambar 5.3 Sequence Diagram Lihat Kombinasi Produk.....	79
Gambar 5.4 <i>Class Diagram</i> Perancangan Umum	80
Gambar 5.5 <i>Class Diagram</i> pada <i>Model</i>	80
Gambar 5.6 <i>Class Diagram</i> pada <i>Controller</i>	81
Gambar 5.7 Perancangan ERD	82
Gambar 5.8 Perancangan PDM	83
Gambar 5.9 Perancangan Antarmuka Halaman Transaksi	86
Gambar 5.10 Perancangan Antarmuka Prediksi Pendapatan.....	87
Gambar 5.11 Perancangan Antarmuka Kombinasi Produk	88
Gambar 5.12 Implementasi Antarmuka Halaman Transaksi	97
Gambar 5.13 Implementasi Antarmuka Prediksi Pendapatan	98
Gambar 5.14 Implementasi Antarmuka Kombinasi Produk	98
Gambar 6.1 <i>Flowgraph method InsertTableCart</i>	100
Gambar 6.2 <i>Flowgraph method Prediction_Data</i>	102
Gambar 6.3 <i>Flowgraph method Recommendation_Data</i>	104
Gambar 6.4 <i>Flowgraph method Prediction_Data</i>	106

DAFTAR LAMPIRAN

LAMPIRAN A SURAT BUKTI ELISITASI KEBUTUHAN.....	152
LAMPIRAN B HASIL WAWANCARA.....	153
LAMPIRAN C SAMPEL DATA PENJUALAN.....	156
LAMPIRAN D PERHITUNGAN MANUAL PREDIKSI.....	157
LAMPIRAN E PERHITUNGAN MANUAL KOMBINASI PRODUK.....	160
LAMPIRAN F KUESIONER <i>SYSTEM USABILITY SCALE</i>	170
LAMPIRAN G RESPONDEN <i>SYSTEM USABILITY SCALE</i>	174
LAMPIRAN H SAMPEL MENU EDD’S WAFFLE.....	178



BAB 1 PENDAHULUAN

1.1 Latar belakang

Bisnis *restaurant*, merupakan bisnis yang memiliki peluang berkembang terbesar dan prospek yang cerah (Sularto, Wardoyo dan Yunitasari, 2015). Saat ini, bisnis kuliner selalu tersedia di setiap wilayah baik dalam skala kecil maupun skala besar. Industri kuliner yang bertumpu pada makanan dan minuman dikatakan sebagai bisnis dengan potensi yang tinggi. Hal ini dikarenakan keberadaannya yang merupakan salah satu kebutuhan utama dari berbagai kalangan. Dalam bisnis kuliner, pemilik usaha dituntut untuk memiliki fasilitas dan pelayanan yang lebih unggul sesuai dengan kebutuhan konsumen untuk mempertahankan eksistensinya (Santoso, 2016).

Edd's Waffle merupakan salah satu *restaurant* di Kota Pekanbaru yang berdiri di tahun 2018. *Restaurant* ini didirikan oleh Gledies Putri Irza, seorang lulusan manajemen bisnis James Cook University Singapore. Saat ini, terdapat beberapa sajian yang tersedia di Edd's Waffle, seperti *waffle*, nasi dan mie, *snacks*, *mocktails*, *latte* hingga *juice*. Seiring perkembangannya saat ini, Edd's Waffle saat ini merupakan salah satu *restaurant* yang digemari remaja, hal ini tidak terlepas dari lokasi Edd's Waffle sendiri yang terletak di dekat salah satu universitas ternama di kota Pekanbaru (Irza, 2018). Harga yang ditawarkan Edd's Waffle sendiri cukup bervariasi, yakni diantara 7.000-33.000 rupiah untuk setiap satuannya.

Dalam bisnis kuliner, sebuah *restaurant* membutuhkan sistem pendukung yang lebih dari sekedar *cash register* dan *cash drawer* (Gillum dan Rob, 2011). Saat ini, Edd's Waffle menggunakan mesin kasir standar yakni Cash Register Sharp XE-A102 untuk mendukung transaksi penjualan. Dengan mesin kasir standar ini, meninggalkan permasalahan dalam hal perkembangan penjualan dikarenakan kemampuannya yang hanya mampu mencetak resi penjualan dan menyediakan laporan keuangan dalam wadah *X-reports* dan *Z-reports*. *X-reports* berfungsi untuk menampilkan lampiran daftar transaksi yang telah terjadi, serta *Z-reports* untuk menampilkan lampiran tersebut secara periodik. Dengan keterbatasan dari mesin kasir sendiri, menyebabkan proses pencatatan penjualan kerap dilakukan dengan cara menduplikasi informasi yang tersedia ke *platform* lain. Hal ini mengakibatkan beberapa permasalahan yang lebih mendalam baik dari sisi kelalaian saat melakukan duplikasi informasi hingga ketersediaan laporan yang ada dimana bersifat statis sehingga menghasilkan informasi yang tidak optimal dari data-data penjualan yang ada (Irza, 2018).

Sistem *point of sale* saat ini populer dikarenakan keberadaannya yang menghadirkan kecepatan dan kenyamanan pada transaksi bisnis. Pada umumnya, sistem *point of sale* terdiri atas layanan transaksi, laporan, manajemen penjualan, dan lainnya (Kabir dan Han, 2016). Lebih lanjut, pengembangan sistem *point of sale* tentunya juga dapat memanfaatkan data-data yang tersedia agar dapat menghasilkan informasi-informasi yang berguna. Informasi tersebut dapat berupa

pengetahuan yang dihasilkan pada saat ini maupun di waktu yang akan datang. Informasi pada saat ini dapat di contohkan dengan mengidentifikasi *trend* pembelian yang ada hingga saat ini, sedangkan informasi di waktu yang akan datang dapat di contohkan dengan estimasi atau prediksi pendapatan.

Seiring dengan perkembangannya, Edd's Waffle tentunya dituntut untuk semakin optimal, baik dari sisi kualitas maupun layanan untuk menjaga kepercayaan dari konsumen. Beberapa hal yang dapat dilakukan untuk mencapainya dengan menjadikan data-data pembelian konsumen menjadi informatif dengan mengidentifikasi *trend* yang ada hingga memperkirakan pendapatan yang akan diterima. Dalam bisnis kuliner, *trend* memiliki peranan penting dalam mengoptimalkan penjualan. *Trend* yang dimaksud meliputi jenis produk baik makanan dan minuman dari setiap kategori yang paling diminati dan kurang diminati. Dengan memperhatikan *trend* penjualan yang berdasarkan kepada data-data penjualan yang ada dapat mengurangi hal-hal yang tidak diinginkan seperti *clearance sales* (Cote dan McCarthy, 2015). *Clearance sales* merupakan salah satu cara yang dilakukan penjual untuk menghindari kerugian dengan cara menurunkan harga untuk produk yang jumlah penjualannya tidak mencapai target. Selain untuk menghindari *clearance sales*, adanya *trend* juga dapat dijadikan sebagai acuan untuk memprediksi secara akurat mengenai jenis produk mana yang diutamakan untuk dijual beserta jumlahnya untuk memenuhi kebutuhan konsumen di waktu yang akan datang (Lee, 2001). Kebutuhan di waktu yang akan datang berkaitan pula dengan prediksi. Prediksi adalah salah satu komponen penting dalam keberhasilan *restaurant* (Lasek, Cercone dan Saunders, 2016). Edd's Waffle saat ini belum mampu memprediksi atau memperkirakan pendapatan. Padahal, dengan adanya prediksi pendapatan, Edd's Waffle dapat menjadikannya sebagai acuan dalam menetapkan target pendapatan di waktu yang akan datang. Selain itu, keuntungan dengan adanya prediksi pendapatan ialah meminimalisir kemungkinan semakin besarnya modal. Lebih lanjut, pengadaan atau modal yang tepat sasaran tentunya akan menghindari kerugian seperti terdapatnya bahan yang kedaluwarsa.

Kajian penelitian terdahulu menunjukkan bahwa terdapat beberapa faktor krusial dalam pengembangan *restaurant*, terutama *restaurant* yang tergolong *stand alone* atau berdiri sendiri. Secara garis besar, terdapat 2 faktor yang mempengaruhi perkembangan bisnis *restaurant*, yakni faktor internal dan eksternal. Faktor internal terdiri dari produk atau *menu* yang ditawarkan, kemampuan finansial, sistem pendukung, dan *staff*. Sedangkan faktor eksternal yaitu lokasi, segmentasi pasar, kompetitor, dan status ekonomi konsumen (Jalis, Kassim dan Mohamad, 2012). Oleh karena itu, penelitian ini berorientasi untuk mendukung perkembangan bisnis *restaurant* Edd's Waffle baik dari sisi pengadaan sistem pendukung yang mampu mencatat produk yang ditawarkan, hingga mengidentifikasi *trend* pembelian agar dapat kompetitif di antara para kompetitor.

Adapun terdapat 2 algoritme yang diajukan untuk pengembangan penelitian ini. Algoritme pertama yang digunakan ialah algoritme *Holt Exponential*

Smoothing. Algoritme ini merupakan salah satu jenis dalam metode *Exponential Smoothing* selain *Single Exponential Smoothing (SES)*, *Double Exponential Smoothing (DES)*, hingga *Adaptive Response Rate Exponential Smoothing (ARRES)*. Pemilihan algoritme ini dikarenakan keunggulannya pada data time series yang memiliki tren yang bervariasi. Algoritme ini memuluskan tren data secara langsung dan memperdiksi data asli tanpa menggunakan *Secondary Exponential Smoothing*. Karena fleksibilitas yang tinggi, metode ini digunakan secara luas (Li, 2013). Selanjutnya, algoritme yang digunakan ialah algoritme *Apriori*. Algoritme ini merupakan turunan dari metode *Association Rules* untuk mengidentifikasi *trend* penjualan.

Dari uraian permasalahan yang ada, dapat diketahui bahwa saat ini proses dalam mengorganisir penjualan di Edd's Waffle masih memiliki berbagai permasalahan. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sebuah sistem pendukung penjualan *restaurant* atau biasa disebut *Point Of Sale* untuk digunakan Edd's Waffle. *Point Of Sale* merupakan sistem informasi yang berorientasi pada kegiatan penjualan serta sistem yang membantu proses transaksi (Dian et al., 2015). Sistem yang ingin dikembangkan ialah berbasis web yang dapat mengelola transaksi penjualan, mengobservasi riwayat transaksi, memperhatikan *trend* penjualan, hingga memprediksi pendapatan pada setiap pekan dalam empat pekan selanjutnya. Dengan melakukan pengembangan sistem ini, diharapkan mampu membantu Edd's Waffle dalam hal mengoptimalkan performa bisnis.

1.2 Rumusan Masalah

Sebagaimana berbagai hal yang disampaikan pada latar belakang, maka disusunlah rumusan masalah untuk pengembangan Sistem *Point of Sale* pada Edd's Waffle, diantaranya:

1. Bagaimanakah hasil analisis kebutuhan Sistem *Point of Sale* pada Edd's Waffle Pekanbaru sesuai dengan kebutuhan pengguna ?
2. Bagaimanakah hasil perancangan Sistem *Point of Sale* pada Edd's Waffle Pekanbaru sesuai dengan kebutuhan pengguna ?
3. Bagaimanakah hasil implementasi Sistem *Point of Sale* pada Edd's Waffle Pekanbaru ?
4. Bagaimanakah hasil pengujian unit, integrasi, validasi, dan usabilitas dari Sistem *Point of Sale* pada Edd's Waffle Pekanbaru ?

1.3 Tujuan

Tujuan yang diharapkan dari penelitian ini ialah:

1. Menganalisis kebutuhan sistem *Point of Sale* pada Edd's Waffle sesuai dengan kebutuhan pengguna.
2. Merancang sistem *Point of Sale* pada Edd's Waffle sesuai dengan kebutuhan pengguna.

3. Mengimplementasikan sistem *Point of Sale* pada Edd's Waffle dalam sebuah sistem berbasis web.
4. Menghasilkan sistem *Point of Sale* pada Edd's Waffle dalam sebuah sistem berbasis web yang telah melewati tahap pengujian unit, integrasi, validasi, dan usability sehingga dapat digunakan oleh Edd's Waffle.

1.4 Manfaat

Adapun manfaat yang diperoleh dari penelitian ini adalah sebagai berikut:

1. Menghasilkan sebuah media pendukung penjualan dalam bentuk sistem berbasis web sehingga membantu Edd's Waffle.
2. Meminimalisir kemungkinan kesalahan yang dapat ditimbulkan untuk mengorganisir penjualan.
3. Memungkinkan Edd's Waffle untuk menggunakan prediksi pendapatan sebagai acuan dalam mengatur sasaran penjualan.
4. Mengidentifikasi *trend* penjualan untuk mengoptimalkan performa penjualan dari Edd's Waffle.

1.5 Batasan masalah

Untuk mencegah terjadinya penyimpangan dalam proses penulisan penelitian dan tujuan yang sudah direncanakan, maka penulis menetapkan batasan-batasan sebagai berikut:

1. Batasan dari sistem ialah lingkungan operasional sistem dimana hanya dapat diakses oleh *administrator*, *supervisor*, dan kasir dengan beberapa fitur seperti pengelolaan transaksi penjualan, laporan penjualan, mengamati *trend* penjualan, hingga prediksi pendapatan.
2. Algoritme yang digunakan untuk memprediksi pendapatan ialah algoritme *Holt Exponential Smoothing* sedangkan untuk mendapatkan rekomendasi kombinasi produk ialah *Apriori*.
3. Sistem yang dibangun menggunakan bahasa pemrograman *C#* pada *framework* ASP.NET MVC dan ditujukan pada *platform* web.
4. Sistem menggunakan basis data SQL Server.

1.6 Sistematika pembahasan

Sistematika pembahasan deskripsi singkat bagian-bagian yang terdapat dalam skripsi ini adalah sebagai berikut :

1. **BAB I PENDAHULUAN**

Pada bab ini di utarakan latar belakang penelitian, rumusan masalah penelitian, tujuan dari penelitian, manfaat dari penelitian, batasan masalah penelitian hingga sistematika pembahasan penelitian.

2. **BAB II LANDASAN KEPUSTAKAAN**

Pada bab ini di uraikan dan menjelaskan teori-teori terdahulu yang berkaitan dan mendukung fondasi dari penelitian.

3. **BAB III METODE PENELITIAN**

Pada bab ini di utarakan bagaimana dalam menjelaskan metode dan langkah kerja yang dilakukan dalam proses perancangan, bagaimana dalam melakukan analisis kebutuhan, dan cara dalam melakukan implementasi yang menjadi objek studi kasus skripsi.

4. **BAB IV ANALISIS KEBUTUHAN**

Pada bab ini di utarakan bagaimana analisis dalam melakukan elisitasi kebutuhan, penentuan kebutuhan baik dari sisi kebutuhan fungsional dan non-fungsional, dan pemodelan kebutuhan.

5. **BAB V PERANCANGAN DAN IMPLEMENTASI**

Pada bab ini di uraikan proses perancangan dari hasil analisis kebutuhan sebelumnya sehingga dapat menjadi acuan dalam mengimplementasikan sistem. Selain itu, juga akan di uraikan proses implementasi dari berbagai teori yang ada serta mengacu pada kebutuhan dan perancangan dalam melakukan pengembangan sistem.

6. **BAB VI: PENGUJIAN**

Pada bab ini di uraikan proses pengujian dalam bentuk verifikasi dan validasi sehingga sistem dipastikan layak digunakan calon pengguna.

7. **BAB VII PENUTUP**

Pada bab ini di uraikan berbagai kesimpulan dan saran-saran untuk pengembangan lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Kajian pustaka yang digunakan menjelaskan bagaimana dilakukannya analisa dan kajian mengenai penelitian terdahulu yang sejenis mengenai pengembangan sistem sebagai pendukung transaksi penjualan. Terdapat 3 penelitian yang dijadikan sebagai acuan. Penelitian pertama dilakukan di sebuah *restaurant* bernama *Hakubi Japanese Brasserie*, dimana fokus pada penelitian ini ialah mengembangkan sistem yang terintegrasi secara penuh untuk membantu *restaurant* dalam meningkatkan efektifitas dan efisiensi sehingga dapat mengeliminasi keperluan akan banyaknya sumber daya manusia dan tenaga fisik. Sistem *point of sale* ini dikembangkan dalam platform web dan basis data *oracle* dengan rancangan antar muka yang disesuaikan dengan layaknya sebuah sistem *point of sale* dengan beberapa fitur seperti *tracking* pemesanan saat ini, pengaturan stok gudang, dan ketersediaan laporan keuangan (Gillum dan Rob, 2011).

Penelitian kedua yang diangkat ialah mengenai peramalan jumlah kebutuhan perusahaan dengan menggunakan metode *exponential smoothing* untuk menghadirkan efisiensi di bidang *supply chain management*. Jumlah *dataset* yang digunakan berkisar di angka 3880 di setiap bulannya untuk penjualan buku, *Compact Disc* (CD), dan *Digital Versatile Disc* (DVD). Hasil dari peramalan menggunakan metode *exponential smoothing* menunjukkan bahwa metode ini jauh lebih akurat dibandingkan dengan beberapa metode lain yang tersedia termasuk metode yang ada di dalam perusahaan sendiri (Taylor, 2011). Penelitian ini lebih lanjut memiliki kesamaan dengan penelitian yang diajukan, yaitu dalam hal melakukan prediksi dengan menggunakan metode *exponential smoothing*.

Penelitian terakhir menjelaskan bagaimana pengguna dapat menerima kombinasi-kombinasi berupa rekomendasi produk pada sebuah *e-commerce* yang dapat di beli berdasarkan riwayat pencarian dan transaksi pengguna dengan menggunakan algoritme *Apriori*. Dengan orientasi data yang hanya berkaitan dengan riwayat pencarian dan transaksi, penggunaan algoritme *Apriori* sendiri mengeliminasi masalah dalam memberikan rekomendasi produk kepada pengguna seperti tidak diperlukannya penelusuran profil pengguna. Algoritma *Apriori* sendiri merupakan algoritma klasik untuk mempelajari *association rules* (Saxena dan Gaur, 2015). *Association Rules* sendiri merupakan salah satu teknik pada *data mining* untuk mengidentifikasi sebuah kombinasi *item*, dimana algoritma *Apriori* sendiri berfungsi untuk mencari pola hubungan antar satu atau lebih *item* dalam sebuah *dataset*.

Pada penelitian ini, dikembangkan sebuah sistem *point of sale* dengan memanfaatkan data-data yang tersedia sehingga dapat dimanfaatkan oleh Edd's Waffle. Beberapa penelitian yang telah disebutkan sebelumnya akan menjadi dasar atau acuan untuk penelitian ini. Pada penelitian yang mengangkat topik pengembangan sistem *point of sale*, penelitian ini memiliki kemiripan dari studi

kasus yang diangkat yaitu berasal dari *restaurant*. Selain itu, penelitian ini juga mengembangkan sistem *point of sale* pada platform berbasis web. Namun, terdapat pula beberapa perbedaan seperti lingkup pekerjaan dari penelitian ini yang tidak menyentuh bagian-bagian seperti pengelolaan stok pada gudang hingga *tracking* barang yang masuk dan keluar dari gudang. Lebih lanjut, penelitian ini menawarkan pemanfaatan data yang tersedia untuk menghasilkan informasi-informasi baru, dimana hal ini tidak ditemui pada penelitian yang dijadikan sebagai acuan. Kemudian, pada penelitian selanjutnya yang dijadikan sebagai acuan, yakni penelitian dengan memanfaatkan metode *exponential smoothing*, terdapat perbedaan dimana pada penelitian ini penggunaan metode tersebut ditujukan untuk memprediksi pendapatan *restaurant*. Selanjutnya, pada penelitian terakhir yang dijadikan sebagai acuan, yakni mengenai pemanfaatan algoritme apriori pada konsep *association rules*, terdapat perbedaan tujuan penggunaan algoritme dimana pada penelitian ini ditujukan untuk mencari kombinasi demi kombinasi agar mendapatkan rekomendasi paket makanan yang dapat dijual.

2.2 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah disiplin ilmu yang mempelajari berbagai aspek perencanaan dan produksi perangkat lunak dengan prinsip-prinsip rekayasa untuk menghasilkan perangkat lunak yang handal dan bekerja secara efisien pada mesin yang nyata (Pressman, 2011). Lebih lanjut, rekayasa perangkat lunak ialah disiplin teknik yang berkaitan dengan semua proses pembuatan perangkat lunak dimulai dari saat melakukan spesifikasi sistem hingga pemeliharaan setelah sistem digunakan oleh pengguna (Sommerville, 2011). Didalam rekayasa perangkat lunak, dilakukan pendekatan sistematis yang biasa disebut *software process*. *Software process* sendiri terdiri atas empat kegiatan secara garis besar (Sommerville, 2011), yaitu:

1. *Software specification*: tahapan dimana ditentukan perangkat lunak yang akan dibangun beserta batasan-batasan operasi dari perangkat lunak tersebut.
2. *Software development*: tahapan dimana dilakukan perancangan dan pembangunan perangkat lunak.
3. *Software validation*: tahapan dimana dilakukan pengujian perangkat lunak untuk melakukan validasi atau menjamin kesesuaian dengan permintaan pelanggan.
4. *Software evolution*: tahapan dimana dilakukan perubahan terhadap perangkat lunak yang telah dibangun untuk menyesuaikan dengan perubahan permintaan pelanggan dan pasar.

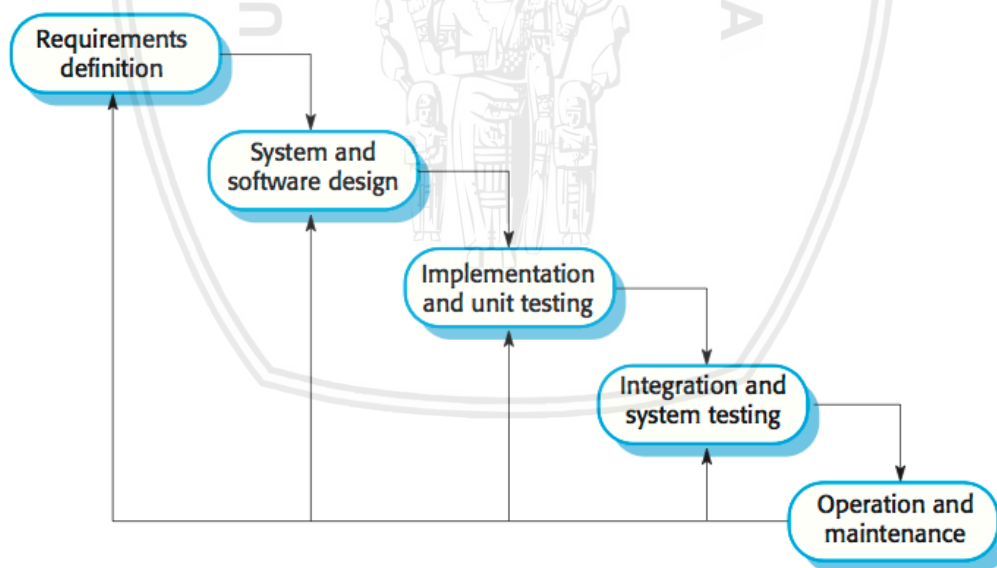
2.3 Software Development Life Cycle

Software Development Life Cycle (SDLC) merupakan proses yang digunakan untuk mengembangkan perangkat lunak mulai dari proses perancangan, pengembangan, hingga pengujian perangkat lunak. SDLC memiliki tujuan untuk menghasilkan produk perangkat lunak yang memiliki kualitas tinggi, yang telah

diselesaikan dengan waktu dan biaya yang sama dengan perkiraan (Sharma & Singh, 2016). Terdapat beberapa jenis SDLC sebagai bentuk model proses pengembangan perangkat lunak baik yang tergolong konvensional hingga modern seperti *waterfall*, *prototyping*, *iterative* dan *incremental*, *spiral development*, *extreme programming*, dan *rapid application development*. Jenis pengembangan perangkat lunak dibagi menjadi 7 tahapan yakni tahapan analisis, kebutuhan, perancangan, implementasi, pengujian, *deployment*, serta pemeliharaan dan perbaikan (Caytiles, 2014).

2.3.1 Model Waterfall

Model *waterfall* adalah salah satu jenis dari SDLC dalam pengembangan perangkat lunak. Pada penelitian ini, digunakan model *waterfall* karena dalam pengembangannya, dilakukan secara bertahap tanpa adanya tumpang tindih. Selain itu, proses yang terdapat di dalam model *waterfall* memiliki skema yang berurutan mulai dari analisa hingga pemeliharaan. Oleh karena itu, dengan menggunakan model *waterfall* yang di setiap prosesnya memiliki spesifikasinya masing-masing, dapat menghasilkan perangkat lunak yang berkualitas baik. Pengembangan perangkat lunak dengan model *waterfall* pada penelitian ini tidak dilakukan hingga proses pemeliharaan atau *maintenance*. Tahapan dalam model *waterfall* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Model Waterfall

Sumber : (Sommerville, 2011)

1. Requirement definition

Pada tahapan ini, dilakukan analisa dalam menentukan masalah bisnis, mengidentifikasi penyebab permasalahan, dan menawarkan solusi dalam bentuk kebutuhan pada sistem. Pendefinisian kebutuhan didefinisikan secara rinci hingga menjadi spesifikasi sistem dengan berkonsultasi dengan pengguna.

2. *System and software design*
Pada tahapan ini, dilakukan proses perancangan yang dihasilkan dari alokasi kebutuhan perangkat lunak dan perangkat keras agar sistem dibuat sesuai dengan rancangan yang telah dibuat.
3. *Implementation and unit testing*
Pada tahapan ini, dilakukan pengimplementasian dari desain sistem yang telah dijabarkan untuk menghasilkan produk perangkat lunak baik dalam bentuk kumpulan program maupun unit program. Unit testing sendiri digunakan untuk melakukan verifikasi kepada setiap unit.
4. *Integration and system testing*
Pada tahapan ini, dilakukan pengujian secara menyeluruh untuk memastikan bahwa kebutuhan perangkat lunak telah terpenuhi disertai dengan performa sistem yang baik. Dalam tahapan ini, dilakukan dua jenis pengujian yakni *blackbox testing* dan *whitebox testing*.
5. *Operation and maintenance*
Pada tahapan ini, sistem akan beroperasi dan digunakan oleh pengguna, dan seiring berjalannya waktu, sistem akan mengalami perubahan baik berupa pemeliharaan maupun perbaikan.

2.4 Pendekatan Berorientasi Objek

Pendekatan berorientasi objek atau biasa disebut *Object Oriented (OO)* merupakan salah satu pendekatan dalam pengembangan perangkat lunak. Dengan menggunakan OO, maka pengembangan perangkat lunak akan dimulai dengan tahapan *Object Oriented Analysis (OOA)*. OOA merupakan tahapan untuk mendefinisikan kebutuhan perangkat lunak disertai objek dan kelas yang didapatkan dari permasalahan. Kemudian, dilakukan *Object Oriented Design (OOD)* yang didasarkan pada tahapan sebelumnya yaitu OOA. OOD terdiri dari perancangan arsitektur sistem, komponen data, dan antarmuka. Selanjutnya, proses perancangan tersebut akan diimplementasikan menjadi kode program dengan menggunakan bahasa pemrograman berorientasi objek atau *Object Oriented Programming* yang biasa disingkat OOP. Sedangkan pada pendekatan berorientasi objek bisa menggunakan strategi pengujian seperti pengujian unit, pengujian integrasi, dan pengujian validasi (Pressman, 2010).

2.5 Pemodelan Berorientasi Objek




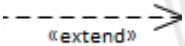
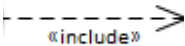
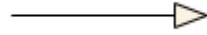
Pemodelan berorientasi objek memodelkan OOA dan OOD dalam UML (*Unified Model Language*) (Pressman, 2010). UML menawarkan standar dalam memodelkan perangkat lunak, dimana pada pengembangan perangkat lunak ini akan menggunakan tiga diagram UML, yakni:

1. *Use Case Diagram*

Diagram pada UML pertama yang digunakan adalah *use case diagram*. *Use case diagram* merupakan sebuah diagram yang digunakan untuk memodelkan perilaku aktor terhadap suatu sistem (Whitten dan Lonnie D. Bentley, 2007). *Use*

case menggambarkan sebuah interaksi antara satu atau lebih aktor dengan fungsionalitas perangkat lunak yang akan dibangun. Di dalam *use case* terdapat notasi yaitu aktor, *use case*, relasi (asosiasi, *includes*, ekstensi dan generalisasi). Notasi yang ada di dalam *use case diagram* dapat dilihat pada Tabel 2.1.

Tabel 2.1 Notasi Use Case Diagram

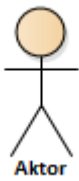

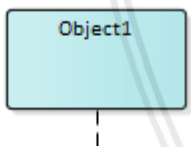

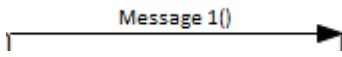
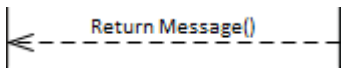
Notasi	Deskripsi Notasi
Aktor 	Segala hal pada luar sistem yang akan menggunakan atau berinteraksi dengan sistem untuk melakukan sesuatu (bertukar informasi)
Use Case 	Sebuah urutan langkah-langkah dan setiap langkah – langkah itu saling terkait dalam proses tindakan, baik tindakan yang manual dan otomatis
Asosiasi / <i>association</i> 	Mengidentifikasi interaksi atau hubungan antara setiap aktor dengan setiap use case
Ekstensi / <i>extend</i> 	Hubungan <i>use case</i> tambahan terhadap <i>use case</i> yang lain dan dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan.
Includes 	Hubungan <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya.
Generalisasi / <i>generalization</i> 	Hubungan suatu class yang umum (general) antara suatu class yang lebih spesifik dengan mengekstraksi karakteristik dari dua class atau lebih dan menggabungkan class-class tersebut dalam superclass umum.

2. Sequence Diagram


Sequence diagram adalah diagram yang menggambarkan bagaimana sebuah objek berinteraksi dengan objek lainnya. Interaksi suatu objek dengan objek lain pada *sequence diagram* dilakukan melalui pesan. Interaksi ini menggambarkan sebuah eksekusi atau operasi dari sebuah fitur perangkat lunak (Whitten dan Lonnie D. Bentley, 2007). Pada pemodelan *sequence diagram* dijelaskan bahwa urutan proses yang terjadi untuk mencapai suatu kebutuhan

sistem. Notasi-notasi yang ada pada *sequence diagram* dipaparkan dalam Tabel 2.2 dan Tabel 2.3.

Tabel 2.2 Notasi *Sequence Diagram*

Notasi	Deskripsi Notasi
<p>Aktor</p> 	Merupakan sebuah komponen yang digunakan sebagai perwakilan dari pengguna untuk berinteraksi dengan sistem
<p>Lifeline</p> 	Suatu garis dengan titik-titik yang terhubung dengan suatu objek
<p>Object</p> 	Digunakan untuk mempresentasikan sebuah objek yang bertindak pada sistem
<p>Activation bar</p> 	Digunakan untuk mewakili eksekusi suatu operasi dari objek, dan panjang dari kotak pada garis titik-titik ini merupakan sebuah durasi aktivasi pada operasi
<p>Input message</p> 	Simbol yang digunakan untuk mengirim pesan antar kelas ataupun bisa dari aktor menuju sistem
<p>Output Message (return)</p> 	Menggambarkan pesan/hubungan antar obyek, yang menunjukkan urutan kejadian yang terjadi.
<p>Recursive</p>	Simbol yang digunakan untuk mengirim pesan pada dirinya sendiri

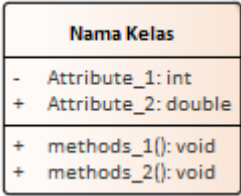
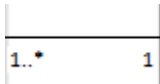
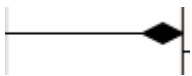
Tabel 2.3 Notasi Sequence Diagram (Lanjutan)

Notasi	Deskripsi Notasi
<p><i>Frame</i></p> 	Simbol yang menggambarkan area interaksi
Entity Class,	Suatu bagian dari sistem yang terdapat kumpulan kelas yang berupa entitas-entitas berbentuk gambaran awal pada sistem dan digunakan untuk menyusun basis data dan menyimpan informasi pada basis data
Boundary Class,	Merupakan kumpulan suatu kelas yang menjadi antarmuka atau interaksi antara aktor dengan sistem
Control class	Suatu objek yang terdapat logika aplikasi.


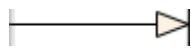
3. Class Diagram

Class Diagram adalah penggambaran grafis struktur objek dari suatu sistem. Pada *class diagram*, di dalamnya menunjukkan objek dari kelas-kelas dari suatu sistem yang saling berhubungan (Whitten dan Lonnie D. Bentley, 2007). Notasi yang terdapat pada *class diagram* dapat dilihat pada Tabel 2.4 dan Tabel 2.5.

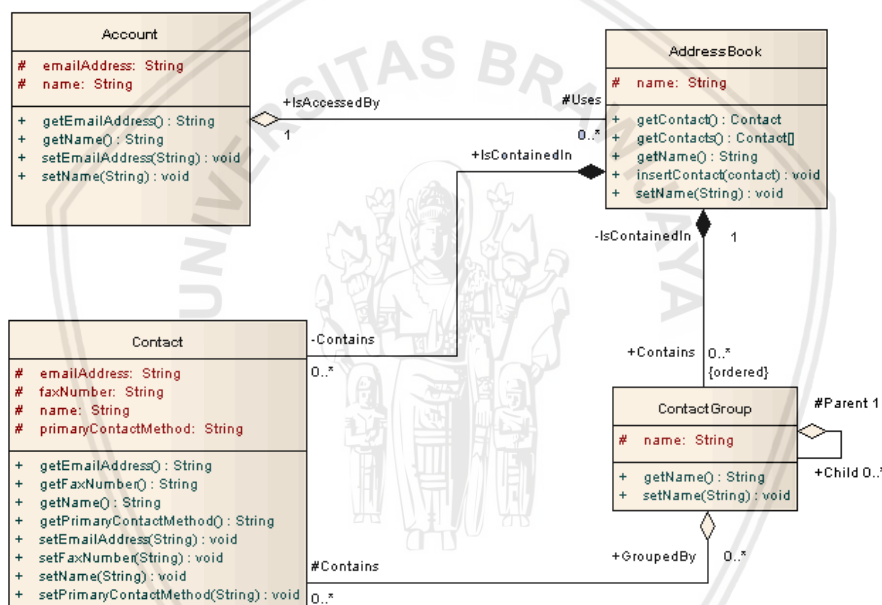
Tabel 2.4 Notasi Class Diagram

Notasi	Deskripsi Notasi
<p><i>Class / Kelas</i></p> 	Sekelompok objek yang memiliki atribut, operasi, relasi dan semantik yang umum.
<p><i>Association</i></p> 	Suatu hubungan antara bagian dua kelas, dengan artian setiap objek dalam kelas yang satu berhubungan dengan semua objek pada kelas yang lain, demikian sebaliknya.
<p><i>Composition</i></p> 	Hubungan yang menggambarkan ketergantungan antar class. Composition bentuk khusus dari agregasi.

Tabel 2.5 Notasi Class Diagram (Lanjutan)

Notasi	Deskripsi Notasi
<p>Aggregations</p> 	Hubungan dengan makna bahwa suatu class menjadi atribut untuk class yang lain
<p>Generalizations</p> 	Hubungan antar class dengan arti dari umum ke khusus atau menyatakan hubungan inheritance

Contoh *class diagram* dapat dilihat pada Gambar 2.2. Pada *class diagram* tersebut, direpresentasikan atribut dan operasi yang terlibat di dalam setiap *class*. Selain itu, pada Gambar 2.2 dijelaskan jenis tipe data atribut dan jenis *modifier* (*public/private/protected*).



Gambar 2.2 Contoh Class Diagram

2.6 Edd’s Waffle

Edd’s Waffle merupakan *restaurant* yang terletak di Jl. Marsan Sejahtera, Panam, Kota Pekanbaru, Riau, Indonesia. *Restaurant* yang berdiri di awal 2018 ini menjual berbagai jenis makanan dengan fokus utama berupa varian rasa *Waffle*. Gambar bangunan fisik tampak depan Edd’s Waffle dapat dilihat pada Gambar 2.3. 2018 merupakan tahun yang krusial bagi Edd’s Waffle, dengan kisaran harga yang bersaing dengan *restaurant* atau *cafe* lainnya, jumlah pengunjung *restaurant* kerap menunjukkan kenaikan dimana mayoritas pengunjung kebanyakan didominasi oleh remaja. Saat ini, Edd’s Waffle menggunakan mesin kasir standar yakni Cash Register Sharp XE-A102 yang hanya mampu mendukung penjualan dalam hal transaksi dan penyimpanan uang (Irza, 2018).



Edd's Waffle memiliki karyawan yang terdiri dari berbagai posisi seperti *Owner, Supervisor, Chef, Barista, Waitress*, hingga *Cashier*. Edd's Waffle menjual berbagai jenis menu dimana untuk makanan dikelompokkan dalam *Waffle, Nasi dan Mie, Western, dan Snacks*. Sedangkan untuk minuman. Sedangkan untuk minuman dikelompokkan dalam *Coffee, Tea, Juice, Latte, Mocktail, dan Frappe*. Rentang harga yang ditawarkan juga bervariasi mulai dari Rp 7.000 hingga Rp. 33.000.



Gambar 2.3 Edd's Waffle

2.7 Teknologi Pengembangan Sistem

2.7.1 ASP .NET MVC Framework

ASP .Net MVC merupakan *framework* atau kerangka kerja untuk pengembangan web (Freeman, 2016). ASP.Net MVC dapat dijalankan pada sistem operasi *Microsoft Windows*. Adapun *framework* ini menawarkan kerapian dan efektifitas dengan mengimplementasikan pola *Model – View – Controller (MVC)*. Dengan pola MVC, struktur dari kode program yang dihasilkan ASP .Net MVC memiliki susunan yang jelas.

MVC adalah pola arsitektur yang digunakan untuk pengembangan perangkat lunak (Caytiles, 2014). Arsitektur dari pola *MVC* memisahkan struktur program kedalam 3 bagian atau representasi yang berbeda *MVC* merupakan metode dalam pembuatan pengembangan aplikasi yang memisahkan data (*Model*) dengan tampilan (*View*) dan bagaimana cara mengontrol dan meresponnya (*Controller*). Nantinya, *Model* akan mewakili struktur data yang biasanya terdiri dari fungsi-fungsi yang membantu dalam pengelolaan basis data seperti melakukan pembaruan, penambahan, dan pengurangan data. Sedangkan *View* berarti bagian yang mengatur tampilan sistem. Lalu *Controller* akan menjadi penyambung atau berperan sebagai jembatan dalam menghubungkan *View* dan *Model*. Nantinya, *Controller* akan memproses suatu data agar ditampilkan pada sistem.

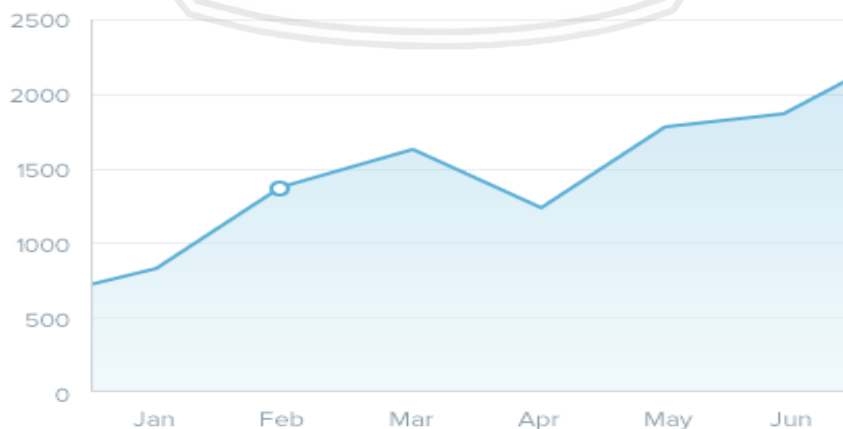
2.7.2 Entity Framework

Entity Framework adalah *Object Relational Mapping* (ORM) buatan Microsoft untuk *.NET framework* guna memetakan tabel dari *database* menjadi objek. Dengan menggunakan *Entity Framework*, pengguna dapat mengeliminasi baris kode dalam mengakses data. Untuk menggunakan *Entity Framework*, *Integrated Development Environment* (IDE) yang dapat digunakan ialah Visual Studio Community 2017, dimana sudah terintegrasi secara otomatis dengan *.NET framework* dan *database SQL Server*. Penggunaan *Entity Framework* terbagi menjadi 2 jenis, yakni:

1. *Code First*: Sebuah metode pada *Entity Framework* dimana dilakukan proses pembuatan kode terlebih dahulu, lalu membuat *table* berdasarkan kelas tersebut.
2. *Database First*: Sebuah metode yang melakukan pembuatan *table* terlebih dahulu, lalu membuat kode program.

2.7.3 Kendo UI

Kendo UI merupakan *library* JQuery yang digunakan untuk pengembangan web yang tidak bergantung pada mempermudah pengguna dalam mengulas dan menampilkan data yang ada pada *Database*. Kendo UI merupakan produk yang dirilis oleh Telerik, dimana salah satunya ditujukan kepada pengguna *framework ASP .Net MVC* untuk meningkatkan produktifitas (Adams, 2013). Pada Kendo UI, terdapat beberapa *plugins* yang mendukung aplikasi berbasis web seperti *auto complete*, *grid*, dan *list view*. *Auto Complete* memberikan *suggestion* secara otomatis kepada pengguna ketika hendak melakukan pencarian. Lalu *grid* menampilkan isi *table* secara otomatis berdasarkan kolom pada tabel *database*. Kemudian *list view* untuk menampilkan nilai pada *database* dengan dukungan bantuan seperti *Searching*, *Sorting*, dan *Filtering*. Kendo UI juga mendukung representasi *data source* untuk ditampilkan dalam visualisasi *chart*. Contoh visualisasi *chart* Kendo UI dapat dilihat pada Gambar 2.4.



Gambar 2.4 Visualisasi Chart Kendo UI

2.8 Point of Sale

Saat ini, *Point of Sale* kerap digunakan di berbagai negara untuk mendukung transaksi bisnis. *Point of Sale* merupakan sebuah sistem yang mengandung berbagai tugas vital atau krusial seperti transaksi, laporan, manajemen penjualan, dan lainnya (Kabir dan Han, 2016). Sebuah sistem *point of sale* harus mampu memanfaatkan data penjualan yang ada dengan menghasilkan beberapa jenis laporan penjualan seperti riwayat transaksi, produk terlaris, hingga penelusuran data penjualan dalam skala waktu tertentu. (Polanz, 2011). Selain itu, dengan adanya penarikan informasi dari data-data transaksi yang ada, dapat memberikan informasi untuk pengambilan kebijakan yang lebih akurat, seperti dalam hal penambahan stok produk, dimana dengan menganalisis *trend* terlebih dahulu dapat mengurangi kemungkinan adanya *clearance sales* atau keusangan produk (Harvey, 2013). Lebih lanjut, data yang ada dengan memanfaatkan *point of sale* dapat digunakan untuk meramalkan kebutuhan yang dapat menjadi informasi krusial di waktu yang akan datang (Zhu, 2013).

2.9 Data Mining

Data mining adalah sebuah disiplin yang mengekstraksi informasi dari sekumpulan data yang banyak. Tujuan dari adanya *data mining* ialah untuk memperluas pemahaman dan wawasan dari data yang ada sehingga mendukung untuk mengambil keputusan yang berkaitan (Cafaro, Epicoco dan Pulimeno, 2018). Contoh dari pemanfaatan *data mining* biasa diterapkan di dalam kehidupan. Salah satu contoh penerapan *data mining* yaitu *cross-market analysis* yaitu pemanfaatan *data mining* untuk mengetahui pola keterkaitan antara penjualan suatu produk dengan produk yang lain. Dengan informasi yang didapatkan dari *cross-market analysis* penjual dapat mengetahui produk mana saja yang memiliki keterkaitan dalam penjualan sehingga penjual dapat menambahkan persediaan barang-barang yang memiliki keterkaitan yang tinggi untuk memperoleh keuntungan yang lebih besar. Selain itu, pemanfaatan *data mining* juga dapat dilakukan untuk melakukan peramalan maupun prediksi. Dengan terdapatnya beberapa data yang relevan, memanfaatkan *data mining* dapat menghasilkan pengetahuan yang dapat digunakan di waktu yang akan datang.

2.9.1 Holt's Exponential Smoothing

Holt's Exponential Smoothing ialah salah satu jenis metode linear pada *Exponential Smoothing* (Holt, 2004). Keunggulan dari penggunaan metode ini ialah keberadaannya pada data time series yang memiliki alur atau tren yang bervariasi. Dengan metode ini, maka akan memuluskan tren data secara langsung dan memprediksi data asli. Karena fleksibilitas yang tinggi, metode ini digunakan secara luas (Li, 2013). Metode ini sendiri dapat digunakan untuk melakukan prediksi. Dalam bisnis *restaurant*, prediksi dari sisi pendapatan yang baik dapat mempermudah dan membantu pihak *restaurant* dalam menyusun target dan memperkirakan pendapatan yang akan diterima. *Holt's Exponential Smoothing*

memiliki dua dasar rumus *smoothing* dan sebuah rumus prediksi. Kedua rumus *smoothing* tersebut masing-masing untuk dua faktor *time series* yang dapat dilihat pada persamaan 2.1 - 2.5.

Inisialisasi:

$$S_1 = A_1 \quad (2.1)$$

$$T_1 = \frac{(y_2 - y_1) + (y_4 - y_3)}{2} \quad (2.2)$$

Rumus *Holt's Exponential Smoothing*:

$$F_{T+m} = S_T + T_T * m \quad (2.3)$$

$$T_t = \beta(S_t - S_{t-1}) + (1 - \beta)T_{t-1} \quad (2.4)$$

$$S_t = \alpha y_t + (1 - \alpha)(S_{t-1} + T_{t-1}) \quad (2.5)$$

Keterangan:

S_t	: <i>Exponential Smoothing</i>
T_t	: Tren
F_{T+m}	: Hasil Prediksi
S_1	: Data Peramalan Pertama
A_1	: Data Aktual Pertama
T_1	: Nilai Tren Pertama
Y_x	: Data Aktual ke x
α	: Konstanta pemulusan
β	: Konstanta untuk tren

Lebih lanjut, terdapat beberapa metode perhitungan yang biasa digunakan dalam menghitung kesalahan prediksi (*forecast error*). Perhitungan ini dapat digunakan untuk membandingkan beberapa model peramalan yang berbeda, dan juga dapat digunakan untuk mengawasi prediksi, dan untuk memastikan apakah prediksi itu berjalan dengan baik atau tidak. Salah satunya yakni ialah *Mean Absolute Percentage Error* (MAPE), yaitu metode yang dapat digunakan untuk mengukur tingkat akurasi. MAPE merupakan ukuran ketetapan relatif berdasarkan nilai absolut yang digunakan untuk mengetahui persentase penyimpangan hasil prediksi dengan data aktual. MAPE mengukur rata-rata *error absolute* sebagai persentase bukan pada setiap periode melainkan persentase dari rata-rata *error rate absolute* sejumlah periode data aktual.

2.9.2 *Apriori*

Algoritme *Apriori* memanfaatkan pengetahuan dari frekuensi atribut yang diketahui sebelumnya untuk memproses informasi selanjutnya. *Apriori* adalah algoritme yang digunakan untuk mendapatkan maupun melakukan pencarian *frequent itemset* menggunakan teknik asosiasi atau *association rule*. *Association Rules* sendiri merupakan teknik *data mining* dalam pencarian relasi asosiasi atau mengambil data tersembunyi dalam suatu kumpulan atau himpunan data yang besar. Dalam Algoritme *Apriori*, dilakukan penetapan nilai *minimum support*

sebagai nilai *threshold* untuk mendapatkan kandidat (Cafaro, Epicoco dan Pulimeno, 2018). Setelah proses seleksi pada tahapan pertama selesai, maka akan dilakukan pengkombinasian dengan produk lainnya yang juga melewati batas minimal. Pengujian pada algoritme *Apriori* salah satunya dilakukan dengan menganalisis pengaruh minimum *confidence* dan minimum *support* untuk menghasilkan *rule*. Selain itu, dapat juga menggunakan nilai *lift ratio* untuk melihat kuat atau tidaknya aturan asosiasi dengan membandingkan dengan nilai *benchmark confidence*. Dimana diasumsikan kejadian item dari *consequent* dalam suatu transaksi adalah independen dengan kejadian dari *antecedent* dari suatu asosiasi.

2.10 Pengujian Perangkat Lunak

Pengujian perangkat lunak digunakan untuk mendesain sekumpulan *test cases* yang memiliki kemungkinan tinggi dalam menemukan *error* yang terdapat dalam sistem yang dibangun (Pressman, 2010). Pengujian pada perangkat lunak bertujuan untuk menemukan *error* maupun hal yang tidak sepatasnya. Hal-hal tersebut dapat ditemukan pada konten, *function*, *usability*, *navigability*, *performance*, kapasitas dan keamanan. Teknik atau metode yang digunakan dalam kasus penelitian ini adalah *white-box-testing*, *black-box testing*, dan *integration testing*. Sedangkan dalam hal untuk memastikan kemudahan penggunaan, dilakukan *usability testing*.

2.10.1.1 Unit Testing

Pengujian unit adalah pengujian yang pertama kali dilakukan. Hal yang harus diuji dalam pengujian unit pada pemrograman berorientasi objek ialah pengujian pada level *class* atau *method*. Karakteristik utama dari pengujian unit adalah pengujian dilakukan pada level unit secara individu dan terisolasi dari komponen lainnya yang ada pada sistem, hal ini dilakukan untuk mencegah pengaruh dari komponen lainnya (Spillner, Linz dan Schaefer, 2014). Pada pengujian unit biasanya dilakukan menggunakan teknik *white box*. *White Box Testing* adalah salah satu cara untuk menguji suatu aplikasi atau software dengan cara melihat modul untuk dapat meneliti dan menganalisa kode dari program yang di buat ada yang salah atau tidak. Pressman (2010) menjelaskan bahwa *white-box testing*, disebut juga *glass-box testing*, menggunakan struktur kontrol, yang merupakan bagian dari *component level design*, untuk menghasilkan *test cases*. Dengan kata lain, pengujian *whitebox* hanya dapat didesain apabila *component-level design* (atau *source code*) sudah dibuat. *Test cases* yang dihasilkan dengan menggunakan metode *white-box testing*:

1. Menjamin bahwa seluruh *independent path* yang ada di dalam suatu *modul* dapat dilakukan paling tidak satu kali.
2. Mencoba seluruh *logical decision* dan memberikan kesesuaian hasil nilai *true* atau *false* berdasarkan kondisi yang diberikan.
3. Melakukan percobaan pada semua *loops* sesuai dengan batasan yang diberikan.

4. Melakukan percobaan pada struktur data internal untuk memastikan bahwa struktur tersebut valid.

White box testing didasarkan pada pengujian prosedur yang terjadi secara detail. Jalur-jalur logis (*logical paths*) yang dilalui pada perangkat lunak dan kolaborasi antar komponen diujikan berdasarkan sekumpulan kondisi (*conditions*) dan/atau perulangan-perulangan (*loops*) tertentu. Dalam pengujian sistem ini, akan digunakan *basis path testing*. Dengan menggunakan metode *basis path*, dapat menghasilkan pengukuran kompleksitas logis (*logical complexity*) dari rancangan secara prosedural. Kemudian, pengukuran tersebut dapat digunakan menentukan dasar dari kumpulan jalur yang dieksekusi.

2.10.1.2 Integration Testing

Tujuan dari pengujian integrasi adalah menjamin bahwa semua komponen atau unit yang berhubungan dapat berinteraksi dengan baik dan benar sesuai dengan kebutuhan (Spillner, Linz, dan Schaefer, 2014). Apabila menemukan kecacatan, adanya pengujian integrasi akan mengekspos kecacatan dari sisi cara berinteraksi antar komponen yang saling berhubungan. Pada *integration testing* terdapat dua strategi yang bisa digunakan yaitu *Top-down integration* dan *Bottom-up integration*. Strategi *Top-down integration*, pengujian akan dimulai dari komponen level atas (komponen utama) dan komponen level atas akan memanggil komponen level bawah. Apabila komponen level bawah belum selesai dibangun maka pengujian bisa memanfaatkan *test stub* yaitu sebuah komponen yang bisa menyimulasikan komponen level bawah yang belum selesai dibangun. Sedangkan untuk strategi *Bottom-up integration*, pengujian dimulai dari komponen level bawah menuju komponen level atas (komponen utama). Apabila komponen level atas belum selesai dibangun maka pengujian bisa menggunakan *test driver* dimana *test driver* ini digunakan untuk menyimulasikan komponen level atas yang belum selesai dibangun.

2.10.1.3 Validation Testing

Pengujian validasi adalah pengujian yang harus dilakukan sebelum sistem digunakan pada lingkungannya. Teknik *black-box testing* merupakan salah satu strategi pengujian yang digunakan untuk mengetahui kesesuaian fungsi tertentu dari suatu produk dengan tujuan awal didesainnya produk tersebut. *Black-box testing* merupakan sebuah pendekatan yang ditujukan untuk menguji sistem berdasarkan spesifikasi sistem tanpa harus mengakses atau melakukan uji terhadap kode program (Sommerville, 2011). *Black-box testing*, disebut juga sebagai *behavioural testing*, dikhususkan pada kebutuhan perangkat lunak. Pengujian *black-box* dilaksanakan untuk menemukan *error* ketika skrip dieksekusi. *Black-box testing* melakukan pencarian *error* dalam kategori (Pressman, 2010):

1. Fungsi yang salah atau fungsi yang tidak lengkap,
2. Antar muka yang *error*,
3. *Error* dalam melakukan akses basis data eksternal atau akses data struktur,
4. *Error* dalam *behavior* atau kinerja, dan

5. Error dalam inialisasi dan *termination*.

Black-box testing cenderung dilaksanakan pada langkah terakhir karena pengujian ini tidak mempedulikan struktur kontrol. *Black-box testing* dilakukan dengan membuat sekumpulan *test case* yang telah didesain untuk dapat menemukan *error* pada level validasi perangkat lunak (*software validation level*). Pada setiap *case* yang diberikan, diharapkan agar dapat menemukan *error* sebanyak mungkin dengan waktu dan usaha seminimal mungkin. Setiap *test case* dapat dibuat untuk dapat melakukan uji coba terhadap setiap fungsi yang dibuat. Kemudian akan di validasi apakah telah beroperasi sepenuhnya dan disaat yang bersamaan melakukan pencarian *error* yang mungkin terjadi.

2.10.1.4 Usability Testing

Usability Testing dilakukan untuk melakukan pengujian sistem dengan cara menguji sistem secara langsung kepada sejumlah responden untuk mengetahui tingkat kemudahan penggunaan sebuah sistem. Pengujian ini dilakukan setelah sistem sudah siap digunakan sehingga diperoleh penilaian berdasarkan hasil dan analisis responden yang dihitung berdasarkan metode yang digunakan. Dalam pengujian usability kali ini digunakan metode *System Usability Scale* (SUS). Metode SUS merupakan sebuah pengujian pada sisi pengguna yang menyediakan alat ukur yang “quick and dirty” dan dapat diandalkan (Brooke, 1986). Pada SUS, responden diberikan kesempatan untuk mengevaluasi sistem tanpa adanya pembekalan maupun diskusi terlebih dahulu.

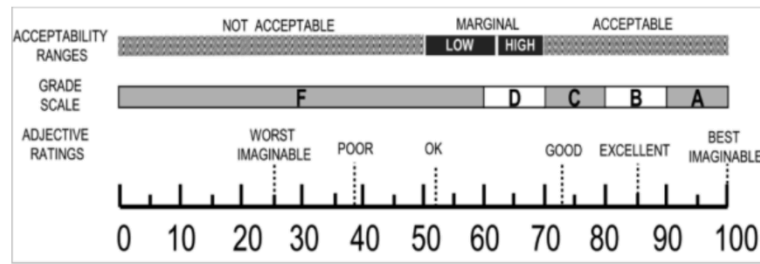
SUS terdiri dari 10 pertanyaan dengan menggunakan skala *likert* satu hingga lima. Keterangan dari setiap *likert* dapat dilihat pada Tabel 2.6. Pertanyaan nomor ganjil (1, 3, 5, 7, 9) merupakan pertanyaan yang mengarah kepada hal positif. Sedangkan pertanyaan nomor genap (2, 4, 6, 8, 10) merupakan pertanyaan dengan bernada negatif. Penilaian dilakukan dengan pengurangan satu nilai dari respon yang diberikan *user*, untuk pertanyaan ganjil dan lima dikurangi dengan respon yang diberikan *user* untuk pertanyaan genap. Setelah itu skor SUS didapat dengan cara mengkalikan total skor dengan 2.5.

Tabel 2.6 Keterangan skor skala *likert* tiap pernyataan SUS

No	Keterangan
1	Sangat Tidak Setuju
2	Tidak Setuju
3	Netral
4	Setuju
5	Sangat Setuju

Skor akhir SUS akan berada pada kisaran 0-100. Kriteria skor akhir dapat dilihat pada Gambar 2.5 dimana berdasarkan skor akhir SUS tersebut dapat diketahui seberapa tinggi tingkat usability dan akseptabilitas (acceptable) dari sistem yang

dikembangkan. Penilaiannya berdasarkan tiga kategori yaitu *Not Acceptable* dengan rentang skor SUS 0 - 50.9, *Marginal* 51 - 70.9, dan *Acceptable* 71 - 100.

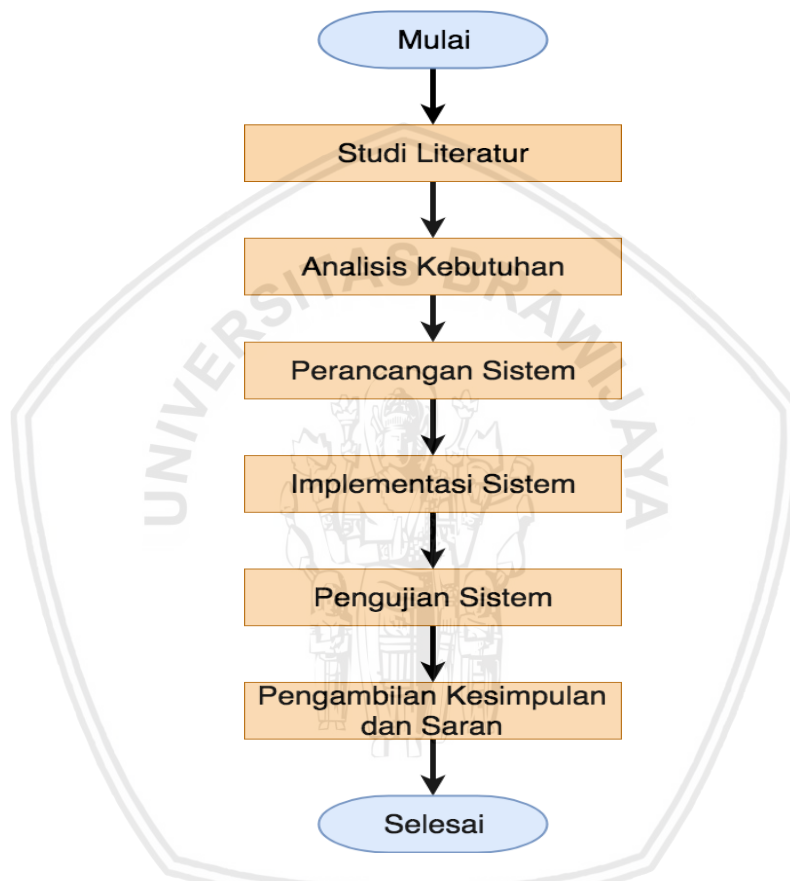


Gambar 2.5 Rating dan skala konversi skor rata-rata SUS



BAB 3 METODOLOGI

Pada penelitian ini terdapat enam tahapan metodologi yang dilakukan. Metodologi yang digunakan untuk mengerjakan penelitian ini meliputi studi literatur, analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian sistem, Pengambilan kesimpulan dan saran, hingga selesai. Alur penelitian dapat disimak pada Gambar 3.1.



Gambar 3.1 Diagram Alur Penelitian

3.1 Studi Pustaka

Referensi utama yang digunakan pada penelitian ini berkaitan dengan studi literatur. Studi literatur dilakukan dengan mencari relevansi pada setiap dasar teori dimana memiliki landasan baik dari jurnal yang telah ada, buku, website dan sumber lainnya. Sebelumnya, dasar teori telah dibahas pada bab 2. Teori dan pustaka yang berkaitan mengenai skripsi ini diantaranya ialah rekayasa perangkat lunak, *software development life cycle*, model *Waterfall*, algoritme *Holt Exponential Smoothing*, algoritme Apriori, hingga teknologi pengembangan sistem.

3.2 Analisis Kebutuhan

Dalam melakukan analisis kebutuhan, dilakukan komunikasi dengan menggali kebutuhan kepada narasumber pada studi kasus. Sehingga, saat hendak melakukan perancangan dan implementasi sistem, setiap kebutuhan calon pengguna telah terdefiniskan dengan baik dan akurat. Analisis kebutuhan yang baik akan menghasilkan sistem yang baik pula. Selain itu, elisitasi kebutuhan yang benar tentunya akan meminimalisir penambahan kebutuhan demi kebutuhan saat pengerjaan sistem. Sedangkan analisis kebutuhan yang tidak baik dapat mengakibatkan kesalahan mendasar dimana nantinya dapat mempengaruhi sistem yang dihasilkan sendiri. Proses analisis kebutuhan sistem sendiri meliputi:

1. Elisitasi Kebutuhan

Wawancara menjadi teknik elisitasi yang digunakan pada penelitian ini. Dengan melakukan wawancara kepada *owner* Edd's Waffle, akan dilakukan penggalian informasi dalam kelompok pertanyaan sehingga setiap kebutuhan dapat dijabarkan secara rinci. Hasil yang di dapat dari elisitasi kebutuhan dapat meliputi gambaran umum mengenai sistem yang akan dibangun, fitur-fitur yang tersedia, serta pihak-pihak yang terlibat. Kemudian, hasilnya diajukan kepada klien untuk di validasi dan di verifikasi sesuai kebutuhan dari Edd's Waffle.

2. Spesifikasi kebutuhan

Pada proses ini, kebutuhan yang sudah didapatkan lebih di spesifikkan. Dalam proses ini juga melakukan identifikasi aktor yang akan menggunakan sistem ini dan juga kebutuhan yang sudah didapatkan ke dalam dua jenis kebutuhan yaitu fungsional dan non-fungsional untuk mengetahui secara rinci aktifitas apa saja yang bisa dilakukan oleh pengguna maupun sistem dan apa saja batasan-batasan sistem yang akan dibangun.

3. Manajemen Kebutuhan

Pada proses ini dilakukan kontrol terhadap kebutuhan yang sudah didefinisikan dan memberikan kode pada kebutuhan dalam menulis kebutuhan sistem dengan kode POSEW-F-01 untuk kebutuhan fungsional dan POSEW-NF-01 untuk kebutuhan non fungsional.

4. Pemodelan Kebutuhan

Setelah seluruh kebutuhan telah terdefinisi dengan benar, akan dilakukan identifikasi aktor-aktor yang terlibat didalam sistem, *use case diagram* dan *use case scenario*. Pada *use case diagram* akan menggambarkan hubungan aktor dengan *use case* dan hubungan suatu *use case* dengan *use case* lainnya. Setiap *use case* pada *use case diagram* akan paparkan lebih rinci pada *use case scenario* mengenai bagaimana interaksi aktor dengan sistem dan apa saja alternatif yang terdapat pada sistem.

3.3 Perancangan Sistem

Perancangan sistem merupakan rancangan dari setiap elisitasi dan analisis kebutuhan yang dimodelkan dalam sistem. Perancangan sistem nantinya akan menjadi acuan dalam melakukan implementasi hingga pengujian sistem.

1. Perancangan Arsitektur

Dalam perancangan arsitektur ini akan dilakukan pemodelan dengan menggunakan diagram UML seperti sequence diagram dan class diagram. *Sequence diagram* berguna untuk menggambarkan interaksi antara objek dan pesan yang dikirim atau yang diterima oleh objek-objek tersebut. pada penelitian ini akan dipaparkan tiga *sequence diagram* yang merupakan tiga fungsionalitas utama pada sistem yang dikembangkan.

2. Perancangan Komponen

Dalam perancangan komponen ini akan dituliskan beberapa sampel algoritme-algoritme utama yang diambil dari *controller*. Algoritma ini akan ditulis dalam bentuk *pseudocode*. Nantinya, perancangan komponen ini akan dijadikan sebagai acuan dalam melakukan implementasi kode program.

3. Perancangan Data

Dalam perancangan data memuat rancangan tabel *database* yang menjelaskan proses normalisasi dari ERD (*Entity Relational Diagram*). Perancangan data ini akan menjadi dasar dalam implementasi *database* pada sistem.

4. Perancangan Antarmuka

Perancangan antarmuka dari sistem yang akan dibangun ini terdiri dari tata letak komponen yang harus disediakan oleh sistem berdasarkan kebutuhan sistem. Dalam perancangan antarmuka ini akan dituliskan beberapa sampel antarmuka utama berdasarkan level pengguna. Perancangan antarmuka ini akan menjadi dasar dari implementasi antarmuka sistem.

3.4 Implementasi Sistem

Tahapan implementasi adalah tahapan yang dilakukan setelah tahapan perancangan sistem selesai dan setiap kebutuhan yang telah didefinisikan disetujui oleh pengguna sistem. Tahapan ini adalah tahapan untuk mengimplementasikan perangkat lunak yang akan dibangun sesuai dengan perancangan sistem.

Proses yang ada dalam implementasi sistem ini antara lain:

1. Implementasi sistem dilakukan dengan mengacu pada spesifikasi sistem yang digunakan untuk membangun aplikasi ini.
2. Implementasi kode program mengacu pada perancangan komponen yang memuat algoritme-algoritme pada perancangan sistem. Sistem akan

dibangun dengan bahasa pemrograman C# menggunakan ASP .NET MVC *framework*. Implementasi sistem sendiri ditujukan untuk *platform web*.

3. Implementasi basis data di implementasikan menggunakan basis data *SQL Server*.
4. Implementasi antarmuka, implementasi ini dibuat berdasarkan perancangan antarmuka pada perancangan sistem.

3.5 Pengujian Sistem

Dalam melakukan evaluasi terhadap produk perangkat lunak, dilakukan pengujian perangkat lunak untuk meminimalisir kemungkinan adanya kecacatan maupun kesalahan. Sehingga, diharapkan dihasilkan sistem yang memiliki kemampuan yang baik dan dapat diterima oleh calon pengguna. Terdapat empat pengujian yang dilakukan pada sistem, yaitu:

1. *Unit Testing*

Unit testing merupakan pengujian terkecil pada perangkat lunak yang dapat dilakukan dengan menggunakan teknik *White box* jenis *basis path testing*. Pengujian yang dilakukan dengan menguji setiap unit seperti *class*, komponen atau objek dari perangkat lunak yang telah didefinisikan pada perancangan. Teknik basis *path* digunakan untuk menguji kode program berdasarkan algoritma yang terdapat pada setiap fungsi pada setiap kelas yang pengujiannya menggunakan tiga sample uji, teknik basis *path* merupakan pengujian *white box* pada pengujian ini cara pengujian dilakukan dengan melihat kedalam unit atau modul untuk melihat kode – kode program yang ada dan menganalisis apakah ada kesalahan atau tidak.

2. *Integration Testing*

Pengujian integrasi (*Integration Testing*) adalah pengujian yang difokuskan pada gabungan unit-unit atau modul-modul yang membentuk kesatuan fungsional yang didasarkan pada informasi dari deskripsi perancangan awal perangkat lunak dengan tujuan untuk menemukan kesalahan antarmuka antar modul. Pada pengujian ini, dilakukan pengujian guna memastikan bahwa semua komponen atau unit yang berhubungan dapat berinteraksi dengan baik dan benar sesuai dengan kebutuhan.

3. *Validation Testing*

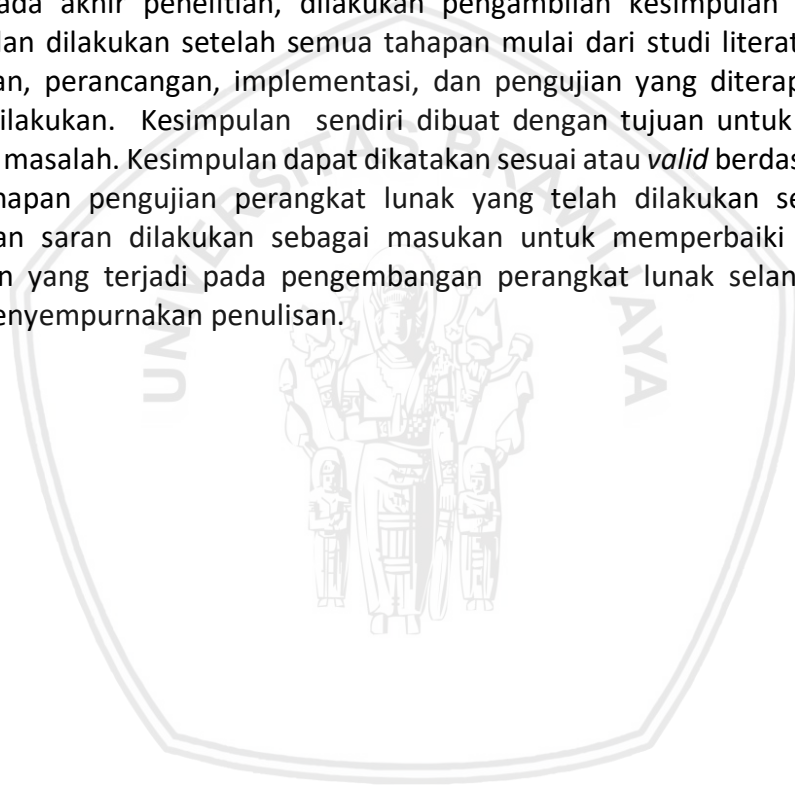
Pada pengujian ini dilakukan verifikasi dan validasi bahwa semua kebutuhan pada perangkat lunak sudah dibuat dengan benar. Strategi dalam melakukan pengujian validasi adalah dengan cara melakukan serangkaian *Black Box Testing*. Teknik pengujian ini dilakukan dengan cara menjalankan atau dengan mengeksekusi sistem apakah sudah sesuai dengan proses bisnis yang diinginkan. Secara singkat, pada pengujian ini dilakukan pengecekan bahwa semua fungsi pada perangkat lunak sudah dibuat dengan benar. Sehingga, apabila pengujian ini selesai dilakukan, maka akan dapat diterima oleh calon pengguna.

4. Usability Testing

Usability testing dilakukan untuk memastikan bahwa sistem yang dibangun sudah sesuai dengan dapat digunakan dengan mudah, mudah dipahami oleh *user* dan mudah untuk dioperasikan oleh *user* nantinya. Pengujian ini dilakukan ketika sistem sudah siap digunakan dan digunakan oleh pengguna sehingga pengguna dapat merasakan akan kemudahan menggunakan sistem dan juga sistem mudah dipelajari dan dipahami. Pengujian ini nantinya akan dilakukan dengan memanfaatkan metode *System Usability Scale (SUS)* dengan cara memberikan kuesioner kepada responden.

3.6 Pengambilan Kesimpulan dan Saran

Pada akhir penelitian, dilakukan pengambilan kesimpulan dan saran. Kesimpulan dilakukan setelah semua tahapan mulai dari studi literatur, analisis kebutuhan, perancangan, implementasi, dan pengujian yang diterapkan sudah selesai dilakukan. Kesimpulan sendiri dibuat dengan tujuan untuk menjawab rumusan masalah. Kesimpulan dapat dikatakan sesuai atau *valid* berdasarkan hasil pada tahapan pengujian perangkat lunak yang telah dilakukan sebelumnya. Sedangkan saran dilakukan sebagai masukan untuk memperbaiki kesalahan-kesalahan yang terjadi pada pengembangan perangkat lunak selanjutnya dan untuk menyempurnakan penulisan.



BAB 4 ANALISIS KEBUTUHAN

Analisis kebutuhan merupakan tahapan pertama yang dilakukan dalam proses pengembangan sistem. Melalui tahapan ini, setiap kebutuhan akan didefinisikan baik meliputi aktor yang berperan hingga bagaimana kebutuhan tersebut dimodelkan. Pada analisis kebutuhan, pemodelan diwujudkan dalam *use case diagram* dan *use case scenario*. Kedua pemodelan ini bertujuan untuk meningkatkan pemahaman akan setiap kebutuhan yang terdapat pada sistem. *Use case diagram* mengidentifikasi aktor-aktor yang terlibat pada sebuah interaksi dan nama dari tipe interaksi tersebut. Sedangkan *use case scenario* berfungsi untuk mengidentifikasi skenario pada suatu sistem (Sommerville, 2011).

4.1 Gambaran Umum Sistem

Sistem *point of sale* pada *restaurant Edd's Waffle* ditujukan untuk mengorganisir penjualan baik dari sisi transaksi, laporan, hingga berbagai hal yang bertujuan untuk meningkatkan performa penjualan. Sistem ini dibangun pada basis *platform web*. Secara gambaran umum, sistem ini dikembangkan ke dalam 6 bagian utama, yaitu:

1. Transaksi Penjualan

Dalam bagian transaksi penjualan, *cashier* dapat melakukan proses menambah transaksi penjualan, menghapus produk yang terdapat dalam suatu transaksi yang sedang di proses, menambahkan diskon, dan mencetak transaksi penjualan.

2. Pengelolaan Produk

Bagian pengelolaan produk hanya dapat diakses oleh *supervisor*. Dalam bagian pengelolaan produk, *supervisor* dapat menambahkan produk, mengubah produk yang telah ada, menghapus produk yang telah ada, dan melihat produk yang ada saat ini. Produk sendiri di deskripsikan dengan kode produk, nama produk, kategori produk, dan harga produk. Selain itu, *supervisor* juga dapat mengelola kategori produk yang nantinya akan mengklasifikasikan produk-produk yang tersedia.

3. Riwayat Transaksi

Bagian riwayat transaksi hanya dapat diakses oleh *supervisor*. Melalui riwayat transaksi, *supervisor* dapat melihat daftar transaksi dalam rentang waktu tertentu, dengan informasi yang disajikan meliputi rincian dari transaksi yang bersangkutan.

4. *Monitoring* Performa Penjualan

Bagian *monitoring* penjualan hanya dapat diakses oleh *supervisor*. Melalui bagian ini, *supervisor* dapat memantau perkembangan penjualan dalam visualisasi grafik. Informasi yang disajikan dalam bagian *monitoring* performa penjualan meliputi perkembangan transaksi dalam kurun waktu

mingguan maupun bulanan yang meliputi jumlah, rata-rata, nilai maksimum dan minimum, hingga total pendapatan. Selain itu, *supervisor* juga dapat melihat sepuluh hari dengan jumlah pendapatan tertinggi maupun terendah.

5. Prediksi Pendapatan

Supervisor berkemungkinan untuk melihat prediksi pendapatan mingguan untuk empat pekan kedepan. Prediksi ini di dapat dengan memanfaatkan algoritme *Holt Exponential Smoothing* yang didasarkan pada data-data historis pendapatan, sehingga dapat membantu pihak Edd's Waffle untuk menyusun target pendapatan di waktu yang akan datang.

6. Trend Penjualan

Bagian ini hanya dapat di akses oleh *supervisor*. Melalui *trend* penjualan, *supervisor* dapat melihat daftar produk dengan jumlah penjualan tertinggi. Hal ini ditujukan untuk mengoptimalkan pemanfaatan produk dari Edd's Waffle agar produk-produk yang ada dapat di siapkan sebagaimana mestinya. Selain itu, dengan memanfaatkan algoritme *Apriori*, memungkinkan *supervisor* untuk menemukan produk potensial yang dapat dikombinasikan sehingga menambah referensi dalam membuat paket makanan.

4.2 Identifikasi Aktor

Aktor ialah pihak yang terlibat dalam sistem ini. Terdapat empat aktor yang terlibat pada sistem ini yaitu *guest*, *supervisor*, *cashier* dan *administrator*. Deskripsi dari masing-masing aktor atau identifikasi aktor dapat dilihat pada Tabel 4.1.

Tabel 4.1 Identifikasi Aktor

No.	Aktor	Deskripsi
1.	<i>Guest</i>	<i>Guest</i> adalah segmen pengguna yang belum login atau pengunjung pada sistem.
2.	<i>Supervisor</i>	<i>Supervisor</i> adalah segmen pengguna yang berfungsi untuk mengawasi dan memperhatikan performa penjualan serta mengambil kebijakan-kebijakan untuk penjualan. Kebijakan yang dapat diambil <i>supervisor</i> ialah seperti mengelola produk baik dari sisi pembaruan maupun perubahan harga.
3.	<i>Cashier</i>	<i>Cashier</i> adalah segmen pengguna yang berfungsi untuk melayani proses transaksi yang ada di Edd's Waffle.
4.	<i>Administrator</i>	<i>Administrator</i> adalah segmen pengguna yang berfungsi untuk melakukan pengaturan dalam hal akun yang terdaftar pada sistem hingga nilai parameter untuk prediksi pendapatan dan kombinasi produk.

4.3 Kebutuhan Fungsional Sistem

Kebutuhan fungsional mendefinisikan layanan-layanan yang harus terdapat pada sistem. Kebutuhan fungsional di dapat melalui hasil analisis yang telah dilakukan setelah adanya wawancara dengan *owner* dari Edd's Waffle. Kebutuhan fungsional harus di validasi dan verifikasi untuk memastikan bahwa tidak ada kebutuhan yang cenderung ambigu dan dapat diidentifikasi. Setiap kebutuhan fungsional pada sistem akan di definisikan dalam kode POSEW-F-XX. POSEW merupakan singkatan dari *Point of Sale* Edd's Waffle, sedangkan F adalah singkatan dari fungsional, lalu XX mendefinisikan nomor kebutuhan fungsional. Daftar kebutuhan fungsional pada penelitian ini bisa dipaparkan dalam Tabel 4.2 - 4.10.

Tabel 4.2 Daftar Kebutuhan Fungsional

No.	Kode	Kebutuhan	Use Case
1.	POSEW-F-01	Sistem harus dapat melakukan autentikasi terhadap <i>Guest</i> sebagai <i>Cashier</i> , <i>Supervisor</i> , atau <i>Admin</i> .	<i>Login</i>
	POSEW-F-01-01	Sistem harus menyediakan masukan data untuk <i>username</i> dan <i>password</i> .	
2.	POSEW-F-02	Sistem harus dapat mengeluarkan <i>Supervisor</i> dan <i>Cashier</i> dari sistem sebagai <i>Guest</i> .	<i>Logout</i>
3.	POSEW-F-03	Sistem harus dapat menambahkan transaksi.	Tambah Transaksi
	POSEW-F-03-01	Sistem harus dapat menambahkan transaksi baru dengan masukan data yang dibutuhkan adalah daftar pesanan (meliputi kode produk, kuantitas produk, nama produk, harga produk, dan total harga produk), kode transaksi, diskon, metode pembayaran, total harga, jumlah uang tunai, dan jumlah uang kembalian.	
	POSEW-F-03-02	Sistem harus dapat menampilkan kode transaksi terkini. Kode transaksi memiliki format: "1+yy:mm:dd:+nomor transaksi" dimana yy merepresentasikan tahun saat ini, mm merepresentasikan bulan saat ini, dan dd merepresentasikan tanggal saat ini. Sedangkan untuk nomor transaksi akan selalu bertambah sebanyak 1 dari transaksi terakhir yang dilakukan.	

Tabel 4.3 Daftar Kebutuhan Fungsional (Lanjutan)

No.	Kode	Kebutuhan	Use Case
3.	POSEW-F-03-03	Sistem harus dapat mencetak transaksi yang ditambahkan dengan informasi yang disajikan adalah nama <i>restaurant</i> , alamat <i>restaurant</i> , nomor telfon <i>restaurant</i> , tanggal dan waktu, kode transaksi, daftar pesanan (nama produk, kuantitas produk, harga produk, dan total harga produk), metode pembayaran, diskon, total harga, jumlah bayar tunai, dan jumlah kembalian dengan keluaran file dalam format pdf.	Tambah Transaksi
	POSEW-F-03-04	Sistem hanya mengizinkan masukan angka untuk kuantitas produk, harga produk, diskon, dan jumlah uang tunai.	
4.	POSEW-F-04	Sistem harus dapat menghapus item yang ada dalam daftar pesanan pada transaksi aktif.	Hapus <i>Item</i> Transaksi
5.	POSEW-F-05	Sistem harus dapat memulai ulang transaksi.	Reset Transaksi
	POSEW-F-05-01	Sistem harus dapat memulai ulang transaksi dengan mengosongkan setiap <i>field</i> dan tabel yang telah terisi.	
6.	POSEW-F-06	Sistem harus dapat menampilkan referensi produk.	Lihat Referensi Produk
	POSEW-F-06-01	Sistem harus dapat menampilkan referensi produk dengan informasi yang disajikan adalah kode produk dan nama produk.	
7.	POSEW-F-07	Sistem harus dapat mencari referensi produk.	Cari Referensi Produk
	POSEW-F-07-01	Sistem harus dapat mencari referensi produk berdasarkan kode produk dan nama produk.	
8.	POSEW-F-08	Sistem harus dapat menampilkan jumlah total transaksi per pekan.	Lihat Total Transaksi Per Pekan
	POSEW-F-08-01	Sistem harus dapat menampilkan jumlah total transaksi yang telah terjadi dengan skala waktu setiap pekan dalam visualisasi plot pada grafik <i>line chart</i> .	
9.	POSEW-F-09	Sistem harus dapat menampilkan jumlah total transaksi per bulan.	Lihat Total Transaksi Per Bulan

Tabel 4.4 Daftar Kebutuhan Fungsional (Lanjutan)

No.	Kode	Kebutuhan	Use Case
9.	POSEW-F-09-01	Sistem harus dapat menampilkan jumlah total transaksi yang telah terjadi dengan skala waktu setiap bulan dalam visualisasi plot pada grafik <i>line chart</i> .	Lihat Total Transaksi Per Bulan
10.	POSEW-F-10	Sistem harus dapat menampilkan total pendapatan per pekan dari transaksi yang telah terjadi.	Lihat Total Pendapatan Transaksi Per Pekan
	POSEW-F-10-01	Sistem harus dapat menampilkan total total pendapatan dari transaksi yang telah terjadi dengan skala waktu setiap pekan dalam visualisasi plot pada grafik <i>line chart</i> .	
11.	POSEW-F-11	Sistem harus dapat menampilkan total pendapatan per bulan dari transaksi yang telah terjadi	Lihat Total Pendapatan Transaksi Per Bulan
	POSEW-F-11-01	Sistem harus dapat menampilkan total pendapatan dari transaksi yang telah terjadi dengan skala waktu setiap bulan dalam visualisasi plot pada grafik <i>line chart</i> .	
12.	POSEW-F-12	Sistem harus dapat menampilkan jumlah pendapatan tertinggi per pekan dari transaksi yang telah terjadi.	Lihat Pendapatan Transaksi Tertinggi Per Pekan
	POSEW-F-12-01	Sistem harus dapat menampilkan jumlah pendapatan tertinggi dari transaksi yang telah terjadi dengan skala waktu setiap pekan dalam visualisasi plot pada grafik <i>line chart</i> .	
13.	POSEW-F-13	Sistem harus dapat menampilkan jumlah pendapatan tertinggi per bulan dari transaksi yang telah terjadi.	Lihat Pendapatan Transaksi Tertinggi Per Bulan
	POSEW-F-13-01	Sistem harus dapat menampilkan jumlah pendapatan tertinggi dari transaksi yang telah terjadi dengan skala waktu setiap bulan dalam visualisasi plot pada grafik <i>line chart</i> .	
14.	POSEW-F-14	Sistem harus dapat menampilkan jumlah pendapatan terendah per pekan dari transaksi yang telah terjadi.	Lihat Pendapatan Transaksi Terendah Per Pekan
	POSEW-F-14-01	Sistem harus dapat menampilkan jumlah pendapatan terendah dari transaksi yang telah terjadi dengan skala waktu setiap pekan dalam visualisasi plot pada grafik <i>line chart</i> .	

Tabel 4.5 Daftar Kebutuhan Fungsional (Lanjutan)

No.	Kode	Kebutuhan	Use Case
15.	POSEW-F-15	Sistem harus dapat menampilkan jumlah pendapatan terendah per pekan dari transaksi yang telah terjadi.	Lihat Pendapatan Transaksi Terendah Per Bulan
	POSEW-F-15-01	Sistem harus dapat menampilkan jumlah pendapatan terendah dari transaksi yang telah terjadi dengan skala waktu setiap bulan dalam visualisasi plot pada grafik <i>line chart</i> .	
16.	POSEW-F-16	Sistem harus dapat menampilkan rata-rata pendapatan per pekan dari transaksi yang telah terjadi.	Lihat Rata Rata Pendapatan Transaksi Per Pekan
	POSEW-F-16-01	Sistem harus dapat menampilkan rata-rata pendapatan dari transaksi yang telah terjadi dengan skala waktu setiap pekan dalam visualisasi plot pada grafik <i>line chart</i> .	
17.	POSEW-F-17	Sistem harus dapat menampilkan rata-rata pendapatan per bulan dari transaksi yang telah terjadi.	Lihat Rata Rata Pendapatan Transaksi Per Bulan
	POSEW-F-17-01	Sistem harus dapat menampilkan rata-rata pendapatan dari transaksi yang telah terjadi dengan skala waktu setiap bulan dalam visualisasi plot pada grafik <i>line chart</i> .	
18.	POSEW-F-18	Sistem harus dapat menampilkan prediksi pendapatan.	Lihat Prediksi Pendapatan
	POSEW-F-18-01	Sistem harus dapat menampilkan prediksi pendapatan dengan skala waktu 4 pekan yang akan datang setelah transaksi terakhir dalam visualisasi plot pada grafik <i>line chart</i> .	
19.	POSEW-F-19	Sistem harus dapat menampilkan 10 hari dengan pendapatan tertinggi.	Lihat 10 Hari Dengan Pendapatan Tertinggi
	POSEW-F-19-01	Sistem harus dapat menampilkan 10 hari dengan pendapatan tertinggi dengan informasi yang disajikan adalah tanggal dan total pendapatan dari transaksi yang terjadi pada tanggal tersebut dalam visualisasi plot pada grafik <i>area chart</i> .	
20.	POSEW-F-20	Sistem harus dapat menampilkan 10 hari dengan pendapatan terendah.	Lihat 10 Hari Dengan Pendapatan Terendah

Tabel 4.6 Daftar Kebutuhan Fungsional (Lanjutan)

No.	Kode	Kebutuhan	Use Case
20.	POSEW-F-20-01	Sistem harus dapat menampilkan 10 hari dengan pendapatan terendah dengan informasi yang disajikan adalah tanggal dan total pendapatan dari transaksi yang terjadi pada tanggal tersebut dalam visualisasi plot pada grafik <i>area chart</i> .	Lihat 10 Hari Dengan Pendapatan Terendah
21.	POSEW-F-21	Sistem harus dapat menampilkan riwayat transaksi.	Lihat Riwayat Transaksi
	POSEW-F-21-01	Sistem harus dapat menampilkan riwayat transaksi dengan informasi yang disajikan adalah kode transaksi, tanggal transaksi, metode pembayaran transaksi, diskon pada transaksi, total biaya transaksi, total transaksi, dan total pendapatan.	
22.	POSEW-F-22	Sistem harus dapat menampilkan rincian dari riwayat transaksi.	Lihat Rincian Riwayat Transaksi
	POSEW-F-22-01	Sistem harus dapat menampilkan rincian dari riwayat transaksi dengan informasi yang disajikan adalah daftar kode produk, nama produk, jumlah produk, dan total harga produk pada transaksi yang bersangkutan.	
23.	POSEW-F-23	Sistem harus dapat menghapus transaksi yang telah terjadi.	Hapus Transaksi
24.	POSEW-F-24	Sistem harus dapat mencari transaksi dari sekumpulan riwayat transaksi.	Cari Transaksi
	POSEW-F-24-01	Sistem harus dapat mencari transaksi dari sekumpulan riwayat transaksi berdasarkan kode transaksi, tanggal transaksi, metode pembayaran, diskon transaksi, dan total harga.	
	POSEW-F-24-02	Sistem harus dapat mencari transaksi dari sekumpulan riwayat transaksi berdasarkan rentang minimal dan maksimal kode transaksi, tanggal transaksi, diskon transaksi, dan total harga.	
25.	POSEW-F-25	Sistem harus dapat mengurutkan transaksi dari sekumpulan riwayat transaksi baik dimulai dari posisi terendah maupun tertinggi.	Urut Transaksi
	POSEW-F-25-01	Sistem harus dapat mengurutkan transaksi dari sekumpulan riwayat transaksi berdasarkan kode transaksi, tanggal transaksi, metode pembayaran, dan total harga.	

Tabel 4.7 Daftar Kebutuhan Fungsional (Lanjutan)

No.	Kode	Kebutuhan	Use Case
26.	POSEW-F-26	Sistem harus dapat menampilkan kombinasi produk yang bertemu dari transaksi yang telah terjadi.	Lihat Kombinasi Produk
	POSEW-F-26-01	Sistem harus dapat menampilkan kombinasi produk yang bertemu dari transaksi yang telah terjadi dengan informasi yang disajikan adalah nama produk yang di kombinasikan dan jumlah kombinasi produk tersebut bertemu dalam satu transaksi.	
27.	POSEW-F-27	Sistem harus dapat mencari kombinasi produk.	Cari Kombinasi Produk
28.	POSEW-F-28	Sistem harus dapat mengurutkan kombinasi produk baik dimulai dari posisi terendah maupun tertinggi.	Urut Kombinasi Produk
	POSEW-F-28-01	Sistem harus dapat mengurutkan kombinasi produk berdasarkan kombinasi produk dan jumlah kombinasi produk bertemu dalam satu transaksi baik dimulai dari posisi terendah maupun tertinggi.	
29.	POSEW-F-29	Sistem harus dapat menampilkan jumlah produk yang telah terjual.	Lihat Jumlah Produk Terjual
	POSEW-F-29-01	Sistem harus dapat menampilkan jumlah produk yang telah terjual dengan informasi yang disajikan adalah nama produk dan jumlah produk terjual.	
30.	POSEW-F-30	Sistem harus dapat mencari jumlah produk terjual berdasarkan nama produk.	Cari Jumlah Produk Terjual
31.	POSEW-F-31	Sistem harus dapat mengurutkan jumlah produk terjual baik dimulai dari posisi terendah maupun tertinggi.	Urut Jumlah Produk Terjual
	POSEW-F-31-01	Sistem harus dapat mengurutkan jumlah produk terjual berdasarkan total produk telah terjual baik dimulai dari posisi terendah maupun tertinggi.	
32.	POSEW-F-32	Sistem harus dapat menampilkan daftar produk yang telah terdaftar.	Lihat Daftar Produk
	POSEW-F-32-01	Sistem harus dapat menampilkan daftar produk yang telah terdaftar dengan informasi yang disajikan adalah kode produk, nama produk, kategori produk, dan harga produk.	

Tabel 4.8 Daftar Kebutuhan Fungsional (Lanjutan)

No.	Kode	Kebutuhan	Use Case
33.	POSEW-F-33	Sistem harus dapat mencari produk.	Cari Produk
	POSEW-F-33-01	Sistem harus dapat mencari produk berdasarkan kode produk, nama produk, kategori produk. Dan harga produk.	
34.	POSEW-F-34	Sistem harus dapat menambahkan produk.	Tambah Produk
	POSEW-F-34-01	Sistem harus dapat menambahkan produk dengan masukan data yang dibutuhkan adalah kode produk, nama produk, kategori produk, dan harga produk.	
	POSEW-F-34-02	Sistem hanya mengizinkan masukan angka untuk harga produk.	
35.	POSEW-F-35	Sistem harus dapat menghapus produk yang telah terdaftar.	Hapus Produk
36.	POSEW-F-36	Sistem harus dapat menampilkan daftar kategori yang telah terdaftar.	Lihat Daftar Kategori
37.	POSEW-F-37	Sistem harus dapat mencari kategori pada sekumpulan kategori yang telah terdaftar.	Cari Kategori
38.	POSEW-F-38	Sistem harus dapat menambahkan kategori.	Tambah Kategori
	POSEW-F-38-01	Sistem harus dapat menambahkan kategori dengan data yang dibutuhkan adalah nama kategori.	
39.	POSEW-F-39	Sistem harus dapat menghapus kategori yang telah terdaftar.	Hapus Kategori
40.	POSEW-F-40	Sistem harus dapat menampilkan daftar akun yang telah terdaftar.	Lihat Daftar Akun
	POSEW-F-40-01	Sistem harus dapat menampilkan daftar akun yang telah terdaftar dengan informasi yang disajikan adalah <i>username</i> dan <i>role</i> .	
41.	POSEW-F-41	Sistem harus dapat mencari akun pada sekumpulan akun yang telah terdaftar.	Cari Akun
	POSEW-F-41-01	Sistem harus dapat mencari akun pada sekumpulan akun yang telah terdaftar berdasarkan <i>username</i> dan <i>role</i> .	

Tabel 4.9 Daftar Kebutuhan Fungsional (Lanjutan)

No.	Kode	Kebutuhan	Use Case
42.	POSEW-F-42	Sistem harus dapat menambahkan akun.	Tambah Akun
	POSEW-F-42-01	Sistem harus dapat menambahkan akun dengan masukan data yang dibutuhkan adalah <i>username</i> , <i>password</i> , dan <i>role</i> .	
43.	POSEW-F-43	Sistem harus dapat menghapus akun yang telah terdaftar.	Hapus Akun
44.	POSEW-F-44	Sistem harus dapat menyediakan pengaturan untuk menetapkan nilai parameter untuk memprediksi pendapatan.	Atur Nilai Parameter Prediksi
	POSEW-F-44-01	Sistem harus dapat menyediakan pengaturan untuk menetapkan nilai <i>alpha</i> dan <i>beta</i> untuk memprediksi pendapatan dengan rentang antara 0,1 hingga 1.	
45.	POSEW-F-45	Sistem harus dapat menampilkan informasi mengenai prediksi pendapatan.	Lihat Informasi Prediksi Pendapatan
	POSEW-F-45-01	Sistem harus dapat menampilkan informasi mengenai prediksi pendapatan dengan informasi yang disajikan adalah nilai <i>alpha</i> dan <i>beta</i> .	
	POSEW-F-45-02	Sistem harus dapat menampilkan informasi mengenai prediksi pendapatan dengan informasi yang disajikan adalah visualisasi prediksi pendapatan untuk 4 pekan yang akan datang dalam visualisasi plot pada grafik <i>area chart</i> .	
46.	POSEW-F-46	Sistem harus dapat menyediakan pengaturan untuk menetapkan nilai parameter yaitu nilai <i>minimum support</i> .	Atur Nilai Parameter Produk Potensial
	POSEW-F-46-01	Sistem harus dapat menyediakan pengaturan untuk menetapkan nilai parameter yaitu nilai <i>minimum support</i> dengan rentang persen antara 1 hingga 100.	
47.	POSEW-F-47	Sistem harus dapat menampilkan daftar produk potensial.	Lihat Daftar Produk Potensial
	POSEW-F-47-01	Sistem harus dapat menampilkan daftar produk potensial dengan informasi yang disajikan adalah nama produk dan total produk terjual.	



Tabel 4.10 Daftar Kebutuhan Fungsional (Lanjutan)

No.	Kode	Kebutuhan	Use Case
48.	POSEW-F-48	Sistem harus dapat mencari produk potensial pada sekumpulan daftar produk potensial.	Cari Produk Potensial
49.	POSEW-F-49	Sistem harus dapat mengurutkan produk potensial.	Urut Produk Potensial
	POSEW-F-49-01	Sistem harus dapat mengurutkan produk potensial berdasarkan produk potensial dan jumlah produk terjual baik dimulai dari posisi terendah maupun tertinggi.	
50.	POSEW-F-50	Sistem harus dapat menampilkan informasi mengenai kombinasi produk.	Lihat Informasi Produk Potensial
	POSEW-F-50-01	Sistem harus dapat menampilkan informasi mengenai kombinasi produk dengan informasi yang disajikan adalah nilai <i>minimum support</i> saat ini, nilai batas minimum produk potensial, dan jumlah produk potensial.	

4.4 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah sekumpulan pernyataan mengenai batasan layanan yang akan disediakan oleh sistem. Kebutuhan non-fungsional biasanya memaparkan kebutuhan yang terkait dengan karakteristik seperti *usability*, *reliability*, *performance*, dan *availability*. Di dalam setiap kebutuhan non-fungsional yang ada akan diberikan kode POSEW-NF-XX-YY, dimana POSEW merupakan singkatan dari *Point of Sale Edd's Waffle*, sedangkan NF merupakan inisial untuk *non-functional* atau kebutuhan non-fungsional, XX akan merepresentasikan nomor urutan kebutuhan. Daftar kebutuhan non-fungsional pada penelitian ini bisa dipaparkan pada Tabel 4.11.

Tabel 4.11 Daftar Kebutuhan Non-Fungsional

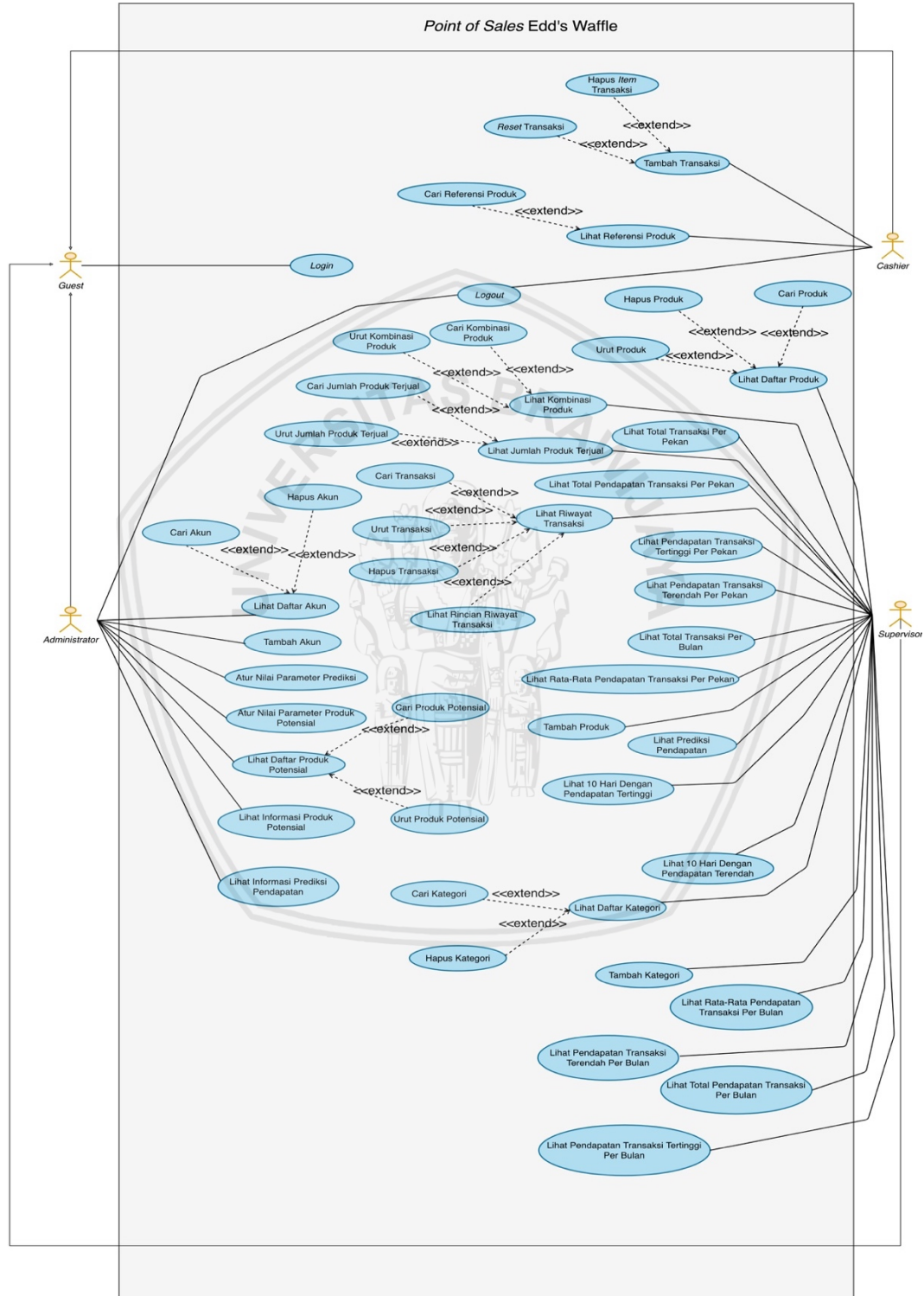
No	Kode Kebutuhan	Deskripsi Kebutuhan
1	POSEW-NF-01	<i>Usability</i> yaitu kemudahan dalam menggunakan sistem yang digunakan oleh setiap segmen pengguna.

4.5 Pemodelan Kebutuhan

4.5.1 Use Case Diagram

Use case diagram adalah diagram yang menggambarkan interaksi antara sistem dan lingkungannya, atau juga dapat dikatakan untuk menjelaskan interaksi antara sistem dengan aktor eksternal sistem (Sommerville, 2011). Use case diagram dibuat berdasarkan kebutuhan fungsional yang telah didefinisikan, tujuannya adalah agar pembaca diagram dapat melihat dan memahami gambaran

fungsionalitas sistem yang akan dibangun. *Use case diagram point of sale Edd's Waffle* yang dibuat dapat dilihat pada Gambar 4.1. *Use case diagram* pada Gambar 4.1 merepresentasikan interaksi antara 50 kebutuhan fungsional yang melibatkan empat segmen pengguna.



Gambar 4.1 Use Case Diagram Point of Sale Edd's Waffle

4.5.2 Use Case Scenario

Use case scenario adalah sekumpulan skenario untuk satu *use case*. *Use case scenario* digunakan untuk mendeskripsikan mengenai informasi detail *use case*, langkah yang terlibat dalam sebuah *use case*. Pada *use case scenario* terdiri dari lima bagian utama. Lima bagian tersebut adalah aktor, kondisi awal, skenario utama, skenario alternatif, dan kondisi akhir. Skenario setiap *use case* terdapat skenario untuk interaksi yang normal dan terdapat skenario untuk setiap kemungkinan alternatif. *Use case scenario* pada penelitian ini ditunjukkan pada Tabel 4.12 hingga 4.83.

Salah satu fungsionalitas yang terdapat pada sistem ialah fungsionalitas *login* untuk melakukan validasi hak akses pengguna. *Use case scenario login* dapat dilihat pada Tabel 4.12.

Tabel 4.12 Use Case Scenario Login

<i>Flow of Events Login</i>	
Kode Kebutuhan	POSEW-F-01
<i>Objective</i>	<i>Use case</i> ini berguna dalam hal mengizinkan <i>Guest</i> untuk mengakses sistem sebagai <i>Supervisor, Cashier</i> atau <i>Administrator</i> .
<i>Actor</i>	<i>Guest</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>login</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> Aktor mengisi <i>username</i> dan <i>password</i> kemudian menekan tombol <i>Login</i>. Sistem menampilkan halaman utama <i>Supervisor, Cashier,</i> atau <i>Administrator</i>.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> Jika <i>username</i> dan <i>password</i> tidak terdaftar pada sistem, sistem menampilkan peringatan "<i>Incorrect Username / Password – Please Try Again</i>".
<i>Post-Condition</i>	Sistem menampilkan halaman utama <i>Supervisor, Cashier,</i> atau <i>Administrator</i> .

Fungsionalitas selanjutnya yang terdapat pada sistem ialah fungsionalitas *logout*. Fungsionalitas ini bertujuan untuk mengeluarkan aktor dari *role* yang dimiliki. *Use case scenario logout* dapat dilihat pada Tabel 4.13.

Tabel 4.13 Use Case Scenario Logout

<i>Flow of Events Logout</i>	
Kode Kebutuhan	POSEW-F-02
<i>Objective</i>	<i>Use case ini berguna untuk mengeluarkan Supervisor, Cashier, dan Administrator dari sistem.</i>
<i>Actor</i>	<i>Supervisor, Cashier, dan Administrator</i>
<i>Pre-condition</i>	Aktor sudah login.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan menu <i>Sign Out</i>. 2. Sistem menampilkan halaman <i>login</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan halaman <i>login</i> .

Fungsionalitas ketiga ialah fungsionalitas tambah transaksi untuk menyelesaikan tahapan pemesanan atau transaksi di Edd' Waffle. *Use case scenario* tambah transaksi dapat dilihat pada Tabel 4.14 - 4.17.

Tabel 4.14 Use Case Scenario Tambah Transaksi

<i>Flow of Events Tambah Transaksi</i>	
Kode Kebutuhan	POSEW-F-03
<i>Objective</i>	<i>Use case ini berguna untuk menambahkan transaksi.</i>
<i>Actor</i>	<i>Cashier</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>Transaction</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan <i>Transaction Code (textfield)</i> terkini. 2. Aktor mengisi <i>Product Code (textfield)</i>. 3. Sistem menampilkan <i>Product Name (textfield)</i> dan <i>Product Price (textfield)</i> yang sesuai dengan <i>Product Code (textfield)</i> yang telah di masukkan. 4. Aktor mengisi <i>Product Quantity (textfield)</i>.

Tabel 4.15 Use Case Scenario Tambah Transaksi (Lanjutan)

Flow of Events Tambah Transaksi	
Main Flow	<ol style="list-style-type: none"> 5. Sistem menampilkan <i>Transaction Code (textfield)</i> terkini. 6. Aktor mengisi <i>Product Code (textfield)</i>. 7. Sistem menampilkan <i>Product Name (textfield)</i> dan <i>Product Price (textfield)</i> yang sesuai dengan <i>Product Code (textfield)</i> yang telah di masukkan. 8. Aktor mengisi <i>Product Quantity (textfield)</i>. 9. Sistem menampilkan <i>Total Price Product (textfield)</i> yang merupakan kalkulasi perkalian antara <i>Product Quantity (textfield)</i> dan <i>Product Price (textfield)</i>. 10. Aktor menekan tombol “Add To Cart”. 11. Sistem menampilkan <i>Product Code, Product Name, Quantity, Price, Total Price</i> terkini yang telah di masukkan aktor dalam representasi tabel. 12. Sistem menampilkan <i>Sub Total</i> berdasarkan kalkulasi jumlah <i>Total Price</i> dalam tabel. 13. Aktor memilih <i>Payment Method (dropdown)</i>, dan mengisi <i>Paid (textfield)</i>. 14. Sistem menampilkan <i>Change</i> yang merupakan kalkulasi pengurangan antara <i>Paid (textfield)</i> dan <i>Sub Total (textfield)</i>. 15. Aktor menekan tombol “Check Out”



Tabel 4.16 Use Case Scenario Tambah Transaksi (Lanjutan)

Flow of Events Tambah Transaksi	
Main Flow	<p>16. Sistem membuka jendela <i>browser</i> baru dengan menampilkan resi transaksi dimana informasi yang disajikan adalah nama <i>restaurant</i>, alamat <i>restaurant</i>, nomor telfon <i>restaurant</i>, tanggal dan waktu, kode transaksi, daftar pesanan (nama produk, kuantitas produk, harga produk, total harga produk), metode pembayaran, diskon, total harga, jumlah bayar tunai, dan jumlah kembalian dalam ekstensi PDF.</p> <p>17. Aktor menyimpan atau mencetak resi transaksi dan menutup jendela <i>browser</i>.</p> <p>Sistem menampilkan peringatan "<i>Transaction Successfully Added !</i>"</p>
Alternative Flow	<ol style="list-style-type: none"> 1. Jika masukan data pada <i>Product Code (textfield)</i> tidak diisi maka sistem akan memunculkan peringatan "<i>Please fill the Product Code correctly</i>" 2. Jika masukan data pada <i>Product Quantity (textfield)</i> tidak diisi maka sistem akan memunculkan peringatan "<i>Please fill the Product Quantity correctly</i>" 3. Jika masukan data pada <i>Product Code (textfield)</i> tidak sesuai dengan kode produk yang telah terdaftar maka sistem akan memunculkan peringatan "<i>Item not found</i>" 4. Jika masukan data uang yang di bayarkan pada transaksi kurang dari total harga transaksi maka sistem akan memunculkan peringatan "<i>Insufficient Money Entered – Please Make Total Paid Is Bigger Than Sub Total</i>".

Tabel 4.17 Use Case Scenario Tambah Transaksi (Lanjutan)

Flow of Events Tambah Transaksi	
Main Flow	<ol style="list-style-type: none"> 5. Jika tidak ada <i>item</i> yang terdapat pada tabel maka sistem akan memunculkan peringatan “No Item Cart Found – Please Do A Transaction Correctly” 6. Jika aktor mengisi <i>Discount</i> dalam persen (<i>textfield</i>), maka sistem akan melakukan kalkulasi pada <i>Sub Total</i> (<i>textfield</i>)
Post-Condition	Sistem menampilkan peringatan “Transaction Successfully Added !”

Fungsionalitas selanjutnya yang terdapat pada sistem ialah fungsionalitas hapus item transaksi yang berfungsi untuk menghapus *item* yang telah dimasukkan oleh *Cashier*. *Use case scenario logout* dapat dilihat pada Tabel 4.18.

Tabel 4.18 Use Case Scenario Hapus Item Transaksi

Flow of Events Hapus Item Transaksi	
Kode Kebutuhan	POSEW-F-04
Objective	<i>Use case</i> ini berguna untuk menghapus <i>item</i> yang telah di masukkan aktor pada tabel di halaman <i>transaction</i>
Actor	<i>Cashier</i>
Pre-condition	Sistem sudah menampilkan daftar <i>item</i> dalam representasi tabel pada halaman <i>Transaction</i> .
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol “Remove” . 2. Sistem menghapus <i>item</i> yang telah dipilih aktor dan melakukan kalkulasi pengurangan sejumlah total harga produk pada <i>Sub Total</i> (<i>textfield</i>) dan kalkulasi ulang pada <i>Change</i> (<i>textfield</i>).
Alternative Flow	-
Post-Condition	Sistem menghapus <i>item</i> yang telah dipilih aktor dan melakukan kalkulasi pengurangan sejumlah total harga produk pada <i>Sub Total</i> (<i>textfield</i>) dan kalkulasi ulang pada <i>Change</i> (<i>textfield</i>).

Fungsionalitas selanjutnya yang terdapat pada sistem ialah fungsionalitas *reset* transaksi yang berfungsi untuk memulai ulang proses penambahan transaksi. *Use case scenario* *reset* transaksi dapat dilihat pada Tabel 4.19.

Tabel 4.19 Use Case Scenario Reset Transaksi

<i>Flow of Events</i> Reset Transaksi	
Kode Kebutuhan	POSEW-F-05
<i>Objective</i>	<i>Use case</i> ini berguna untuk memulai ulang proses tambah transaksi.
<i>Actor</i>	<i>Cashier</i>
<i>Pre-condition</i>	Komponen pada halaman <i>Transaction</i> sudah di isi.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol "<i>Reset</i>". 2. Sistem mengosongkan seluruh <i>field</i> dan tabel yang terdapat pada halaman transaksi.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem mengosongkan seluruh <i>field</i> dan tabel yang terdapat pada halaman transaksi.

Use case scenario selanjutnya yaitu lihat referensi produk sebagai referensi *cashier* untuk mengetahui daftar produk yang tersedia. *Use case scenario* lihat referensi produk dapat dilihat pada Tabel 4.20 dan Tabel 4.21.

Tabel 4.20 Use Case Scenario Lihat Referensi Produk

<i>Flow of Events</i> Reset Transaksi	
Kode Kebutuhan	POSEW-F-06
<i>Objective</i>	<i>Use case</i> ini berguna untuk melihat daftar produk sebagai referensi.
<i>Actor</i>	<i>Cashier</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>ProductReferences</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar produk yang telah terdaftar dengan informasi yang disajikan adalah kode produk dan nama produk.
<i>Alternative Flow</i>	-

Tabel 4.21 Use Case Scenario Lihat Referensi Produk (Lanjutan)

<i>Flow of Events</i> Reset Transaksi	
<i>Post-Condition</i>	Sistem menampilkan daftar produk yang telah terdaftar dengan informasi yang disajikan adalah kode produk dan nama produk.

Fungsionalitas lainnyayang terdapat pada sistem ialah fungsionalitas cari referensi produk yang berfungsi untuk membantu aktor dalam hal menemukan produk agar dapat mempercepat maupun mempermudah proses transaksi. *Use case scenario* cari referensi produk dapat dilihat pada Tabel 4.22.

Tabel 4.22 Use Case Scenario Cari Referensi Produk

<i>Flow of Events</i> Reset Transaksi	
Kode Kebutuhan	POSEW-F-07
<i>Objective</i>	<i>Use case</i> ini berguna untuk mencari referensi produk.
<i>Actor</i>	<i>Cashier</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>ProductReferences</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar produk yang telah terdaftar dengan informasi yang disajikan adalah kode produk dan nama produk. 2. Aktor melakukan pengisian pada kolom pencarian yang terdapat pada kolom <i>code</i> dan <i>name</i>. 3. Sistem menampilkan hasil pencarian berdasarkan masukan data dari aktor.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan hasil pencarian berdasarkan masukan data dari aktor.

Fungsionalitas selanjutnya yang terdapat pada sistem ialah fungsionalitas lihat total transaksi per pekan yang berfungsi untuk menampilkan jumlah total transaksi di setiap pekan, terhitung sejak hari pertama Edd's Waffle melakukan transaksi. *Use case scenario* lihat total transaksi per pekan dapat dilihat pada Tabel 4.23.

Tabel 4.23 Use Case Scenario Lihat Total Transaksi Per Pekan

<i>Flow of Events</i> Lihat Total Transaksi Per Pekan	
Kode Kebutuhan	POSEW-F-08
<i>Objective</i>	<i>Use case</i> ini berguna untuk menampilkan jumlah total transaksi per pekan.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>SalesMonitoring</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih label <i>Weeks</i> dan <i>Number of Transaction</i>. 2. Sistem menampilkan jumlah total transaksi di setiap pekan dalam visualisasi plot pada grafik <i>line chart</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan jumlah total transaksi di setiap pekan dalam visualisasi plot pada grafik <i>line chart</i> .

Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitas lihat total transaksi per bulan. *Use case scenario* lihat total transaksi per bulan dapat dilihat pada Tabel 4.24.

Tabel 4.24 Use Case Scenario Lihat Total Transaksi Per Bulan

<i>Flow of Events</i> Lihat Total Transaksi Per Bulan	
Kode Kebutuhan	POSEW-F-09
<i>Objective</i>	<i>Use case</i> ini berguna untuk menampilkan jumlah total transaksi per bulan.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Aktor memilih menu <i>SalesMonitoring</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih label <i>Months</i> dan <i>Number of Transaction</i>. 2. Sistem menampilkan jumlah total transaksi di setiap bulan dalam visualisasi plot pada grafik <i>line chart</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan jumlah total transaksi di setiap bulan dalam visualisasi plot pada grafik <i>line chart</i> .

Fungsionalitas selanjutnya yang terdapat pada sistem ialah fungsionalitas lihat total pendapatan transaksi per pekan yang berfungsi untuk menampilkan total pendapatan dari transaksi yang telah terjadi dalam skala waktu setiap pekan. *Use case scenario* lihat total pendapatan transaksi per pekan dapat dilihat pada Tabel 4.25.

Tabel 4.25 Use Case Scenario Lihat Total Pendapatan Transaksi Per Pekan

<i>Flow of Events</i> Lihat Total Pendapatan Transaksi Per Pekan	
Kode Kebutuhan	POSEW-F-10
<i>Objective</i>	<i>Use case</i> ini berguna untuk menampilkan total pendapatan transaksi per pekan dari transaksi yang telah terjadi.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>SalesMonitoring</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih label <i>Weeks</i> dan <i>Total Revenue</i>. 2. Sistem menampilkan jumlah total pendapatan transaksi di setiap pekan dalam visualisasi plot pada grafik <i>line chart</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan jumlah total pendapatan transaksi di setiap pekan dalam visualisasi plot pada grafik <i>line chart</i> .

Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitas lihat total pendapatan transaksi per bulan. *Use case scenario* lihat total pendapatan transaksi per bulan dapat dilihat pada Tabel 4.26 dan Tabel 4.27.

Tabel 4.26 Use Case Scenario Lihat Total Pendapatan Transaksi Per Bulan

<i>Flow of Events</i> Lihat Total Pendapatan Transaksi Per Bulan	
Kode Kebutuhan	POSEW-F-11
<i>Objective</i>	<i>Use case</i> ini berguna untuk menampilkan total pendapatan transaksi per bulan dari transaksi yang telah terjadi.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>SalesMonitoring</i>

Tabel 4.27 Use Case Scenario Lihat Total Pendapatan Transaksi Per Bulan (Lanjutan)

<i>Flow of Events</i> Lihat Total Pendapatan Transaksi Per Bulan	
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih label <i>Months</i> dan <i>Total Revenue</i>. 2. Sistem menampilkan jumlah total pendapatan transaksi di setiap bulan dalam visualisasi plot pada grafik <i>line chart</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan jumlah total pendapatan transaksi di setiap bulan dalam visualisasi plot pada grafik <i>line chart</i> .

Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitas lihat pendapatan transaksi tertinggi per pekan untuk menampilkan pendapatan tertinggi dari suatu transaksi dalam skala waktu setiap pekan. *Use case scenario* lihat pendapatan transaksi tertinggi per pekan dapat dilihat pada Tabel 4.28.

Tabel 4.28 Use Case Scenario Lihat Pendapatan Transaksi Tertinggi Per Pekan

<i>Flow of Events</i> Lihat Pendapatan Transaksi Tertinggi Per Pekan	
Kode Kebutuhan	POSEW-F-12
<i>Objective</i>	<i>Use case</i> ini berguna untuk menampilkan jumlah pendapatan tertinggi dari transaksi yang telah terjadi dengan skala waktu setiap pekan.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>SalesMonitoring</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih label <i>Weeks</i> dan <i>Maximum Transaction</i>. 2. Sistem menampilkan jumlah pendapatan tertinggi dalam satu transaksi di setiap pekan dalam visualisasi plot pada grafik <i>line chart</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan jumlah pendapatan tertinggi dalam satu transaksi di setiap pekan dalam visualisasi plot pada grafik <i>line chart</i> .

Kebutuhan fungsional selanjutnya yang terdapat pada sistem ialah fungsionalitas lihat pendapatan transaksi tertinggi per bulan yang berfungsi untuk menampilkan pendapatan tertinggi dari suatu transaksi yang telah terjadi dalam skala waktu setiap bulan. *Use case scenario* lihat pendapatan transaksi tertinggi per bulan dapat dilihat pada Tabel 4.29.

Tabel 4.29 Use Case Scenario Lihat Pendapatan Transaksi Tertinggi Per Bulan

<i>Flow of Events</i> Lihat Pendapatan Transaksi Tertinggi Per Bulan	
Kode Kebutuhan	POSEW-F-13
<i>Objective</i>	<i>Use case</i> ini berguna untuk menampilkan jumlah pendapatan tertinggi per bulan dari transaksi yang telah terjadi
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>SalesMonitoring</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih label <i>Months</i> dan <i>Maximum Transaction</i>. 2. Sistem menampilkan jumlah pendapatan tertinggi dalam satu transaksi di setiap bulan dalam visualisasi plot pada grafik <i>line chart</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan jumlah pendapatan tertinggi dalam satu transaksi di setiap bulan dalam visualisasi plot pada grafik <i>line chart</i> .

Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitas lihat pendapatan transaksi terendah per pekan. *Use case scenario* lihat pendapatan transaksi terendah per pekan dapat dilihat pada Tabel 4.30 dan Tabel 4.31.

Tabel 4.30 Use Case Scenario Lihat Pendapatan Transaksi Terendah Per Pekan

<i>Flow of Events</i> Lihat Pendapatan Transaksi Terendah Per Pekan	
Kode Kebutuhan	POSEW-F-14
<i>Objective</i>	<i>Use case</i> ini berguna untuk menampilkan jumlah pendapatan terendah per pekan dari transaksi yang telah terjadi.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>SalesMonitoring</i>

Tabel 4.31 Use Case Scenario Lihat Pendapatan Transaksi Terendah Per Pekan (Lanjutan)

<i>Flow of Events</i> Lihat Pendapatan Transaksi Terendah Per Pekan	
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih label <i>Weeks</i> dan <i>Minimum Transaction</i>. 2. Sistem menampilkan jumlah pendapatan terendah dalam satu transaksi di setiap pekan dalam visualisasi plot pada grafik <i>line chart</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan jumlah pendapatan terendah dalam satu transaksi di setiap bulan dalam visualisasi plot pada grafik <i>line chart</i> .

Fungsionalitas selanjutnya yang terdapat pada sistem ialah fungsionalitas lihat pendapatan transaksi terendah per bulan yang berfungsi untuk menampilkan pendapatan terendah dari suatu transaksi yang telah terjadi dalam skala waktu setiap bulan. *Use case scenario* lihat pendapatan transaksi terendah per bulan dapat dilihat pada Tabel 4.32.

Tabel 4.32 Use Case Scenario Lihat Pendapatan Transaksi Terendah Per Bulan

<i>Flow of Events</i> Lihat Pendapatan Transaksi Terendah Per Bulan	
Kode Kebutuhan	POSEW-F-15
<i>Objective</i>	<i>Use case</i> ini berguna untuk menampilkan jumlah pendapatan terendah per bulan dari transaksi yang telah terjadi.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>SalesMonitoring</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih label <i>Months</i> dan <i>Minimum Transaction</i>. 2. Sistem menampilkan jumlah pendapatan terendah dalam satu transaksi di setiap bulan dalam visualisasi plot pada grafik <i>line chart</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan jumlah pendapatan terendah dalam satu transaksi di setiap bulan dalam visualisasi plot pada grafik <i>line chart</i> .

Fungsionalitas selanjutnya yang terdapat pada sistem ialah fungsionalitas lihat total transaksi. *Use case scenario* lihat total transaksi dapat dilihat pada Tabel 4.33.

Tabel 4.33 Use Case Scenario Lihat Rata-Rata Pendapatan Transaksi Per Pekan

<i>Flow of Events</i> Lihat Rata-Rata Pendapatan Transaksi Per Pekan	
Kode Kebutuhan	POSEW-F-16
<i>Objective</i>	<i>Use case</i> ini berguna untuk menampilkan rata-rata pendapatan per pekan dari transaksi yang telah terjadi.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>SalesMonitoring</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih label <i>Weeks</i> dan <i>Average Transaction</i>. 2. Sistem menampilkan rata-rata pendapatan transaksi di setiap pekan dalam visualisasi plot pada grafik <i>line chart</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan rata-rata pendapatan transaksi di setiap pekan dalam visualisasi plot pada grafik <i>line chart</i> .

Use case scenario lihat rata-rata pendapatan transaksi per bulan dapat dilihat pada Tabel 4.34 dan Tabel 4.35.

Tabel 4.34 Use Case Scenario Lihat Rata-Rata Pendapatan Transaksi Per Bulan

<i>Flow of Events</i> Lihat Rata-Rata Pendapatan Transaksi Per Bulan	
Kode Kebutuhan	POSEW-F-17
<i>Objective</i>	<i>Use case</i> ini berguna untuk menampilkan rata-rata pendapatan per bulan dari transaksi yang telah terjadi.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>SalesMonitoring</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih label <i>Months</i> dan <i>Average Transaction</i>.

Tabel 4.35 Use Case Scenario Lihat Rata-Rata Pendapatan Transaksi Per Bulan (Lanjutan)

<i>Main Flow</i>	2. Sistem menampilkan rata-rata pendapatan transaksi di setiap bulan dalam visualisasi plot pada grafik <i>line chart</i> .
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan rata-rata pendapatan transaksi di setiap bulan dalam visualisasi plot pada grafik <i>line chart</i> .

Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitas lihat prediksi pendapatan yang berfungsi untuk menampilkan prediksi pendapatan di waktu yang akan datang. *Use case scenario* lihat prediksi pendapatan dapat dilihat pada Tabel 4.36.

Tabel 4.36 Use Case Scenario Lihat Prediksi Pendapatan

<i>Flow of Events</i> Lihat Prediksi Pendapatan	
Kode Kebutuhan	POSEW-F-18
<i>Objective</i>	<i>Use case</i> ini berguna untuk harus dapat menampilkan prediksi pendapatan.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>SalesMonitoring</i>
<i>Main Flow</i>	1. Sistem menampilkan prediksi pendapatan untuk 4 pekan yang akan datang dalam visualisasi plot pada grafik <i>line chart</i> .
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan prediksi pendapatan untuk 4 pekan yang akan datang dalam visualisasi grafik <i>line chart</i> .

Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitas lihat 10 hari dengan pendapatan tertinggi yang berfungsi untuk menampilkan 10 hari dengan pendapatan tertinggi yang pernah terjadi. *Use case scenario* lihat 10 hari dengan pendapatan tertinggi dapat dilihat pada Tabel 4.37.

Tabel 4.37 Use Case Scenario Lihat 10 Hari Dengan Pendapatan Tertinggi

<i>Flow of Events</i> Lihat 10 Hari Dengan Pendapatan Tertinggi	
Kode Kebutuhan	POSEW-F-19
<i>Objective</i>	<i>Use case</i> ini berguna untuk 10 hari dengan pendapatan tertinggi.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>SalesMonitoring</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan 10 hari dengan pendapatan tertinggi dengan informasi yang disajikan adalah tanggal dan total pendapatan dari transaksi yang terjadi pada tanggal tersebut dalam visualisasi grafik <i>area chart</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan 10 hari dengan pendapatan tertinggi dengan informasi yang disajikan adalah tanggal dan total pendapatan dari transaksi yang terjadi pada tanggal tersebut dalam visualisasi grafik <i>area chart</i> .

Use case scenario lihat 10 hari dengan pendapatan terendah dapat dilihat pada Tabel 4.38.

Tabel 4.38 Use Case Scenario Lihat 10 Hari Dengan Pendapatan Terendah

<i>Flow of Events</i> Lihat 10 Hari Dengan Pendapatan Terendah	
Kode Kebutuhan	POSEW-F-20
<i>Objective</i>	<i>Use case</i> ini berguna untuk 10 hari dengan pendapatan terendah.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>SalesMonitoring</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan 10 hari dengan pendapatan terendah dengan informasi yang disajikan adalah tanggal dan total pendapatan dari transaksi yang terjadi pada tanggal tersebut dalam visualisasi grafik <i>area chart</i>.

Tabel 4.39 Use Case Scenario Lihat 10 Hari Dengan Pendapatan Terendah (Lanjutan)

<i>Flow of Events</i> Lihat 10 Hari Dengan Pendapatan Terendah	
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan 10 hari dengan pendapatan terendah dengan informasi yang disajikan adalah tanggal dan total pendapatan dari transaksi yang terjadi pada tanggal tersebut dalam visualisasi grafik <i>area chart</i> .

Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitas lihat riwayat transaksi yang pernah terjadi di Edd's Waffle untuk melihat kumpulan transaksi yang pernah terjadi. *Use case scenario* lihat riwayat transaksi dapat dilihat pada Tabel 4.40.

Tabel 4.40 Use Case Scenario Lihat Riwayat Transaksi

<i>Flow of Events</i> Lihat Riwayat Transaksi	
Kode Kebutuhan	POSEW-F-21
<i>Objective</i>	<i>Use case</i> ini berguna untuk menampilkan riwayat transaksi.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>History</i>
<i>Main Flow</i>	1. Sistem menampilkan riwayat transaksi dengan informasi yang disajikan adalah kode transaksi, tanggal transaksi, diskon pada transaksi, metode pembayaran transaksi, dan total biaya transaksi, total transaksi, dan total pendapatan dalam representasi tabel.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan riwayat transaksi dengan informasi yang disajikan adalah kode transaksi, tanggal transaksi, diskon pada transaksi, metode pembayaran transaksi, dan total biaya transaksi, total transaksi, dan total pendapatan dalam representasi tabel.

Fungsionalitas selanjutnya yang terdapat pada sistem ialah fungsionalitas lihat rincian riwayat transaksi yang pernah terjadi di Edd's Waffle. Rincian riwayat transaksi merupakan rincian dari riwayat transaksi yang ada. *Use case scenario* lihat rincian riwayat transaksi dapat dilihat pada Tabel 4.41.

Tabel 4.41 Use Case Scenario Lihat Rincian Riwayat Transaksi

<i>Flow of Events</i> Lihat Rincian Riwayat Transaksi	
Kode Kebutuhan	POSEW-F-22
<i>Objective</i>	<i>Use case</i> ini berguna untuk menampilkan rincian dari riwayat transaksi.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>History</i> dan riwayat transaksi tersedia
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih kumpulan data pada riwayat transaksi dan menekan <i>icon expand</i>. 2. Sistem menampilkan rincian dari riwayat transaksi dengan informasi yang disajikan adalah daftar nama produk, jumlah produk, dan harga produk pada transaksi yang bersangkutan.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan rincian dari riwayat transaksi dengan informasi yang disajikan adalah daftar nama produk, jumlah produk, dan harga produk pada transaksi yang bersangkutan.

Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitas hapus transaksi. *Use case scenario* hapus transaksi dapat dilihat pada Tabel 4.42 dan Tabel 4.43.

Tabel 4.42 Use Case Scenario Hapus Transaksi

<i>Flow of Events</i> Hapus Transaksi	
Kode Kebutuhan	POSEW-F-23
<i>Objective</i>	<i>Use case</i> ini berguna untuk menghapus transaksi yang telah terjadi.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>History</i>

Tabel 4.43 Use Case Scenario Hapus Transaksi (Lanjutan)

Flow of Events Hapus Transaksi	
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih riwayat transaksi yang ingin di hapus dan menekan tombol "Delete" 2. Sistem menampilkan peringatan "Are you sure to delete this record ?" 3. Aktor menekan tombol "Oke"
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Apabila aktor menekan tombol "Batal" sebagai respon peringatan "Are you sure you want to delete this record ?" maka riwayat transaksi terpilih tidak jadi di hapus.
<i>Post-Condition</i>	Sistem menghapus transaksi yang dipilih dan menampilkan data riwayat transaksi terkini.

Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitas cari transaksi dimana berguna untuk mencari transaksi dari sekumpulan riwayat transaksi yang ada di Edd's Waffle. *Use case scenario* cari transaksi dapat dilihat pada Tabel 4.44 dan Tabel 4.45.

Tabel 4.44 Use Case Scenario Cari Transaksi

Flow of Events Cari Transaksi	
Kode Kebutuhan	POSEW-F-24
<i>Objective</i>	<i>Use case</i> ini berguna untuk mencari transaksi dari sekumpulan riwayat transaksi.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>History</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan <i>box-area</i> pada kolom riwayat transaksi. 2. Sistem menampilkan <i>pop-up</i> yang menyediakan layanan pencarian. 3. Aktor mencari transaksi berdasarkan kode transaksi, tanggal transaksi, metode pembayaran, total harga atau berdasarkan rentang minimal dan maksimal kode transaksi, tanggal transaksi, dan total harga dan menekan tombol <i>Filter</i>.

Tabel 4.45 Use Case Scenario Cari Transaksi (Lanjutan)

<i>Flow of Events</i> Cari Transaksi	
<i>Main Flow</i>	4. Sistem menampilkan data yang sesuai dengan masukan data yang diberikan aktor.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan data yang sesuai dengan masukan data yang diberikan aktor.

Fungsionalitas selanjutnya yang terdapat pada sistem ialah fungsionalitas urut transaksi dimana berguna untuk mengurutkan transaksi dari sekumpulan riwayat transaksi yang ada di Edd's Waffle. *Use case scenario* urut transaksi dapat dilihat pada Tabel 4.46.

Tabel 4.46 Use Case Scenario Urut Transaksi

<i>Flow of Events</i> Urut Transaksi	
Kode Kebutuhan	POSEW-F-25
<i>Objective</i>	<i>Use case</i> ini berguna untuk mengurutkan transaksi dari sekumpulan riwayat transaksi.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>History</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan <i>title</i> pada salah satu kolom riwayat transaksi. 2. Sistem menampilkan data riwayat transaksi dengan urutan yang dimulai dari posisi terendah hingga tertinggi.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika aktor menekan <i>title</i> sebanyak dua kali pada salah satu kolom riwayat transaksi, maka sistem akan menampilkan data riwayat transaksi dengan urutan yang dimulai dari posisi tertinggi hingga terendah.
<i>Post-Condition</i>	Sistem menampilkan data riwayat transaksi dengan urutan yang dimulai dari posisi terendah hingga tertinggi.

Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitas lihat kombinasi produk. *Use case scenario* lihat kombinasi produk dapat dilihat pada Tabel 4.47.

Tabel 4.47 Use Case Scenario Lihat Kombinasi Produk

<i>Flow of Events</i> Lihat Kombinasi Produk	
Kode Kebutuhan	POSEW-F-26
<i>Objective</i>	<i>Use case</i> ini berguna untuk menampilkan kombinasi produk yang bertemu dari transaksi yang telah terjadi.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>Trends</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan kombinasi produk yang bertemu dari transaksi yang telah terjadi dengan informasi yang disajikan adalah nama produk yang di kombinasi kan dan jumlah kombinasi produk tersebut bertemu.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan kombinasi produk yang bertemu dari transaksi yang telah terjadi dengan informasi yang disajikan adalah nama produk yang di kombinasi kan dan jumlah kombinasi produk tersebut bertemu.

Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitas cari kombinasi produk dimana berguna untuk mencari kombinasi produk yang ada dari sekumpulan kombinasi yang tersedia. *Use case scenario* cari kombinasi produk dapat dilihat pada Tabel 4.48.

Tabel 4.48 Use Case Scenario Cari Kombinasi Produk

<i>Flow of Events</i> Cari Kombinasi Produk	
Kode Kebutuhan	POSEW-F-27
<i>Objective</i>	<i>Use case</i> ini berguna untuk mencari kombinasi produk.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>Trends</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mencari kombinasi produk berdasarkan kombinasi produk pada <i>search area</i>. 2. Sistem menampilkan data kombinasi produk yang sesuai dengan masukan data yang diberikan aktor.
<i>Alternative Flow</i>	-

Tabel 4.49 Use Case Scenario Cari Kombinasi Produk (Lanjutan)

<i>Flow of Events</i> Cari Kombinasi Produk	
<i>Post-Condition</i>	Sistem menampilkan data kombinasi produk yang sesuai dengan masukan data yang diberikan aktor.

Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitas urut kombinasi produk dimana berguna untuk mengurutkan kombinasi produk dari sekumpulan kombinasi yang ada. *Use case scenario* urut kombinasi produk dapat dilihat pada Tabel 4.50.

Tabel 4.50 Use Case Scenario Urut Kombinasi Produk

<i>Flow of Events</i> Urut Kombinasi Produk	
Kode Kebutuhan	POSEW-F-28
<i>Objective</i>	<i>Use case</i> ini berguna untuk mengurutkan kombinasi produk.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>Trends</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan <i>title</i> pada salah satu kolom kombinasi produk. 2. Sistem menampilkan data kombinasi produk dengan urutan yang dimulai dari posisi terendah hingga tertinggi.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika aktor menekan <i>title</i> sebanyak dua kali pada salah satu kolom kombinasi produk, maka sistem akan menampilkan data kombinasi produk dengan urutan yang dimulai dari posisi tertinggi hingga terendah.
<i>Post-Condition</i>	Sistem menampilkan data kombinasi produk dengan urutan yang dimulai dari posisi terendah hingga tertinggi.

Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitas lihat jumlah produk terjual. Melalui fungsionalitas ini, *supervisor* dapat melihat jumlah produk terjual dari transaksi di Edd's Waffle. *Use case scenario* lihat jumlah produk terjual dapat dilihat pada Tabel 4.51.

Tabel 4.51 Use Case Scenario Lihat Jumlah Produk Terjual

<i>Flow of Events</i> Lihat Jumlah Produk Terjual	
Kode Kebutuhan	POSEW-F-29
<i>Objective</i>	<i>Use case</i> ini berguna untuk menampilkan jumlah produk yang telah terjual.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>Trends</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan jumlah produk yang telah terjual dengan informasi yang disajikan adalah nama produk dan jumlah produk terjual.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan jumlah produk yang telah terjual dengan informasi yang disajikan adalah nama produk dan jumlah produk terjual.

Fungsionalitas selanjutnya yang terdapat pada sistem ialah fungsionalitas cari jumlah produk terjual dimana berguna untuk mencari produk dan mengetahui seberapa banyak produk tersebut terjual. *Use case scenario* cari jumlah produk terjual dapat dilihat pada Tabel 4.52.

Tabel 4.52 Use Case Scenario Cari Jumlah Produk Terjual

<i>Flow of Events</i> Cari Jumlah Produk Terjual	
Kode Kebutuhan	POSEW-F-30
<i>Objective</i>	<i>Use case</i> ini berguna untuk mencari jumlah produk terjual.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>Trends</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mencari produk berdasarkan nama produk pada <i>search area</i>. 2. Sistem menampilkan data jumlah produk terjual yang sesuai dengan masukan data yang diberikan aktor.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan data jumlah produk terjual yang sesuai dengan masukan data yang diberikan aktor.



Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitasurut kombinasi produk dimana berguna untuk mengurutkan kombinasi produk dari sekumpulan kombinasi yang ada. *Use case scenario* urut kombinasi produk dapat dilihat pada Tabel 4.53.

Tabel 4.53 Use Case Scenario Urut Jumlah Produk Terjual

<i>Flow of Events</i> Urut Transaksi	
Kode Kebutuhan	POSEW-F-31
<i>Objective</i>	<i>Use case</i> ini berguna untuk mengurutkan jumlah produk terjual.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>Trends</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan <i>title</i> pada salah satu kolom jumlah produk terjual. 2. Sistem menampilkan data dengan urutan yang dimulai dari posisi terendah hingga tertinggi.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika aktor menekan <i>title</i> sebanyak 2 kali pada salah satu kolom jumlah produk terjual, maka sistem akan menampilkan data dengan urutan yang dimulai dari posisi tertinggi hingga terendah.
<i>Post-Condition</i>	Sistem mengurutkan data data dengan urutan yang dimulai dari posisi terendah hingga tertinggi.

Fungsionalitas lainnya ialah fungsionalitas lihat daftar produk yang telah terdaftar. *Use case scenario* lihat daftar produk dapat dilihat pada Tabel 4.54 dan Tabel 4.55.

Tabel 4.54 Use Case Scenario Lihat Daftar Produk

<i>Flow of Events</i> Lihat Daftar Produk	
Kode Kebutuhan	POSEW-F-32
<i>Objective</i>	<i>Use case</i> ini berguna untuk menampilkan daftar produk yang telah terdaftar.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>SetProduct</i>

Tabel 4.55 Use Case Scenario Lihat Daftar Produk (Lanjutan)

<i>Flow of Events</i> Lihat Daftar Produk	
<i>Main Flow</i>	1. Sistem menampilkan daftar produk yang telah terdaftar dengan informasi yang disajikan adalah kode produk, nama produk, kategori produk, dan harga produk.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan daftar produk yang telah terdaftar dengan informasi yang disajikan adalah kode produk, nama produk, kategori produk, dan harga produk.

Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitas cari produk dimana berguna untuk mencari produk dan mengetahui informasi dari produk tersebut. *Use case scenario* cari produk dapat dilihat pada Tabel 4.56.

Tabel 4.56 Use Case Scenario Cari Produk

<i>Flow of Events</i> Cari Produk	
Kode Kebutuhan	POSEW-F-33
<i>Objective</i>	<i>Use case</i> ini berguna untuk mencari produk.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>SetProduct</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mencari produk berdasarkan kode produk, nama produk, kategori produk atau harga produk. 2. Sistem menampilkan data produk yang sesuai dengan masukan data yang diberikan aktor.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan data produk yang sesuai dengan masukan data yang diberikan aktor.

Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitas tambah produk. Melalui fungsionalitas ini, *supervisor* dapat menambahkan produk yang dijual di Edd's Waffle. *Use case scenario* tambah produk dapat dilihat pada Tabel 4.57.



Tabel 4.57 Use Case Scenario Tambah Produk

<i>Flow of Events</i> Tambah Produk	
Kode Kebutuhan	POSEW-F-34
<i>Objective</i>	<i>Use case ini berguna untuk menambahkan produk.</i>
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>SetProduct</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol <i>Add Product</i>. 2. Sistem menampilkan <i>form Add Product</i> dan menampilkan data yang dibutuhkan yaitu kode produk (<i>text field</i>), nama produk (<i>text field</i>), kategori produk (<i>dropdown</i>), dan harga produk (<i>text field</i>). 3. Aktor mengisi <i>form</i> dengan data yang dibutuhkan yaitu kode produk (<i>text field</i>), nama produk (<i>text field</i>), kategori produk (<i>dropdown</i>), dan harga produk (<i>text field</i>) dan menekan tombol <i>Add Product</i>.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika kode produk telah terdaftar, sistem akan menampilkan pesan "<i>Adding Product Failed (Product Code Already Exist) – Please Try Again</i>" 2. Jika terdapat data pada <i>form</i> yang tidak di isi maka sistem akan menampilkan pesan "<i>Error – Please Try Again and Fill The Form Completely</i>" 3. Jika aktor menekan tombol <i>Cancel</i> maka <i>form Add Product</i> akan ditutup.
<i>Post-Condition</i>	Sistem menampilkan peringatan " <i>New Product Successfully Added !</i> ".

Use case scenario hapus produk dapat dilihat pada Tabel 4.58 dan Tabel 4.59.

Tabel 4.58 Use Case Scenario Hapus Produk

<i>Flow of Events</i> Hapus Produk	
Kode Kebutuhan	POSEW-F-35



Tabel 4.59 Use Case Scenario Hapus Produk (Lanjutan)

<i>Flow of Events</i> Hapus Produk	
<i>Objective</i>	<i>Use case</i> ini berguna untuk menghapus produk.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>SetProduct</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih produk yang ingin di hapus dan menekan tombol "<i>Delete</i>" 2. Sistem menampilkan peringatan "<i>Are you sure you want to delete this record ?</i>" 3. Aktor menekan tombol "<i>Oke</i>"
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Apabila aktor menekan tombol "<i>Batal</i>" sebagai respon peringatan "<i>Are you sure you want to delete this record ?</i>" maka produk terpilih tidak jadi di hapus.
<i>Post-Condition</i>	Sistem menghapus data produk yang dipilih dan menampilkan data produk terkini.

Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitas lihat daftar kategori. Melalui fungsionalitas ini, *supervisor* dapat melihat daftar kategori yang telah terdaftar di Edd's Waffle. *Use case scenario* lihat daftar kategori dapat dilihat pada Tabel 4.60 dan Tabel 4.61.

Tabel 4.60 Use Case Scenario Lihat Daftar Kategori

<i>Flow of Events</i> Lihat Daftar Kategori	
Kode Kebutuhan	POSEW-F-36
<i>Objective</i>	<i>Use case</i> ini berguna untuk menampilkan daftar kategori
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>Category</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar kategori yang telah terdaftar dengan informasi yang disajikan adalah nama kategori.
<i>Alternative Flow</i>	-

Tabel 4.61 Use Case Scenario Lihat Daftar Kategori (Lanjutan)

<i>Flow of Events</i> Lihat Daftar Kategori	
<i>Post-Condition</i>	Sistem menampilkan daftar kategori yang telah terdaftar dengan informasi yang disajikan adalah nama kategori.

Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitas cari kombinasi produk dimana berguna untuk mencari kategori yang telah terdaftar. *Use case scenario* cari kategori dapat dilihat pada Tabel 4.62.

Tabel 4.62 Use Case Scenario Cari Kategori

<i>Flow of Events</i> Cari Kategori	
Kode Kebutuhan	POSEW-F-37
<i>Objective</i>	<i>Use case</i> ini berguna untuk mencari kategori.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>Category</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mencari kategori pada <i>search area</i>. 2. Sistem menampilkan data kategori yang sesuai dengan masukan data yang diberikan aktor.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan data kategori yang sesuai dengan masukan data yang diberikan aktor.

Fungsionalitas selanjutnya yang terdapat pada sistem ialah tambah kategori. *Use case scenario* tambah kategori dapat dilihat pada Tabel 4.63 dan Tabel 4.64.

Tabel 4.63 Use Case Scenario Tambah Kategori

<i>Flow of Events</i> Tambah Kategori	
Kode Kebutuhan	POSEW-F-38
<i>Objective</i>	<i>Use case</i> ini berguna untuk menambahkan kategori.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>Category</i>

Tabel 4.64 Use Case Scenario Tambah Kategori (Lanjutan)

<i>Flow of Events</i> Tambah Kategori	
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol <i>Add Category</i>. 2. Sistem menampilkan <i>form Add Category</i> dan menampilkan data yang dibutuhkan yaitu <i>Category Name (text field)</i>. 3. Aktor mengisi <i>form</i> dengan data yang dibutuhkan yaitu <i>Category Name (text field)</i> dan menekan tombol <i>Add Category</i>.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika kode produk telah terdaftar, sistem akan menampilkan pesan "<i>Category Is Already Exist – Please Try Again</i>" 2. Jika terdapat data pada <i>form</i> yang tidak di isi maka sistem akan menampilkan pesan "<i>Error – Please Try Again and Fill The Category Name Correctly</i>" 3. Jika aktor menekan tombol <i>Cancel</i>, maka <i>form Add Category</i> akan ditutup.
<i>Post-Condition</i>	Sistem menampilkan peringatan " <i>New Category Successfully Added !</i> "

Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitas hapus kategori. *Use case scenario* hapus kategori dapat dilihat pada Tabel 4.65 dan Tabel 4.66.

Tabel 4.65 Use Case Scenario Hapus Kategori

<i>Flow of Events</i> Hapus Kategori	
Kode Kebutuhan	POSEW-F-39
<i>Objective</i>	<i>Use case</i> ini berguna untuk menghapus kategori.
<i>Actor</i>	<i>Supervisor</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>Category</i>



Tabel 4.66 Use Case Scenario Hapus Kategori (Lanjutan)

Flow of Events Hapus Kategori	
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih kategori yang ingin di hapus dan menekan tombol “Delete” 2. Sistem menampilkan peringatan “Are you sure you want to delete this record ?” 3. Aktor menekan tombol “Yes”
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Apabila aktor menekan tombol “Batal” sebagai respon peringatan “Are you sure you want to delete this record ?” maka kategori terpilih tidak jadi di hapus.
<i>Post-Condition</i>	Sistem menghapus data kategori yang dipilih dan menampilkan data kategori terkini.

Fungsionalitas selanjutnya yang terdapat pada sistem ialah fungsionalitas lihat daftar akun. Melalui fungsionalitas ini, *administrator* dapat lihat daftar akun yang telah terdaftar di dalam sistem. *Use case scenario* lihat daftar akun dapat dilihat pada Tabel 4.67.

Tabel 4.67 Use Case Scenario Lihat Daftar Akun

Flow of Events Lihat Daftar Akun	
Kode Kebutuhan	POSEW-F-40
<i>Objective</i>	<i>Use case</i> ini berguna untuk menampilkan daftar akun.
<i>Actor</i>	<i>Administrator</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>Accounts</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar akun yang telah terdaftar dengan informasi yang disajikan adalah <i>username</i> dan <i>role</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan daftar akun yang telah terdaftar dengan informasi yang disajikan adalah <i>username</i> dan <i>role</i> .

Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitas cari akun. *Use case scenario* cari akun dapat dilihat pada Tabel 4.68

Tabel 4.68 Use Case Scenario Cari Akun

Flow of Events Cari Akun	
Kode Kebutuhan	POSEW-F-41
Objective	Use case ini berguna untuk mencari akun yang telah terdaftar.
Actor	Administrator
Pre-condition	Sistem sudah menampilkan halaman <i>Accounts</i> .
Main Flow	<ol style="list-style-type: none"> 1. Aktor mencari akun berdasarkan <i>username</i> dan <i>role</i> pada <i>search area</i>. 2. Sistem menampilkan data akun yang sesuai dengan masukan data yang diberikan aktor.
Alternative Flow	-
Post-Condition	Sistem menampilkan data akun yang sesuai dengan masukan data yang diberikan aktor.

Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitas tambah akun. Melalui fungsionalitas ini, *administrator* dapat menambahkan akun sesuai dengan segmen atau *role* nya masing-masing. *Use case scenario* tambah akun dapat dilihat pada Tabel 4.69 dan Tabel 4.70.

Tabel 4.69 Use Case Scenario Tambah Akun

Flow of Events Tambah Akun	
Kode Kebutuhan	POSEW-F-42
Objective	Use case ini berguna untuk menambahkan produk.
Actor	Administrator
Pre-condition	Sistem sudah menampilkan halaman <i>Accounts</i> .
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol <i>Add New Account</i>. 2. Sistem menampilkan <i>form Add New Account</i> dan menampilkan data yang dibutuhkan yaitu <i>Username (text field)</i>, <i>Password (text field)</i>, dan <i>Role (dropdown)</i> kemudian menekan tombol <i>Add Account</i>.

Tabel 4.70 Use Case Scenario Tambah Akun (Lanjutan)

Flow of Events Tambah Akun	
<i>Main Flow</i>	3. Aktor mengisi <i>form</i> dengan data yang dibutuhkan yaitu <i>Username (text field)</i> , <i>Password (text field)</i> , dan <i>Role (dropdown)</i> kemudian
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika Username telah terdaftar, sistem akan menampilkan pesan "<i>Username Is Already Exist – Please Try Again</i>" 2. Jika terdapat data pada <i>form</i> yang tidak di isi maka sistem akan menampilkan pesan "<i>Username / Password Must Be Filled – Please Try Again</i>" 3. Jika aktor menekan tombol <i>cancel</i> maka <i>form Add New Account</i> akan ditutup.
<i>Post-Condition</i>	Sistem menampilkan peringatan " <i>New Account Successfully Added</i> "

Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitas hapus akun. Melalui fungsionalitas ini, *administrator* dapat menghapus akun yang telah terdaftar pada sistem. *Use case scenario* hapus akun dapat dilihat pada Tabel 4.71 dan Tabel 4.72.

Tabel 4.71 Use Case Scenario Hapus Akun

Flow of Events Hapus Akun	
Kode Kebutuhan	POSEW-F-43
<i>Objective</i>	<i>Use case</i> ini berguna untuk menghapus akun yang telah terdaftar.
<i>Actor</i>	<i>Administrator</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>Accounts</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih akun yang ingin di hapus dan menekan tombol "<i>Delete</i>" 2. Sistem menampilkan peringatan "<i>Are you sure to delete this record ?</i>" 3. Aktor menekan tombol "<i>Oke</i>"

Tabel 4.72 Use Case Scenario Hapus Akun (Lanjutan)

Flow of Events Hapus Akun	
<i>Alternative Flow</i>	1. Apabila aktor menekan tombol “Batal” sebagai respon peringatan “Are you sure you want to delete this record ?” maka akun terpilih tidak jadi di hapus.
<i>Post-Condition</i>	Sistem menghapus data akun yang dipilih dan menampilkan daftar akun yang telah terdaftar saat ini.

Fungsionalitas selanjutnya ialah atur nilai parameter prediksi untuk mengatur nilai *alpha* dan *beta* agar dapat memprediksi pendapatan. *Use case scenario* atur nilai parameter prediksi dapat dilihat pada Tabel 4.73.

Tabel 4.73 Use Case Scenario Atur Nilai Parameter Prediksi

Flow of Events Atur Nilai Parameter Prediksi	
Kode Kebutuhan	POSEW-F-44
<i>Objective</i>	<i>Use case</i> ini berguna untuk menetapkan nilai <i>alpha</i> dan <i>beta</i> .
<i>Actor</i>	<i>Administrator</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>MiningPredict</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol <i>Set Parameter Value</i> 2. Sistem menampilkan form dengan informasi yang dibutuhkan adalah nilai <i>alpha</i> (<i>dropdown</i>) dan <i>beta</i> (<i>dropdown</i>) 3. Aktor mengisi nilai <i>alpha</i> (<i>dropdown</i>) dan <i>beta</i> (<i>dropdown</i>) dan menekan tombol <i>Add Parameter Value</i>
<i>Alternative Flow</i>	1. Jika nilai <i>alpha</i> dan <i>beta</i> tidak sesuai dengan ketentuan, maka sistem akan menampilkan peringatan “ <i>Parameter Range only 0.1 to 1 – Please Try Again</i> ”
<i>Post-Condition</i>	Sistem menampilkan peringatan “ <i>New Parameter Value Submitted!</i> ”

Fungsionalitas selanjutnya yang terdapat pada sistem ialah fungsionalitas lihat informasi prediksi pendapatan. Melalui fungsionalitas ini, *administrator* dapat melihat nilai *alpha* dan *beta* saat ini serta visualisasi prediksi untuk 4 pekan yang akan datang. *Use case scenario* lihat informasi prediksi pendapatan dapat dilihat pada Tabel 4.74.

Tabel 4.74 Use Case Scenario Lihat Informasi Prediksi Pendapatan

<i>Flow of Events</i> Lihat Nilai Informasi Prediksi Pendapatan	
Kode Kebutuhan	POSEW-F-45
<i>Objective</i>	<i>Use case</i> ini berguna untuk menampilkan nilai <i>presentase error</i> prediksi pendapatan.
<i>Actor</i>	<i>Administrator</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>MiningPredict</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan nilai <i>alpha</i> dan <i>beta</i> saat ini serta visualisasi prediksi pendapatan untuk 4 pekan yang akan datang dalam visualisasi grafik <i>area chart</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan nilai <i>alpha</i> dan <i>beta</i> saat ini serta visualisasi prediksi pendapatan untuk 4 pekan yang akan datang dalam visualisasi grafik <i>area chart</i> .

Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitas atur nilai parameter kombinasi produk. Melalui fungsionalitas ini, *administrator* dapat mengatur nilai *minimum support*. *Use case scenario* atur nilai parameter produk potensial dapat dilihat pada Tabel 4.75 dan Tabel 4.76.

Tabel 4.75 Use Case Scenario Atur Nilai Parameter Produk Potensial

<i>Flow of Events</i> Atur Nilai Parameter Produk Potensial	
Kode Kebutuhan	POSEW-F-46
<i>Objective</i>	<i>Use case</i> ini berguna untuk menetapkan nilai <i>minimum support</i>
<i>Actor</i>	<i>Administrator</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>MiningTrends</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol <i>Set Min. Support Value</i>

Tabel 4.76 Use Case Scenario Atur Nilai Parameter Produk Potensial (Lanjutan)

<i>Flow of Events</i> Atur Nilai Parameter Produk Potensial	
<i>Main Flow</i>	<ol style="list-style-type: none"> 2. Sistem menampilkan <i>form Set Parameter Value</i> dengan masukan data yang dibutuhkan adalah nilai <i>minimum support</i> 3. Aktor mengisi nilai <i>minimum support</i> dan menekan tombol <i>Add Parameter Value</i>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika nilai <i>minimum support</i> tidak sesuai dengan ketentuan, maka sistem akan menampilkan peringatan "<i>Please Insert The Right Value</i>" 2. Jika nilai <i>minimum support</i> diisi dengan 0 maka sistem akan menampilkan peringatan "<i>Parameter Value Must Be Not 0 – Please Try Again !</i>" 3. Jika nilai <i>minimum support</i> diisi lebih dari 100 maka sistem akan menampilkan peringatan "<i>Parameter Value Must Be Less Than 100% - Please Try Again !</i>"
<i>Post-Condition</i>	Sistem menampilkan peringatan " <i>Sistem menampilkan peringatan "New Parameter Value Submitted !</i> "

Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitas lihat daftar produk potensial. Melalui fungsionalitas ini, *administrator* dapat melihat daftar produk potensial yang telah melewati seleksi nilai *minimum support*. *Use case scenario* lihat daftar produk potensial dapat dilihat pada Tabel 4.77 dan Tabel 4.78.

Tabel 4.77 Use Case Scenario Lihat Daftar Produk Potensial

<i>Flow of Events</i> Lihat Daftar Produk Potensial	
Kode Kebutuhan	POSEW-F-47
<i>Objective</i>	<i>Use case</i> ini berguna untuk menampilkan daftar produk potensial.
<i>Actor</i>	<i>Administrator</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>MiningTrends</i> .

Tabel 4.78 Use Case Scenario Lihat Daftar Produk Potensial (Lanjutan)

<i>Flow of Events</i> Lihat Daftar Produk Potensial	
<i>Main Flow</i>	1. Sistem menampilkan daftar produk potensial dengan informasi yang disajikan adalah nama produk dan jumlah produk terjual.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan daftar produk potensial dengan informasi yang disajikan adalah nama produk dan jumlah produk terjual.

Fungsionalitas lainnya yang terdapat pada sistem ialah fungsionalitas cari produk potensial. *Use case scenario* cari produk potensial dapat dilihat pada Tabel 4.79.

Tabel 4.79 Use Case Scenario Cari Produk Potensial

<i>Flow of Events</i> Cari Produk Potensial	
Kode Kebutuhan	POSEW-F-48
<i>Objective</i>	<i>Use case</i> ini berguna untuk mencari produk potensial.
<i>Actor</i>	<i>Administrator</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>MiningTrends</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mencari produk potensial pada <i>search area</i>. 2. Sistem menampilkan data produk potensial yang sesuai dengan masukan data yang diberikan aktor.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan data produk potensial yang sesuai dengan masukan data yang diberikan aktor.

Use case scenario urut produk potensial dapat dilihat pada Tabel 4.80 dan Tabel 4.81.

Tabel 4.80 Use Case Scenario Urut Produk Potensial

<i>Flow of Events</i> Urut Produk Potensial	
Kode Kebutuhan	POSEW-F-49

Tabel 4.81 *Use Case Scenario* Urut Produk Potensial (Lanjutan)

<i>Flow of Events</i> Urut Produk Potensial	
<i>Objective</i>	<i>Use case</i> ini berguna untuk mengurutkan kombinasi produk.
<i>Actor</i>	<i>Administrator</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>MiningTrends</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan <i>title</i> pada salah satu kolom produk potensial. 2. Sistem menampilkan data produk potensial dengan urutan yang dimulai dari posisi terendah hingga tertinggi.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika aktor menekan <i>title</i> sebanyak dua kali pada salah satu kolom produk potensial, maka sistem akan menampilkan data produk potensial dengan urutan yang dimulai dari posisi tertinggi hingga terendah.
<i>Post-Condition</i>	Sistem menampilkan data produk potensial dengan urutan yang dimulai dari posisi terendah hingga tertinggi.

Fungsionalitas ialah fungsionalitas lihat informasi produk potensial. *Use case scenario* lihat informasi produk potensial dapat dilihat pada Tabel 4.82 dan Tabel 4.83.

Tabel 4.82 *Use Case Scenario* Lihat Informasi Produk Potensial

<i>Flow of Events</i> Lihat Informasi Produk Potensial	
Kode Kebutuhan	POSEW-F-50
<i>Objective</i>	<i>Use case</i> ini berguna untuk menampilkan informasi produk potensial.
<i>Actor</i>	<i>Administrator</i>
<i>Pre-condition</i>	Sistem sudah menampilkan halaman <i>MiningTrends</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan informasi produk potensial dengan informasi yang disajikan adalah nilai <i>minimum support</i>, nilai batas minimum, dan jumlah produk potensial saat ini.

Tabel 4.83 Use Case Scenario Lihat Informasi Produk Potensial (Lanjutan)

<i>Flow of Events</i> Lihat Informasi Produk Potensial	
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan informasi produk potensial dengan informasi yang disajikan adalah nilai <i>minimum support</i> , nilai batas minimum, dan jumlah produk potensial saat ini.



BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini, akan dijelaskan mengenai perancangan dan implementasi sistem *point of sale* pada Edd's Waffle. Tahapan perancangan dan implementasi sistem dilakukan setelah tahapan analisis kebutuhan selesai dilakukan. Pada tahapan perancangan sistem, akan dilakukan perancangan arsitektur, komponen, dan antarmuka. Di dalam tahapan perancangan arsitektur sendiri, akan dilakukan perancangan yang meliputi *sequence diagram* dan *class diagram* yang terdiri dari perancangan umum dan detail.

Sedangkan tahapan yang dilakukan pada tahap implementasi adalah tahapan yang dilakukan setelah tahapan perancangan selesai dilakukan. Implementasi dilakukan berdasarkan hasil dari analisis dan perancangan. Pada tahapan ini akan menjelaskan mengenai implementasi yang telah dilakukan oleh peneliti. Setidaknya, terdapat empat jenis implementasi yang dilakukan. Implementasi tersebut berupa spesifikasi sistem, implementasi kode program, implementasi basis data hingga implementasi antarmuka sistem.

5.1 Perancangan Sistem

Perancangan sistem merupakan tahapan yang ada di pengembangan perangkat lunak. Tahapan ini dilakukan setelah tahapan analisis kebutuhan. Perancangan sistem dilakukan berdasarkan hasil dari analisa dan rekayasa kebutuhan. Pada penelitian ini, perancangan sistem akan dibagi menjadi tiga bagian. Tiga bagian tersebut adalah perancangan arsitektur, perancangan komponen dan perancangan antarmuka. Hasil dari perancangan sistem ini akan dijadikan dasar implementasi sistem.

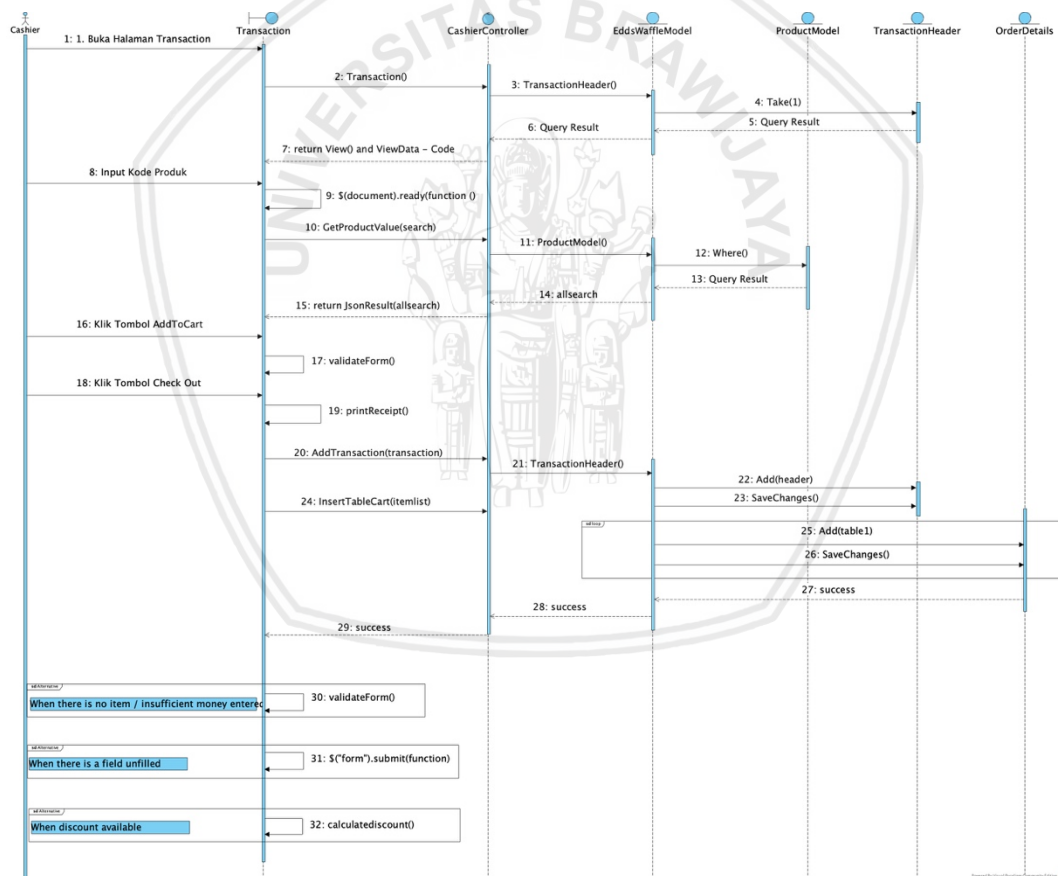
5.1.1 Perancangan Arsitektur

Pada perancangan arsitektur ini akan dijelaskan secara rinci mengenai *sequence diagram* dan *class diagram*. *Sequence diagram* menjelaskan alur pesan antar objek dalam sistem. Lebih lanjut, terdapat tiga *sequence diagram* yang akan dijelaskan dimana ketiganya mewakili proses utama dari sistem *point of sale* Edd's Waffle. Ketiga proses tersebut yakni menambahkan transaksi, mendapatkan prediksi pendapatan, dan mendapatkan rekomendasi kombinasi produk. Selain itu, juga akan dijelaskan mengenai *class diagram* yang menggambarkan *class-class* beserta hubungannya sebagai penyusun sistem.

5.1.1.1 Sequence Diagram Tambah Transaksi

Sequence diagram tambah transaksi ditunjukkan pada Gambar 5.1. *Sequence diagram* ini menunjukkan interaksi objek di dalam sistem saat melakukan proses menambahkan transaksi. Dalam Gambar 5.1 ditunjukkan bahwa terdapat enam objek yang terlibat. Objek tersebut terdiri dari satu *boundary* yaitu *Transaction*, satu *controller* yakni *CashierController*, dan empat model yaitu *EddsWaffleModel*, *ProductModel*, *TransactionHeader*, dan *OrderDetails*. Pada Gambar 5.1 dideskripsikan perpindahan bagaimana *cashier*

membuka halaman *transaction* yang langsung memuat kode transaksi terkini. Selanjutnya, *cashier* menambahkan *item* yang didapat dengan memanggil *method GetProductValue()* dan mendapatkan data dalam representasi *JavaScript Object Notation* (JSON) yang berisikan data-data mengenai produk yang terkait. Produk terkait dikumpulkan dalam tabel. Saat proses transaksi ditambahkan dengan cara menekan tombol *check out*, *boundary Transaction* memanggil fungsi *printReceipt()* di dalam *boundary* tersebut sendiri yang bertipe *javascript*. Transaksi disimpan dalam basis data yang di representasikan dalam dua model, yakni *TransactionHeader* dan *OrderDetails* melalui perantara *EddsWaffleModel*. Secara umum, transaksi disimpan pada *entity TransactionHeader* sedangkan secara khusus seperti rincian belanja disimpan pada *entity OrderDetails*. Kondisi alternatif dilakukan pada *boundary Transaction* sendiri dalam *javascript* pada fungsi *validateForm()* dan ketika form di submit (*Jquery*) serta kalkulasi diskon pada fungsi *calculatediscount()*.



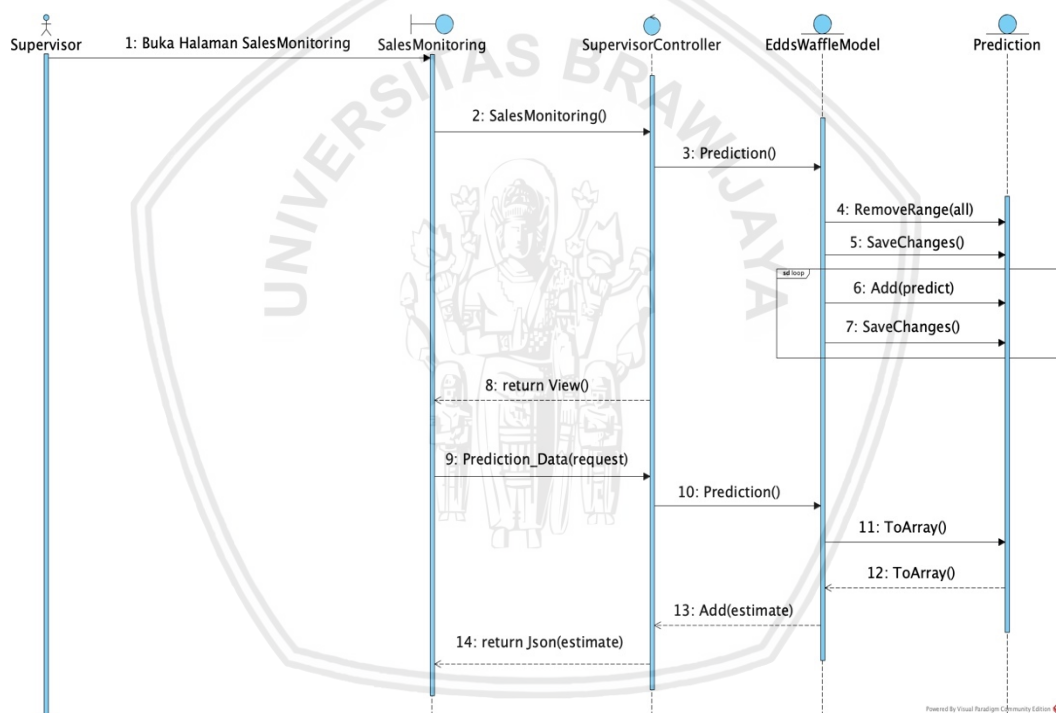
Gambar 5.1 Sequence Diagram Tambah Transaksi

5.1.1.2 Sequence Diagram Lihat Prediksi Pendapatan

Sequence diagram lihat prediksi pendapatan ditunjukkan dalam Gambar 5.2. *Sequence diagram* ini menunjukan interaksi objek di dalam sistem dalam mendapatkan nilai prediksi pendapatan untuk empat pekan yang akan datang, terhitung sejak transaksi terakhir. Terdapat empat objek dimana terdiri dari satu



boundary yaitu *SalesMonitoring*, satu controller yaitu *SupervisorController* dan dua model yaitu *EddsWaffleModel* dan *Prediction*. Untuk menampilkan prediksi pendapatan, dimulai dari boundary *SalesMonitoring* yang memiliki aksi untuk memanggil method *SalesMonitoring* pada *SupervisorController*. Kemudian, dilakukan proses komputasi algoritme dan menghasilkan prediksi untuk empat pekan yang akan datang. Selanjutnya, dilakukan pengosongan pada entity *Predict* dengan menggunakan perintah *RemoveRange* menggunakan parameter *all*. Proses ini dilakukan melalui perantara *EddsWaffleModel*. Lalu, dilakukan penyimpanan dan penambahan nilai prediksi terkini dengan perintah *Add* menggunakan parameter *predict*. Prediksi akan ditampilkan dalam visualisasi *line chart* dengan pemanggilan method *Prediction_Data* dimana boundary *SalesMonitoring* akan mendapatkan visualisasi *chart* melalui JSON yang dikirim oleh *SupervisorController* dengan parameter *estimate*.

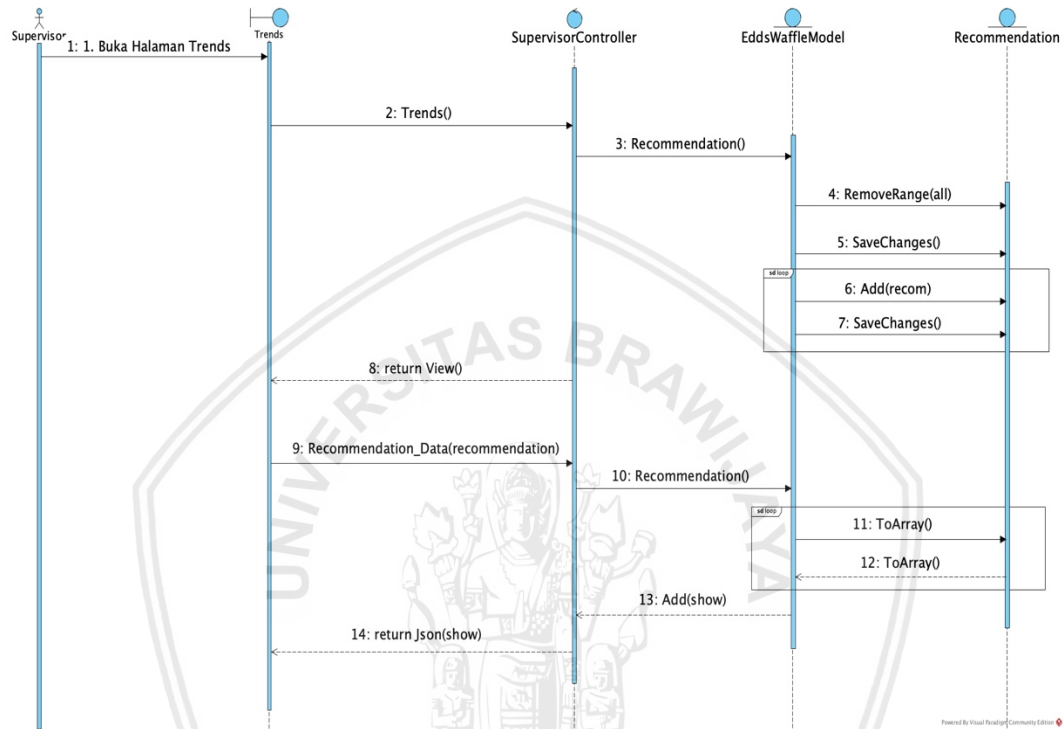


Gambar 5.2 Sequence Diagram Lihat Prediksi Pendapatan

5.1.1.3 Sequence Diagram Lihat Kombinasi Produk

Sequence diagram lihat kombinasi produk dapat dilihat pada Gambar 5.3. Sequence diagram ini menunjukkan interaksi objek di dalam sistem dalam mendapatkan kombinasi produk yang bertemu dalam satu transaksi. Terdapat empat objek dimana terdiri dari satu boundary yaitu *Trends* satu controller yaitu *SupervisorController* dan dua model yaitu *EddsWaffleModel* dan *Recommendation*. Untuk menampilkan kombinasi produk, dimulai dari boundary *Trends* yang saat di buka memanggil method *Trends*. Pada method ini, dilakukan pengosongan pada model *Recommendation* dengan method *RemoveRange()* dan penambahan data terkini melalui proses yang terjadi pada komputasi algoritme di

SupervisorController. Kemudian, dilakukan penambahan pada *model Recommendation* melalui penggunaan *method Add()* dengan parameter *recom*. Proses pengosongan dan penambahan dilakukan melalui perantara *EddsWaffleModel*. Selanjutnya *boundary Trends* akan ditampilkan dan memanggil *method Recommendation_Data*. Kemudian, *method Recommendation_Data* mengambil data yang dibutuhkan dari *model Recommendation* dan menerima data-data dalam representasi JSON dengan parameter *show*.



Gambar 5.3 Sequence Diagram Lihat Kombinasi Produk

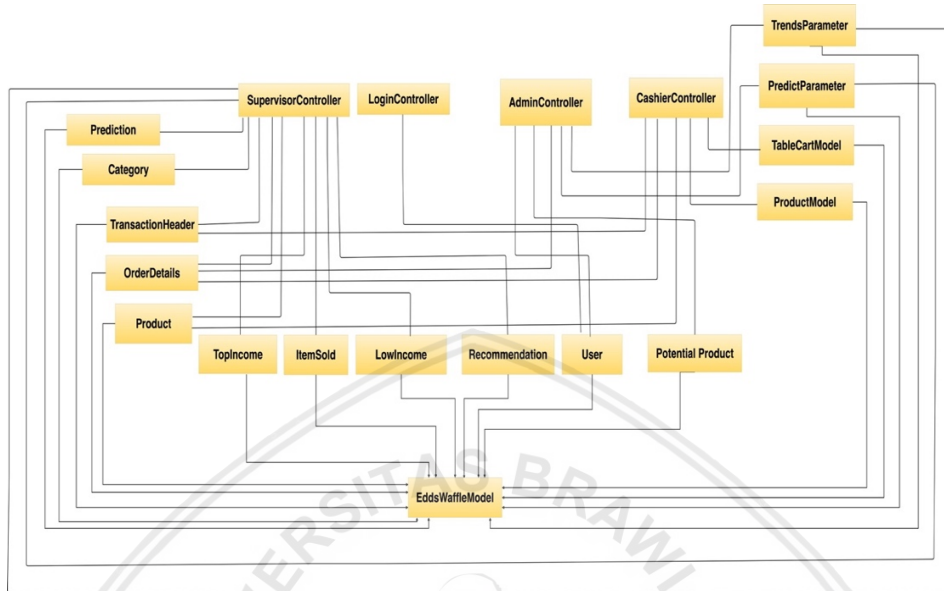
5.1.1.4 Class Diagram

Perancangan *class diagram* merepresentasikan *class* demi *class* yang terdapat pada sistem. perancangan ini dilakukan untuk menggambarkan objek-objek yang terbentuk dan relasi antara objek-objek tersebut. Dalam perancangan *class diagram* terdapat perancangan umum dan perancangan detail. Perancangan umum menggambarkan relasi antar kelas. Sedangkan pada perancangan detail, menggambarkan atribut dan operasi yang ada pada *class*.

1. Perancangan Umum

Perancangan umum dilakukan untuk mendeskripsikan objek-objek yang terbentuk beserta relasi-relasinya. Pada perancangan umum, di deskripsikan hubungan antar *class* yang melibatkan bagian dari *controller* yang meliputi *SupervisorController*, *LoginController*, *AdminController*, dan *CashierController*. Pada bagian *model* terdapat *Prediction*, *Category*, *TransactionHeader*, *OrderDetails*, *Product*, *TopIncome*, *ItemSold*, *LowIncome*, *Recommendation*, *User*, *PotentialProduct*, *ProductModel*, *TableCartModel*, *PredictParameter*, dan

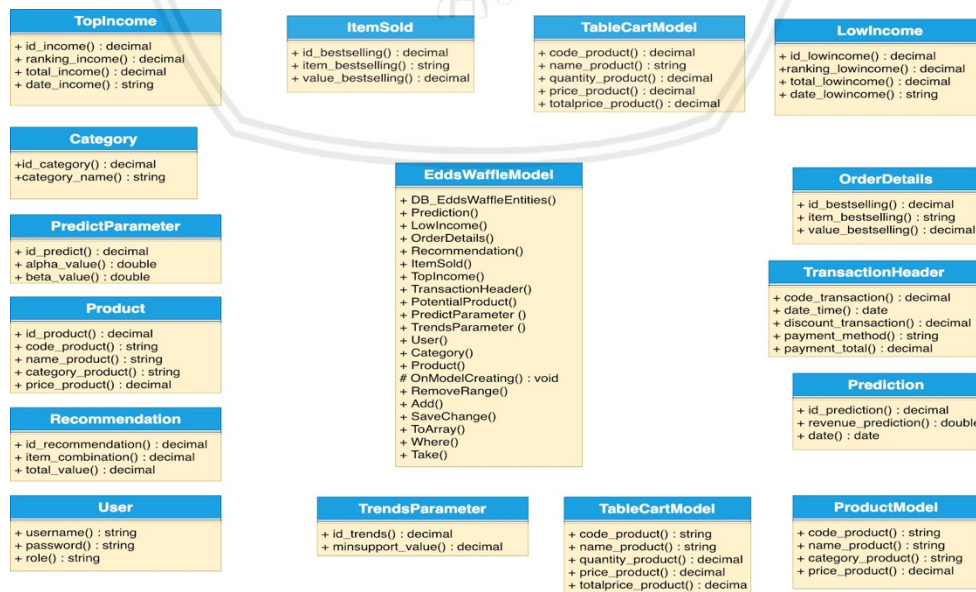
TrendsParameter. Pada bagian *model*, dilakukan generalisasi kepada *EddsWaffleModel*. Pemodelan *class diagram* pada perancangan umum digambarkan melalui Gambar 5.4.



Gambar 5.4 *Class Diagram* Perancangan Umum

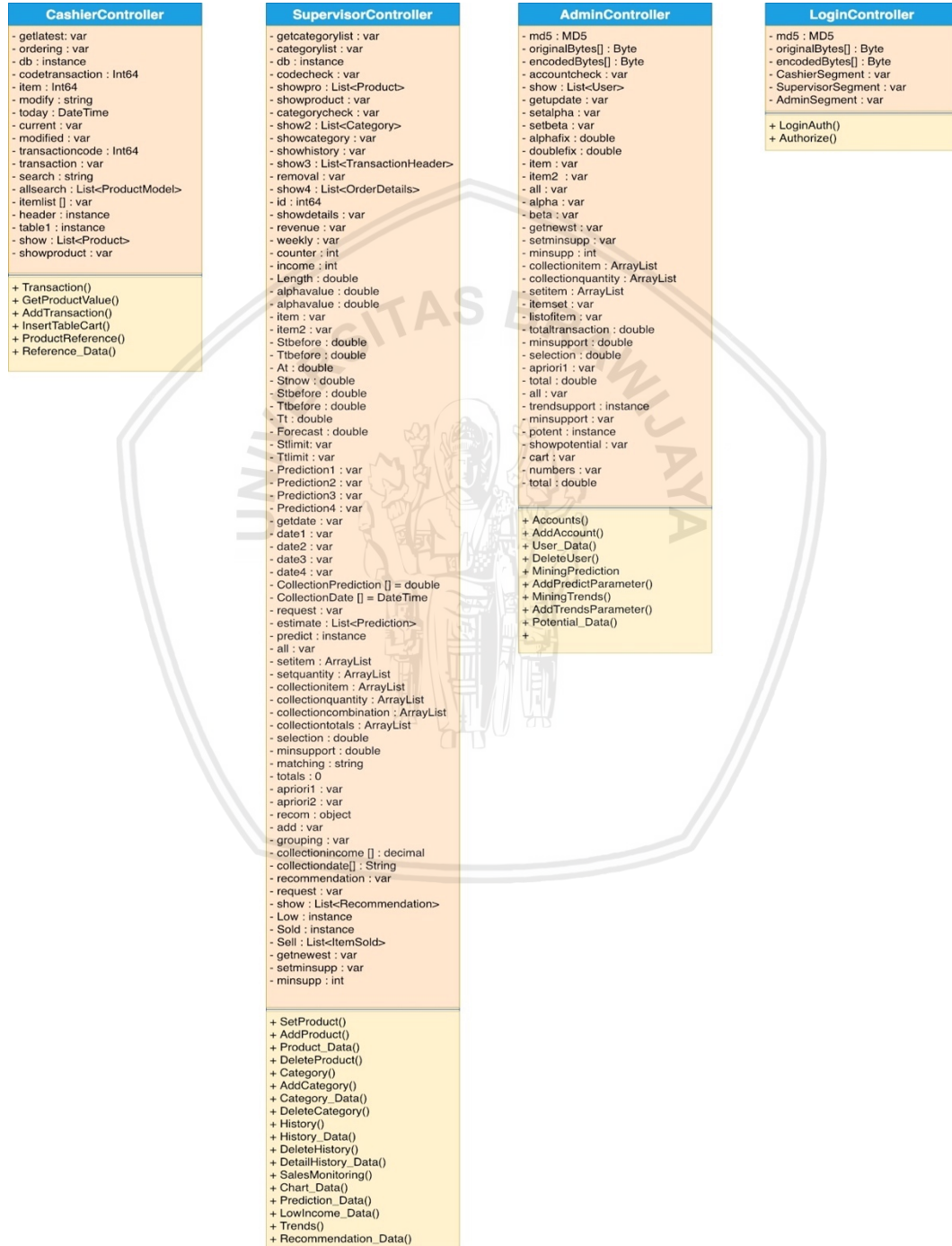
2. Perancangan *Detail*

Perancangan *detail* dilakukan untuk mendefinisikan atribut dan operasi yang terdapat pada setiap *class* yang dirancang pada sistem. Pada perancangan *detail*, ditujukan pada bagian *model* dan *controller*. Pada bagian *model* terdapat enam belas *class*. Perancangan detail yang melibatkan atribut dan operasi pada *class-class* dalam *model* dapat dilihat pada Gambar 5.5.



Gambar 5.5 *Class Diagram* pada *Model*

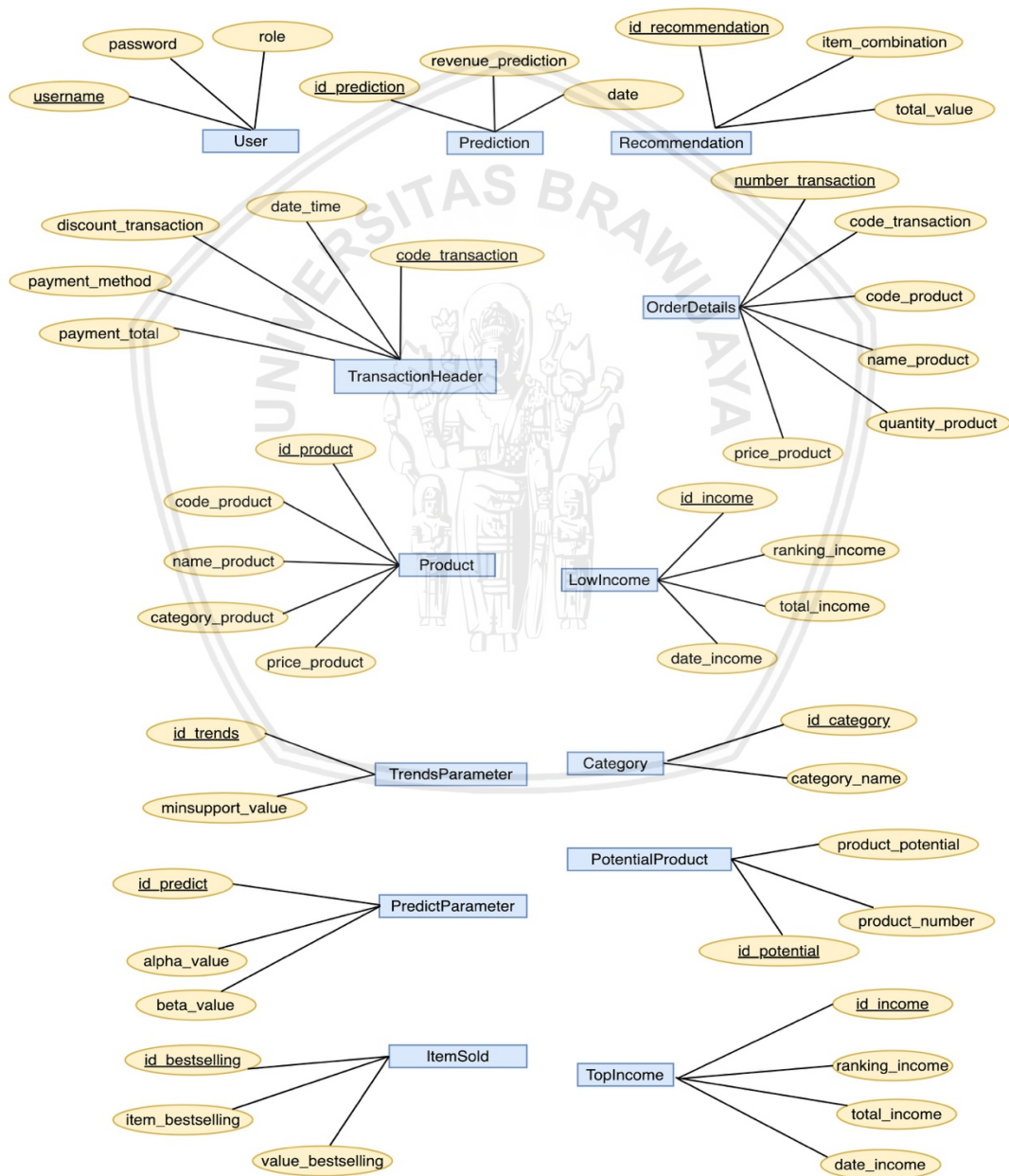
Sedangkan pada bagian *controller* terdapat empat *class* yang meliputi *SupervisorController*, *LoginController*, *AdminController*, dan *CashierController*. Perancangan detail yang melibatkan atribut dan operasi pada *class-class* dalam *controller* dapat dilihat pada Gambar 5.6. Pada perancangan *detail*, di deskripsikan tipe data dari setiap atribut. Selain itu, di deskripsikan pula *access modifiers* yang menandakan bahwa atribut atau operasi terkait bersifat *public*, *private*, maupun *protected*.



Gambar 5.6 Class Diagram pada Controller

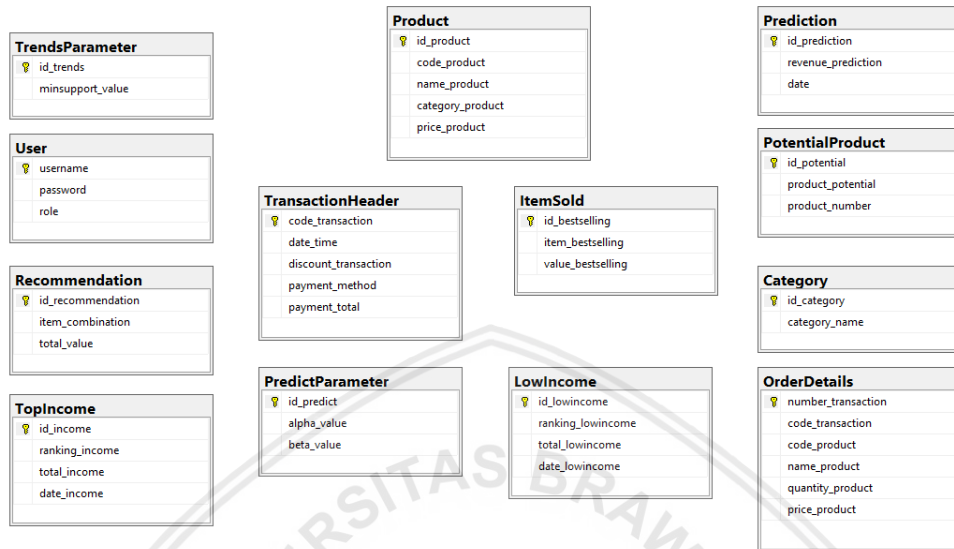
5.1.2 Perancangan Data

Perancangan data dari sistem *point of sale* pada Edd's Waffle akan dibagi menjadi 2 bagian. Perancangan pertama akan dijelaskan pada pembuatan *Entity Relational Diagram* (ERD) seperti pada Gambar 5.7. Pada perancangan ERD, terdapat tiga belas entitas. Tidak terdapat relasi pada ERD, hal ini dikarenakan penggunaan *framework* dalam pembuatan sistem yaitu ASP.NET MVC dan *entity framework* yang memiliki kemampuan untuk melakukan *query expression* dalam sebuah variabel pada LINQ (*Language Integrated Query*). Secara singkat, LINQ dapat menyimpan data pada *collection*, sintaks pada basis data, hingga penyaringan data.



Gambar 5.7 Perancangan ERD

Setelah dilakukan perancangan ERD, akan dirancang *Physical Data Model* (PDM) seperti yang dapat dilihat pada Gambar 5.8. Pada perancangan PDM, jumlah tabel yang dibuat sama dengan jumlah entitas yang telah direncanakan pada ERD.



Gambar 5.8 Perancangan PDM

5.1.3 Perancangan Komponen

Pada perancangan komponen akan memaparkan rincian sub-sistem dari setiap komponen pada perangkat lunak. Pada perancangan komponen akan memaparkan alur algoritme. Pada penelitian ini, akan memaparkan tiga algoritme *method* dalam sistem yang dikembangkan. Ketiga *method* tersebut terlibat dalam perancangan tiga fitur utama pada sistem. Tiga *method* tersebut adalah *method AddTransaction*, *method Prediction_Data* dan *method Recommendation_Data*.

5.1.3.1 Perancangan Komponen *Method InsertTableCart*

Pada Tabel 5.1 dan Tabel 5.2 memaparkan algoritme *method InsertTableCart* yang digunakan untuk menambahkan transaksi pada basis data tabel *OrderDetails*. Dari algoritme ini, dilakukan inialisasi variabel yang relevan. Kemudian dilakukan perulangan untuk mengambil nilai kode transaksi terakhir. Saat proses penyusunan nilai kode transaksi selesai, dilakukan perulangan untuk memasukkan seluruh *item* pada transaksi pada tabel *OrderDetails*. Setelah sukses, pesan transaksi berhasil dikembalikan dalam representasi *Json*.

Nama *class*: *CashierController*

Nama *method*: *InsertTableCart*

Tabel 5.1 *Pseudocode* algoritme *method InsertTableCart*

Pseudocode algoritme <i>method InsertTableCart</i>	
1	var db -> Initiate EddsWaffleEntites database
2	var table1 -> Initiate OrderDetails table from database
3	var getlatest -> Initiate the list of Transaction Header table
4	from database

Tabel 5.2 Pseudocode algoritme method *InsertTableCart* (Lanjutan)

Pseudocode algoritme <i>method InsertTableCart</i>	
5	var ordering -> take 1 row from code_transaction column from getlatest, ordered by descending
6	Int64 codetransaction -> 0
7	Foreach var item in ordering
8	Set int codetransaction = (Int64) item
9	End foreach
10	Int64 modifycode = codetransaction + 1
11	String modify = make modifycode to string
12	today -> Initiate current date
13	var current -> formatting today in yy-mm-dd and to string
14	var modified -> initiate StringBuilder for modify
15	modified remove 8 characters started from character number 1
16	modified insert current started from character number 1
17	modify -> modified to string
18	Int64 transactioncode -> Convert modify to Int64
19	Foreach TableCartModel I in itemlist
20	code_transaction from table1 = transactioncode
21	code_product from table1 = code_product from i
22	name_product from table1 = name_product from i
23	quantity_product from table1 = quantity_product from i
24	price_product from table1 = totalprice_product from i
25	db in OrderDetails table from database add table1
26	db make a save changes
27	End foreach
28	Return "Transaction Successfully Added!" in JSON

5.1.3.2 Perancangan Komponen *Method Prediction_Data*

Pada Tabel 5.3 memaparkan algoritme *method Prediction_Data*. *Method ini* digunakan untuk memanggil nilai-nilai pada tabel basis data *Prediction* agar ditampilkan dalam visualisasi *line chart*. Di dalam algoritme *method Prediction_Data* terdapat sebuah seleksi kondisi sebanyak jumlah nilai pada tabel basis data *Prediction*. Data diambil dan dikembalikan dalam representasi *Json* sebagai nilai nilai pada grafik prediksi pendapatan.

Nama *class*: *SupervisorController*

Nama *method*: *Prediction_Data*

Tabel 5.3 Pseudocode algoritme *method Prediction_Data*

Pseudocode algoritme <i>method Prediction_Data</i>	
1	db -> Initiate EddsWaffleEntites database
2	Var Forecasting -> Get Prediction database in array
3	Estimate -> Initiate the list of Prediction database
4	For int i from 0 until number of Prediction database row count
5	Estimate add Forecasting in each element
6	End for
7	Return estimate in JSON



5.1.3.3 Perancangan Komponen *Method Recommendation_Data*

Pada Tabel 5.4 memaparkan algoritme *method Recommendation_Data* pada *class SupervisorController* untuk menampilkan daftar kombinasi produk yang di rekomendasikan. Melalui algoritme ini, dilakukan inialisasi basis data dan variabel relevan lainnya. Terdapat sebuah seleksi kondisi sebanyak jumlah nilai pada tabel basis data *Recommendation*. Data diambil dan dikembalikan dalam representasi *Json*. *Json* yang berisikan data-data prediksi tersebut nantinya akan di visualisasi kan pada *table grid*.

Nama *class*: *SupervisorController*

Nama *method*: *Recommendation_Data*

Tabel 5.4 Pseudocode Algoritme *Recommendation_Data*

Pseudocode algoritme <i>method Recommendation_Data</i>	
1	db -> Initiate EddsWaffleEntites database
2	Var showrecommendation -> Get Recommendation database in array
3	show -> Initiate the list of Recommendation database
4	For int i from 0 until number of Recommendation database row count
5	show add showrecommendation in each element
6	End for
7	Return recommendation in JSON

5.1.4 Perancangan Antarmuka

Perancangan antarmuka merupakan rancangan tampilan yang akan digunakan sebagai sarana interaksi antara pengguna dengan sistem. Dalam perancangan antarmuka ini akan memaparkan perancangan *layout* sehingga memberikan gambaran perkiraan antarmuka yang nantinya akan tampak pada sistem. Hasil dari perancangan antarmuka akan digunakan sebagai dasar implementasi antarmuka pada sistem. Pada penelitian ini akan memaparkan tiga perancangan antarmuka. Perancangan antarmuka yang digunakan dalam penelitian ini dapat dipaparkan dalam sub bab 5.1.4.1 sampai sub bab 5.1.4.3.

5.1.4.1 Perancangan Antarmuka Halaman Transaksi

Perancangan antarmuka halaman antarmuka ditunjukkan dalam Gambar 5.10. Pada Gambar 5.10, diilustrasikan bahwa pada perancangan antarmuka halaman transaksi terdapat 26 komponen. Komponen tersebut diantaranya adalah *button*, *header*, *table*, *label*, *textfield*, dan *dropdown*. Komponen-komponen tersebut akan dijelaskan pada Tabel 5.5 dan Tabel 5.6. Perancangan antarmuka halaman transaksi digunakan sebagai media bagi *cashier* untuk menambahkan transaksi. Perancangan antarmuka pada halaman transaksi akan dijadikan sebagai acuan atau pedoman dalam melakukan implementasi antarmuka halaman transaksi.



Gambar 5.9 Perancangan Antarmuka Halaman Transaksi

Tabel 5.5 Keterangan Perancangan Antarmuka Halaman Transaksi

No.	Nama Objek	Tipe	Keterangan
1	WELCOME, CASHIER	Header	Header untuk menerangkan segmen pengguna.
2	Product Code	Textfield	Textfield untuk mengisi kode produk.
3	Product Quantity	Textfield	Textfield untuk mengisi kuantitas produk.
4	Product Name	Textfield	Textfield untuk nama produk.
5	Price	Textfield	Textfield untuk harga produk.
6	Total Price	Textfield	Textfield untuk total harga produk.
7	Add To Cart	Button	Button untuk menambahkan item.
8	Transaction	Button	Button untuk halaman transaksi.
9	References	Button	Button untuk referensi produk.
10	Sign Out	Button	Button untuk logout atau keluar.
11	Table daftar pesanan	Table	Table untuk daftar pesanan.
12	Remove	Button	Button untuk menghapus item.
13	No. Transaction	Label	Label nomor transaksi.
14	Code Transaction	Textfield	Textfield untuk kode transaksi.
15	Discount (%)	Label	Label untuk diskon.

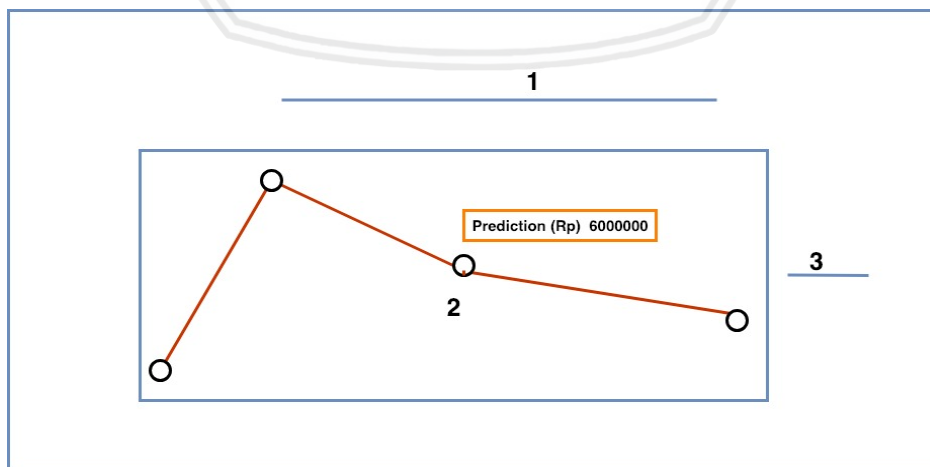


Tabel 5.6 Keterangan Perancangan Antarmuka Halaman Transaksi (Lanjutan)

No.	Nama Objek	Tipe	Keterangan
16	<i>Discount</i>	<i>Textfield</i>	<i>Textfield</i> untuk diskon.
17	<i>Payment Method</i>	<i>Label</i>	<i>Label</i> untuk metode pembayaran.
18	<i>Payment Method</i>	<i>Dropdown</i>	<i>Dropdown</i> untuk metode pembayaran.
19	<i>Sub Total</i>	<i>Label</i>	<i>Label</i> untuk total harga.
20	<i>Sub Total</i>	<i>Textfield</i>	<i>Textfield</i> untuk total harga.
21	<i>Paid</i>	<i>Label</i>	<i>Label</i> untuk jumlah uang tunai dibayarkan.
22	<i>Paid</i>	<i>Textfield</i>	<i>Textfield</i> untuk jumlah uang tunai dibayarkan.
23	<i>Change</i>	<i>Label</i>	<i>Label</i> untuk jumlah kembalian.
24	<i>Change</i>	<i>Textfield</i>	<i>Textfield</i> untuk jumlah kembalian.
25	<i>Reset</i>	<i>Button</i>	<i>Button</i> untuk memulai ulang transaksi.
26	<i>Check Out</i>	<i>Button</i>	<i>Button</i> untuk melakukan finalisasi transaksi.

5.1.4.2 Perancangan Antarmuka Prediksi Pendapatan

Perancangan antarmuka Prediksi pendapatan dapat dilihat pada Gambar 5.11. Dalam Gambar 5.11 diilustrasikan bahwa pada perancangan antarmuka prediksi pendapatan terdapat 3 komponen. Komponen tersebut ialah 1 header, 1 *line chart*, dan 1 *label*. Komponen-komponen tersebut akan dijelaskan pada Tabel 5.7. Antarmuka prediksi pendapatan digunakan untuk menampilkan prediksi pendapatan untuk 4 pekan yang akan datang dalam visualisasi *line chart*.



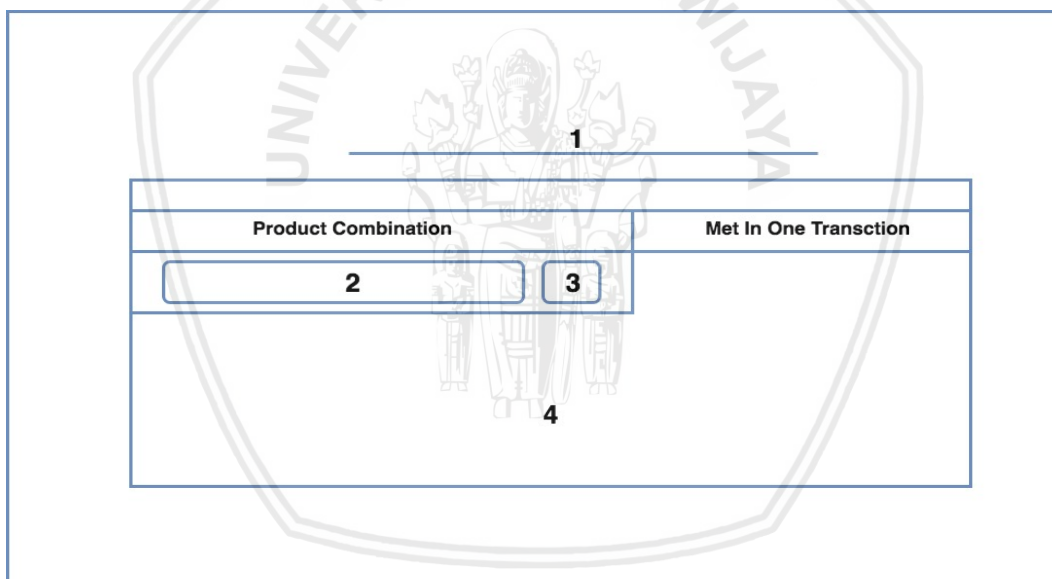
Gambar 5.10 Perancangan Antarmuka Prediksi Pendapatan

Tabel 5.7 Keterangan Perancangan Antarmuka Prediksi Pendapatan

No.	Nama Objek	Tipe	Keterangan
1	<i>Weekly Revenue Prediction</i>	<i>Header</i>	<i>Header yang menjelaskan bagian prediksi pendapatan.</i>
2	<i>Prediction Movement</i>	<i>Line Chart</i>	<i>Chart yang memuat pergerakan prediksi pendapatan.</i>
3	<i>Prediction (Rp)</i>	<i>Label</i>	<i>Label yang menjelaskan keterangan prediksi.</i>

5.1.4.3 Perancangan Antarmuka Kombinasi Produk

Perancangan antarmuka kombinasi produk dapat dilihat pada gambar 5.7. Pada perancangan antarmuka kombinasi produk terdapat 4 komponen. Komponen tersebut ialah header, *search area*, *button*, dan *table grid*. Penjelasan mengenai setiap komponen terkait dapat dilihat pada Tabel 5.6. Perancangan ini ditujukan untuk menjadi acuan dalam hal menampilkan kombinasi produk.



Gambar 5.11 Perancangan Antarmuka Kombinasi Produk

Tabel 5.8 Keterangan Perancangan Antarmuka Kombinasi Produk

No.	Nama Objek	Tipe	Keterangan
1	<i>Product Combination Analysis</i>	<i>Header</i>	<i>Header yang menjelaskan bagian kombinasi pendapatan.</i>
2	<i>Search Product</i>	<i>Search Area</i>	<i>Search Area untuk melakukan pencarian kombinasi produk.</i>



Tabel 5.9 Keterangan Perancangan Antarmuka Kombinasi Produk (Lanjutan)

No.	Nama Objek	Tipe	Keterangan
3	<i>Icon Filter</i>	<i>Button</i>	<i>Button</i> untuk melakukan <i>filter</i> pencarian.
4	<i>Product Combination</i> dan <i>Met In One Transaction</i>	<i>Table Grid</i>	<i>Table</i> visualisasi deretan kombinasi produk dan jumlah kombinasi produk bertemu.

5.2 Implementasi Sistem

Tahap implementasi adalah tahap yang dilakukan setelah tahap perancangan. Tahap implementasi dilakukan berdasarkan hasil dari tahap rekayasa kebutuhan dan tahap perancangan. Dalam pengimplementasian mengandung kode program yang didasarkan pada perancangan algoritme. Kemudian implementasi basis data yang di dasarkan pada perancangan data. Selain itu, juga mengandung hasil implementasi antarmuka pengguna yang sudah didefinisikan di tahap perancangan *layout*. Tahap implementasi pada penelitian ini akan dipaparkan mengenai spesifikasi sistem, implementasi kode program, implementasi basi data dan implementasi antarmuka.

5.2.1 Spesifikasi Sistem

Pada bagian spesifikasi sistem akan memaparkan spesifikasi perangkat keras dan perangkat lunak yang digunakan untuk pengembangan perangkat lunak. Pada spesifikasi perangkat keras akan dipaparkan mengenai besaran *processor*, *secondary memory*, dan *main memory* yang digunakan. Sedangkan untuk spesifikasi perangkat lunak akan dipaparkan mengenai sistem operasi, bahasa pemrograman, IDE dan perangkat penunjang lainnya yang digunakan. Spesifikasi sistem perangkat keras yang digunakan pada penelitian ini dapat ditunjukkan pada Tabel 5.10. Sedangkan untuk spesifikasi perangkat lunak ditunjukkan pada Tabel 5.11 dan Tabel 5.12.

Tabel 5.10 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
<i>Processor</i>	Intel® Core™ i5-5200U CPU @ 2.20GHz
<i>Solid State Drive</i>	120 GB
<i>Main Memori (RAM)</i>	8192 MB DDR3

Tabel 5.11 Spesifikasi Perangkat Lunak

Nama Komponen	Spesifikasi
<i>Operating System</i>	Windows 10
<i>Programming Language</i>	C#

Tabel 5.12 Spesifikasi Perangkat Lunak (Lanjutan)

Nama Komponen	Spesifikasi
<i>Text Editor / IDE</i>	Microsoft Visual Studio 2017 <i>Community Editions 2017 Version 1.8.2-beta.4</i>
<i>Framework</i>	ASP.NET MVC
Editor Dokumentasi	Microsoft® Word 2013
Editor Perancangan	<i>Visual Paradigm Version 15.1 (Build 20181005)</i>

5.2.2 Implementasi Kode Program

Implementasi kode program akan didasarkan pada perancangan algoritme komponen yang telah diterangkan sebelumnya. Pada bagian ini algoritme yang ada di perancangan komponen akan diubah menjadi bahasa pemrograman. Pada penelitian ini bahasa pemrograman yang digunakan adalah bahasa java. Penelitian ini akan memaparkan tiga implementasi method yaitu *method InsertTableCart*, *method Prediction_Data* dan *method Recommendation_Data*. Implementasi kode program akan dipaparkan pada sub bab 5.2.2.1 hingga sub bab 5.2.2.3.

5.2.2.1 Implementasi Kode Program *Method InsertTableCart*

Nama *class*: *SupervisorController*

Nama *method*: *InsertTableCart*

Tabel 5.13 kode program *method InsertTableCart*

Kode sumber <i>method InsertTableCart</i>	
1	<code>DB_EddsWaffleEntities db = new DB_EddsWaffleEntities();</code>
2	<code>OrderDetails table1 = new OrderDetails();</code>
3	<code>var getlatest = db.TransactionHeader.ToList();</code>
4	<code>var ordering = getlatest.OrderByDescending(a =>a.code_transaction)</code>
5	<code>.Select(a => a.code_transaction).Take(1);</code>
6	<code>Int64 codetransaction = 0;</code>
7	<code> foreach (var item in ordering){</code>
8	<code> codetransaction = (Int64)item;</code>
9	<code> }</code>
10	<code>Int64 modifycode = codetransaction + 1;</code>
11	<code>string modify = modifycode.ToString();</code>
12	<code>DateTime today = DateTime.Today;</code>
13	<code>var current = today.ToString("yyyyMMdd");</code>
14	<code>var modified = new StringBuilder(modify);</code>
15	<code>modified.Remove(1, 8);</code>
16	<code>modified.Insert(1, current);</code>
17	<code>modify = modified.ToString();</code>
18	<code>Int64 transactioncode = Convert.ToInt64(modify);</code>
19	<code> foreach (TableCartModel i in itemlist){</code>
20	<code> table1.code_transaction = transactioncode;</code>
21	<code> table1.code_product = i.code_product;</code>
22	<code> table1.name_product = i.name_product;</code>
23	<code> table1.quantity_product = i.quantity_product;</code>



Tabel 5.14 kode program *method InsertTableCart* (Lanjutan)

24	<code>table1.price_product = i.totalprice_product;</code>
25	<code>db.OrderDetails.Add(table1);</code>
26	<code>db.SaveChanges();</code>
27	<code>}</code>
28	<code>return Json("Transaction Successfully Added!");</code>

Pada Tabel 5.13 dan Tabel 5.14 memaparkan implementasi *method InsertTableCart* yang digunakan untuk menambahkan transaksi pada basis data tabel *OrderDetails*. Dari algoritme pada *method* ini, dilakukan inisialisasi variabel yang relevan serta inisialisasi model. Kemudian dilakukan perulangan untuk mengambil nilai kode transaksi terakhir. Setelah kode transaksi di dapatkan, dilakukan modifikasi agar kode transaksi di dapat sesuai dengan spesifikasi kebutuhan, proses dilakukan dengan cara memanfaatkan *StringBuilder* dan konversi tipe data. Proses perulangan kemudian di lakukan dengan menyamakan masukan data dengan kolom pada basis data, termasuk untuk kode transaksi yang telah di modifikasi. Selain itu, dilakukan proses penambahan dengan parameter *table1* dan penyimpanan pada setiap perulangan. Setelah sukses, pesan transaksi berhasil dikembalikan dalam representasi JSON (*JavaScript Object Notation*).

5.2.2.2 Implementasi Kode Program *Method Prediction_Data*

Nama *class*: *SupervisorController*

Nama *method*: *Prediction_Data*

Tabel 5.15 kode program *method Prediction_Data*

Kode sumber <i>method Prediction_Data</i>	
1	<code>DB_EddsWaffleEntities db = new DB_EddsWaffleEntities();</code>
2	<code>var Forecasting = db.Prediction.ToArray();</code>
3	<code>List<Prediction> estimate = new List<Prediction>();</code>
4	<code>for (int i = 0; i < db.Prediction.Count(); i++){</code>
5	<code>estimate.Add(Forecasting.ElementAt(i));</code>
6	<code>}</code>
7	<code>return Json(estimate);</code>

Pada Tabel 5.15 memaparkan hasil implementasi *method Prediction_Data*. *Method ini* digunakan untuk memanggil nilai nilai pada tabel basis data *Prediction* agar ditampilkan dalam visualisasi *line chart*. Di dalam algoritme *method Prediction_Data* terdapat inisialisasi *model* dan variabel yang relevan. Selanjutnya, terdapat sebuah seleksi kondisi sebanyak jumlah nilai pada tabel basis data *Prediction*. Data diambil dan ditambahkan pada *list* dengan nama *estimate*. *Estimate* yang sudah di isi dengan data data terkini dari *model Prediction* dikembalikan dalam representasi JSON sebagai nilai nilai pada grafik prediksi pendapatan.



5.2.2.3 Implementasi Kode Sumber *Method Recommendation_Data*

Nama *class*: *SupervisorController*

Nama *method*: *Recommendation_Data*

Tabel 5.16 kode program *method Recommendation_Data*

Kode sumber <i>method Recommendation_Data</i>	
1	DB_EddsWaffleEntities db = new DB_EddsWaffleEntities();
2	var showrecommendation = db.Recommendation.ToArray();
3	List<Recommendation> show = new List<Recommendation>();
4	for (int i = 0; i < db.Recommendation.Count(); i++){
5	show.Add(showrecommendation.ElementAt(i));
6	}
7	return Json(show.ToDataSourceResult(recommendation));

Pada Tabel 5.16 memaparkan hasil implementasi kode program *method Recommendation_Data*. Melalui algoritme ini, dilakukan inisialisasi basis data dan variabel relevan lainnya. Terdapat sebuah seleksi kondisi sebanyak jumlah nilai pada tabel basis data *Recommendation*. Melalui sebuah *list* dengan nama *show*, dilakukan penambahan nilai nilai pada *model Recommendation*. Data pada *list* dengan nama *show* tersebut dikembalikan dalam representasi JSON.

5.2.3 Implementasi Basis Data

5.2.3.1 Implementasi Tabel *Category*

Pada implementasi tabel *Category* dilakukan *Data Definition Language* (DDL) dengan membuat struktur tabel dan mendefinisikan tipe data dari tabel *Category*. DDL diimplementasi dengan menggunakan perintah SQL yaitu *Create Table*. Tabel *category* digunakan untuk menyimpan data kategori dimana terdiri dari *id_category* dan *category_name*. Implementasi tabel *Category* ditunjukkan pada Tabel 5.17.

Tabel 5.17 Implementasi Tabel *Category*

Implementasi Tabel <i>Category</i>	
1	<code>CREATE TABLE [dbo].[Category] (</code>
2	<code>[id_category] [numeric](18, 0) IDENTITY(1,1) NOT NULL,</code>
3	<code>[category_name] [varchar](50) NOT NULL</code>
4	<code>)</code>

5.2.3.2 Implementasi Tabel *ItemSold*

Pada implementasi tabel *ItemSold* dilakukan *Data Definition Language* (DDL) dengan membuat struktur tabel dan mendefinisikan tipe data dari tabel *ItemSold*. DDL diimplementasi dengan menggunakan perintah SQL yaitu *Create Table*. Tabel *ItemSold* digunakan untuk menyimpan data jumlah produk terjual dimana terdiri dari *id_best-selling*, *item_best-selling*, dan *value_best-selling*. Implementasi tabel *ItemSold* ditunjukkan pada Tabel 5.18.

Tabel 5.18 Implementasi Tabel *ItemSold*

Implementasi Tabel <i>ItemSold</i>	
1	<code>CREATE TABLE [dbo].[ItemSold](</code>
2	<code>[id_bestselling] [numeric](18, 0) IDENTITY(1,1) NOT NULL,</code>
3	<code>[item_bestselling] [varchar](50) NULL,</code>
4	<code>[value_bestselling] [numeric](18, 0) NULL</code>
5	<code>)</code>

5.2.3.3 Implementasi Tabel *LowIncome*

Pada implementasi tabel *LowIncome* dilakukan *Data Definition Language* (DDL) dengan membuat struktur tabel dan mendefinisikan tipe data dari tabel *Prediction*. DDL diimplementasi dengan menggunakan perintah SQL yaitu *Create Table*. Tabel *Prediction* digunakan untuk menyimpan data pendapatan terendah dimana terdiri dari *id_lowincome*, *ranking_lowincome*, *total_lowincome* dan *date_lowincome*. Implementasi tabel *LowIncome* ditunjukkan pada Tabel 5.19.

Tabel 5.19 Implementasi Tabel *LowIncome*

Implementasi Tabel <i>LowIncome</i>	
1	<code>CREATE TABLE [dbo].[LowIncome] (</code>
2	<code>[id_lowincome] [numeric](18, 0) IDENTITY(1,1) NOT NULL,</code>
3	<code>[ranking_lowincome] [numeric](18, 0) NULL,</code>
4	<code>[total_lowincome] [numeric](18, 0) NULL,</code>
5	<code>[date_lowincome] [varchar](50) NULL</code>
6	<code>)</code>

5.2.3.4 Implementasi Tabel *OrderDetails*

Pada implementasi tabel *OrderDetails* dilakukan *Data Definition Language* (DDL) dengan membuat struktur tabel dan mendefinisikan tipe data dari tabel *OrderDetails*. DDL diimplementasi dengan menggunakan perintah SQL yaitu *Create Table*. Tabel *OrderDetails* digunakan untuk menyimpan data rincian transaksi dimana terdiri dari *number_transaction*, *code_transaction*, *code_product*, *name_product*, *quantity_product* dan *price_product*. Implementasi tabel *OrderDetails* ditunjukkan pada Tabel 5.20.

Tabel 5.20 Implementasi Tabel *OrderDetails*

Implementasi Tabel <i>OrderDetails</i>	
1	<code>CREATE TABLE [dbo].[OrderDetails] (</code>
2	<code>[number_transaction] [numeric](18, 0) IDENTITY(1,1) NOT NULL,</code>
3	<code>[code_transaction] [numeric](18, 0) NULL,</code>
4	<code>[code_product] [varchar](255) NULL,</code>
5	<code>[name_product] [varchar](255) NULL,</code>
6	<code>[quantity_product] [numeric](18, 0) NULL,</code>
7	<code>[price_product] [numeric](18, 0) NULL</code>
8	<code>)</code>

5.2.3.5 Implementasi Tabel *PotentialProduct*

Pada implementasi tabel *PotentialProduct* dilakukan *Data Definition Language* (DDL) dengan membuat struktur tabel dan mendefinisikan tipe data dari tabel *PotentialProduct*. DDL diimplementasi dengan menggunakan perintah SQL yaitu *Create Table*. Tabel *PotentialProduct* digunakan untuk menyimpan data



produk potensial dimana terdiri dari *id_potential*, *product_potential*, dan *product_number*. Implementasi tabel *PotentialProduct* ditunjukkan pada Tabel 5.21.

Tabel 5.21 Implementasi Tabel *PotentialProduct*

Implementasi Tabel <i>PotentialProduct</i>	
1	<code>CREATE TABLE [dbo].[PotentialProduct] (</code>
2	<code>[id_potential] [numeric](18, 0) IDENTITY(1,1) NOT NULL,</code>
3	<code>[product_potential] [varchar](80) NULL,</code>
4	<code>[product_number] [numeric](18, 0) NULL</code>
5	<code>)</code>

5.2.3.6 Implementasi Tabel *Prediction*

Pada implementasi tabel *Prediction* dilakukan *Data Definition Language* (DDL) dengan membuat struktur tabel dan mendefinisikan tipe data dari tabel *Prediction*. DDL diimplementasi dengan menggunakan perintah SQL yaitu *Create Table*. Tabel *Prediction* digunakan untuk menyimpan data prediksi pendapatan dimana terdiri dari *id_prediction*, *revenue_prediction* dan *date*. Implementasi tabel *Prediction* ditunjukkan pada Tabel 5.22.

Tabel 5.22 Implementasi Tabel *Prediction*

Implementasi Tabel <i>Prediction</i>	
1	<code>CREATE TABLE [dbo].[Prediction] (</code>
2	<code>[id_prediction] [int] NOT NULL,</code>
3	<code>[revenue_prediction] [float] NULL,</code>
4	<code>[date] [date] NULL</code>
5	<code>)</code>

5.2.3.7 Implementasi Tabel *PredictParameter*

Pada implementasi tabel *PredictParameter* dilakukan *Data Definition Language* (DDL) dengan membuat struktur tabel dan mendefinisikan tipe data dari tabel *PredictParameter*. DDL diimplementasi dengan menggunakan perintah SQL yaitu *Create Table*. Tabel *PredictParameter* digunakan untuk menyimpan data nilai parameter prediksi dimana terdiri dari *id_predict*, *alpha_value* dan *beta_value*. Implementasi tabel *PredictParameter* ditunjukkan pada Tabel 5.23.

Tabel 5.23 Implementasi Tabel *PredictParameter*

Implementasi Tabel <i>PredictParameter</i>	
1	<code>CREATE TABLE [dbo].[PredictParameter] (</code>
2	<code>[id_predict] [numeric](18, 0) IDENTITY(1,1) NOT NULL,</code>
3	<code>[alpha_value] [float] NOT NULL,</code>
4	<code>[beta_value] [float] NOT NULL</code>
5	<code>)</code>

5.2.3.8 Implementasi Tabel *Product*

Pada implementasi tabel *Product* dilakukan *Data Definition Language* (DDL) dengan membuat struktur tabel dan mendefinisikan tipe data dari tabel *Product*. DDL diimplementasi dengan menggunakan perintah SQL yaitu *Create Table*. Tabel *Product* digunakan untuk menyimpan data produk dimana terdiri dari

id_product, *code_product*, *name_product*, *category_product* dan *price_product*. Implementasi tabel *Product* ditunjukkan pada Tabel 5.24.

Tabel 5.24 Implementasi Tabel *Product*

Implementasi Tabel <i>Product</i>	
1	<code>CREATE TABLE [dbo].[Product] (</code>
2	<code>[id_product] [numeric](18, 0) IDENTITY(1,1) NOT NULL,</code>
3	<code>[code_product] [varchar](50) NOT NULL,</code>
4	<code>[name_product] [varchar](50) NOT NULL,</code>
5	<code>[category_product] [varchar](50) NOT NULL,</code>
6	<code>[price_product] [numeric](18, 0) NOT NULL</code>
7	<code>)</code>

5.2.3.9 Implementasi Tabel *Recommendation*

Pada implementasi tabel *Recommendation* dilakukan *Data Definition Language* (DDL) dengan membuat struktur tabel dan mendefinisikan tipe data dari tabel *Recommendation*. DDL diimplementasi dengan menggunakan perintah SQL yaitu *Create Table*. Tabel *Recommendation* digunakan untuk menyimpan data kombinasi produk dimana terdiri dari *id_recommendation*, *item_combination*, dan *total_value*. Implementasi tabel *Recommendation* ditunjukkan pada Tabel 5.25.

Tabel 5.25 Implementasi Tabel *Recommendation*

Implementasi Tabel <i>Recommendation</i>	
1	<code>CREATE TABLE [dbo].[Recommendation] (</code>
2	<code>[id_recommendation] [numeric](18, 0) IDENTITY(1,1) NOT NULL,</code>
3	<code>[item_combination] [varchar](max) NULL,</code>
4	<code>[total_value] [numeric](18, 0) NULL</code>
5	<code>)</code>

5.2.3.10 Implementasi Tabel *TopIncome*

Pada implementasi tabel *ItemSold* dilakukan *Data Definition Language* (DDL) dengan membuat struktur tabel dan mendefinisikan tipe data dari tabel *User*. DDL diimplementasi dengan menggunakan perintah SQL yaitu *Create Table*. Tabel *TopIncome* digunakan untuk menyimpan data 10 pendapatan tertinggi pada suatu hari dimana terdiri dari *id_income*, *ranking_income*, *total_income*, dan *date_income*. Implementasi tabel *TopIncome* ditunjukkan pada Tabel 5.26.

Tabel 5.26 Implementasi Tabel *TopIncome*

Implementasi Tabel <i>TopIncome</i>	
1	<code>CREATE TABLE [dbo].[TopIncome] (</code>
2	<code>[id_income] [numeric](18, 0) IDENTITY(1,1) NOT NULL,</code>
3	<code>[ranking_income] [numeric](18, 0) NULL,</code>
4	<code>[total_income] [numeric](18, 0) NULL,</code>
5	<code>[date_income] [varchar](50) NULL,</code>
6	<code>)</code>

5.2.3.11 Implementasi Tabel *TransactionHeader*

Pada implementasi tabel *TransactionHeader* dilakukan *Data Definition Language* (DDL) dengan membuat struktur tabel dan mendefinisikan tipe data dari tabel *TransactionHeader*. DDL diimplementasi dengan menggunakan perintah SQL

yaitu *Create Table*. Tabel *TransactionHeader* digunakan untuk menyimpan data transaksi dimana terdiri dari *code_transaction*, *date_time*, *discount_transaction*, *payment_method* dan *payment_total*. Implementasi tabel *TransactionHeader* ditunjukkan pada Tabel 5.27.

Tabel 5.27 Implementasi Tabel *TransactionHeader*

Implementasi Tabel <i>TransactionHeader</i>	
1	<code>CREATE TABLE [dbo].[TransactionHeader](</code>
2	<code>[code_transaction] [numeric](18, 0) NOT NULL,</code>
3	<code>[date_time] [date] NOT NULL,</code>
4	<code>[discount_transaction] [numeric](18, 0) NULL,</code>
5	<code>[payment_method] [varchar](20) NOT NULL,</code>
6	<code>[payment_total] [numeric](18, 0) NOT NULL,</code>
7	<code>)</code>

5.2.3.12 Implementasi Tabel *TrendsParameter*

Pada implementasi tabel *TrendsParameter* dilakukan *Data Definition Language* (DDL) dengan membuat struktur tabel dan mendefinisikan tipe data dari tabel *TrendsParameter*. DDL diimplementasi dengan menggunakan perintah SQL yaitu *Create Table*. Tabel *TrendsParameter* digunakan untuk menyimpan data nilai parameter produk potensial dimana terdiri dari *id_trends* dan *minsupport_value*. Implementasi tabel *TrendsParameter* ditunjukkan pada Tabel 5.28.

Tabel 5.28 Implementasi Tabel *TrendsParameter*

Implementasi Tabel <i>TrendsParameter</i>	
1	<code>CREATE TABLE [dbo].[TrendsParameter](</code>
2	<code>[id_trends] [numeric](18, 0) IDENTITY(1,1) NOT NULL,</code>
3	<code>[minsupport_value] [numeric](18, 0) NULL,</code>
4	<code>)</code>

5.2.3.13 Implementasi Tabel *User*

Pada implementasi tabel *User* dilakukan *Data Definition Language* (DDL) dengan membuat struktur tabel dan mendefinisikan tipe data dari tabel *User*. DDL diimplementasi dengan menggunakan perintah SQL yaitu *Create Table*. Tabel *User* digunakan untuk menyimpan data akun dimana terdiri dari *username*, *password*, dan *role*. Implementasi tabel *User* ditunjukkan pada Tabel 5.29.

Tabel 5.29 Implementasi Tabel *User*

Implementasi Tabel <i>User</i>	
1	<code>CREATE TABLE [dbo].[User](</code>
2	<code>[username] [varchar](30) NOT NULL,</code>
3	<code>[password] [varchar](80) NOT NULL,</code>
4	<code>[role] [varchar](20) NOT NULL</code>

5.2.4 Implementasi Antarmuka

Hasil implementasi antarmuka ialah antarmuka yang telah diimplementasikan pada sistem. Implementasi ini di dasarkan pada perancangan antarmuka yang telah dipaparkan sebelumnya. Pada implementasi antarmuka akan dipaparkan 3 antarmuka. Implementasi antarmuka tersebut yaitu halaman

transaksi, prediksi pendapatan, dan kombinasi produk. Penjelasan lebih rinci untuk implementasi antar muka di paparkan pada sub bab 5.2.4.1 hingga sub bab 5.2.4.3.

5.2.4.1 Implementasi Antarmuka Halaman Transaksi

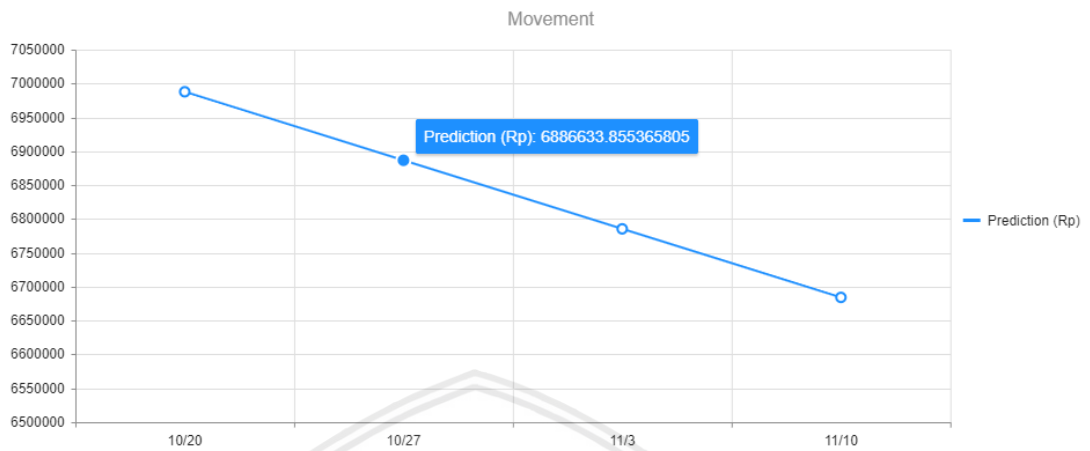
Dalam Gambar 5.13 merupakan gambar implementasi antarmuka halaman transaksi. Implementasi antarmuka ini berdasarkan pada perancangan yang telah dilakukan pada tahapan perancangan antarmuka sebelumnya. Secara umum, pada implementasi antarmuka ini terdapat tombol “Add To Cart” untuk menambahkan *item*, tombol “Remove” untuk menghapus *item* pada transaksi, tombol “Reset” untuk memulai ulang proses transaksi, tombol “Check Out” untuk melakukan finalisasi transaksi, serta beberapa tombol pada *SideBar* untuk mereferensikan ke halaman lain. Selain itu, terdapat beberapa *textfield* untuk mengisi data-data yang dibutuhkan seperti *Product Code*, *Product Quantity*, *Product Name*, *Price*, dan *Total Price*, *Transaction*, *Discount*, *Paid*, dan *Change*. Lebih lanjut, terdapat *label* yang mendefinisikan keterangan pada masing masing *textfield* maupun *dropdown* untuk *Payment Method*.

Gambar 5.12 Implementasi Antarmuka Halaman Transaksi

5.2.4.2 Implementasi Antarmuka Prediksi Pendapatan

Pada Gambar 5.14 terdapat gambar implementasi antarmuka prediksi pendapatan. Implementasi antarmuka ini berdasarkan pada perancangan yang telah dilakukan pada tahapan perancangan antarmuka sebelumnya. Secara umum, pada implementasi antarmuka prediksi pendapatan hanya di isi oleh *header* yang menerangkan bahwa bagian terkait ditujukan untuk menampilkan prediksi pendapatan per pekan. Selain itu, terdapat visualisasi pergerakan data prediksi dalam representasi *line chart*. *Area line chart* secara vertikal ditujukan untuk menampilkan nominal dan secara horizontal untuk menampilkan tanggal. Lebih lanjut, terdapat *label* yang menunjukkan bahwa warna biru digunakan untuk keterangan prediksi dalam kurs rupiah.

Weekly Revenue Prediction



Gambar 5.13 Implementasi Antarmuka Prediksi Pendapatan

5.2.4.3 Implementasi Antarmuka Kombinasi Produk

Dalam Gambar 5.15 merupakan gambar implementasi antarmuka kombinasi produk. Secara umum, implementasi kombinasi produk terdiri atas *header* untuk menampilkan deskripsi mengenai bagian kombinasi produk. Kemudian terdapat *search area* untuk melakukan pencarian kombinasi produk. Selanjutnya terdapat visualisasi *table grid*. Bagian ini mendeskripsikan daftar kombinasi produk dan jumlah kombinasi tersebut bertemu pada satu transaksi.



Gambar 5.14 Implementasi Antarmuka Kombinasi Produk



BAB 6 PENGUJIAN

6.1 Pengujian Unit

Pada pengujian unit akan dilakukan pengujian pada setiap unit atau komponen dari perangkat lunak yang dikembangkan. Pengujian unit ini menggunakan metode *Whitebox testing* dengan teknik pengujian *basis path testing*. Pada pengujian unit ini akan menguji tiga *method*. Tiga *method* tersebut adalah *method InsertTable Cart*, *Prediction_Data*, dan *Recommendation_Data*.

6.1.1 Pengujian Unit Method InsertTableCart

1. Pseudocodes

Pengujian unit pada *method InsertTableCart* dari class *SupervisorController* akan dijelaskan algoritme yang ditunjukkan pada Tabel 6.1.

Tabel 6.1 Pseudocode Algoritme InsertTableCart

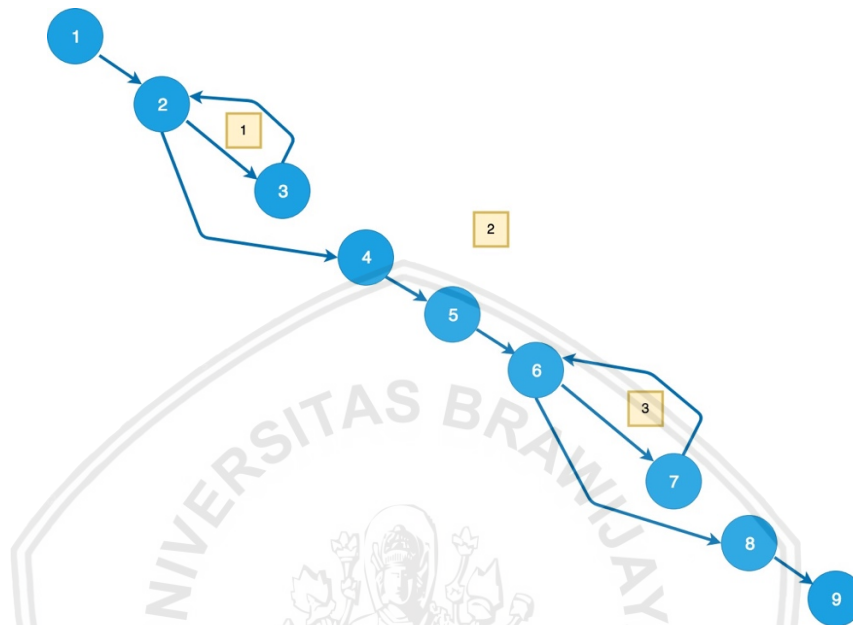
Pseudocode algoritme <i>method InsertTableCart</i>	
1	var db -> Initiate EddsWaffleEntites database
2	var table1 -> Initiate OrderDetails table from database
3	var getlatest -> Initiate the list of Transaction Header table
4	from database
5	var ordering -> take 1 row from code_transaction column from
5	getlatest, ordered by descending
6	Int64 codetransaction -> 0
7	
	Foreach var item in ordering 2
8	Set int codetransaction = (Int64) item 3
9	End foreach 4
10	
	Int64 modifycode = codetransaction + 1
11	String modify = make modifycode to string
12	today -> Initiate current date
13	var current -> formatting today in yy-mm-dd and to string
14	var modified -> initiate StringBuilder for modify
15	modified remove 8 characthers started from characther number 1
16	modified insert current started from characther number 1
17	modify -> modified to string
18	Int64 transactioncode -> Convert modify to Int64
19	Foreach TableCartModel I in itemlist 6
20	code_transaction from table1 = transactioncode
21	code_product from table1 = code_product from i
22	name_product from table1 = name_product from i
22	quantity_product from table1 = quantity_product from i
23	price_product from table1 = totalprice_product from i
24	db in OrderDetails table from database add table1
25	db make a save changes
26	End foreach 8
27	Return "Transaction Successfully Added!" in Json 9



2. Basis Path Testing

2.1. Flow Graph

Dari algoritme yang dipaparkan akan menjadi dasar pembuatan *flowgraph* untuk method *InsertTableCart* yang ditunjukkan dalam Gambar 6.1.



Gambar 6.1 Flowgraph method *InsertTableCart*

2.2. Cyclomatic Complexity

Dari Gambar 6.1 akan diketahui jumlah jalur independent melalui kalkulasi *cyclomatic complexity*. Berikut merupakan hasil kalkulasi *cyclomatic complexity* dari *flowgraph method InsertTableCart*.

- $V(G) = \text{jumlah region} = 3$
- $V(G) = \text{jumlah edge} - \text{jumlah node} + 2 = 10 - 9 + 2 = 3$
- $V(G) = \text{jumlah predicate node} + 1 = 2 + 1 = 3$

2.3. Independent Path

- Jalur 1: 1-2-4-5-6-8-9
- Jalur 2: 1-2-3-2-4-5-6-8-9
- Jalur 3: 1-2-3-2-4-5-6-7-6-8-9

Jalur *independent* yang didapat akan dijadikan dasar pembuatan kasus uji. Kasus uji dan hasil dari pengujian unit *method InsertTableCart* ditunjukkan pada Tabel 6.2.

Tabel 6.2 Hasil pengujian unit *method InsertTableCart*

Jalur	Prosedur uji	Hasil pengujian yang diharapkan	Hasil pengujian	Status
1	<i>Class Driver</i> memanggil <i>method InsertTableCart()</i> dengan data pada <i>TransactionHeader</i> adalah <i>null</i> dan <i>itemlist</i> adalah <i>null</i>	Menjalankan <i>method InsertTableCart()</i> dan menampilkan pesan pada <i>console</i> "Looping 1 Selesai" "Looping 2 Selesai"	Menjalankan <i>method InsertTableCart()</i> dan menampilkan pesan pada <i>console</i> "Looping 1 Selesai" "Looping 2 Selesai"	Valid
2	<i>Class Driver</i> memanggil <i>method InsertTableCart()</i> dengan data pada <i>TransactionHeader</i> adalah 1 dan <i>itemlist</i> adalah <i>null</i>	Menjalankan <i>method InsertTableCart()</i> dan menampilkan pesan pada <i>console</i> "Kode Transaksi Diambil" "Looping 1 Selesai" "Looping 2 Selesai"	Menjalankan <i>method InsertTableCart()</i> dan menampilkan pesan pada <i>console</i> "Kode Transaksi Diambil" "Looping 1 Selesai" "Looping 2 Selesai"	Valid
3	<i>Class Driver</i> memanggil <i>method InsertTableCart()</i> dengan data pada <i>TransactionHeader</i> adalah 1 dan <i>itemlist</i> adalah <i>code_product</i> = "Waffle - 001" <i>name_product</i> = "Cookies and Cream - Waffle" <i>quantity_product</i> = 1 <i>price_product</i> = 26000 <i>totalprice_product</i> = 26000	Menjalankan <i>method InsertTableCart()</i> dan menampilkan pesan pada <i>console</i> "Kode Transaksi Diambil" "Looping 1 Selesai" "Transaksi ditambahkan" "Looping 2 Selesai"	Menjalankan <i>method InsertTableCart()</i> dan menampilkan pesan pada <i>console</i> "Kode Transaksi Diambil" "Looping 1 Selesai" "Transaksi ditambahkan" "Looping 2 Selesai"	Valid

6.1.2 Pengujian Unit Method *Prediction_Data*

1. Pseudocodes

Pengujian unit pada *method Prediction_Data* pada *class SupervisorController* akan dijelaskan algoritme yang ditunjukkan pada Tabel 6.3 dan Tabel 6.4.

Tabel 6.3 Pseudocode Algoritme *method Prediction_Data*

Pseudocode algoritme <i>method Prediction_Data</i>	
1	db -> Initiate EddsWaffleEntites database
2	Var Forecasting -> Get Prediction database in array
3	Estimate -> Initiate the list of Prediction database
4	
5	For int i from 0 until number of Prediction database row count
6	Estimate add Forecasting in each element

} 1

2

3



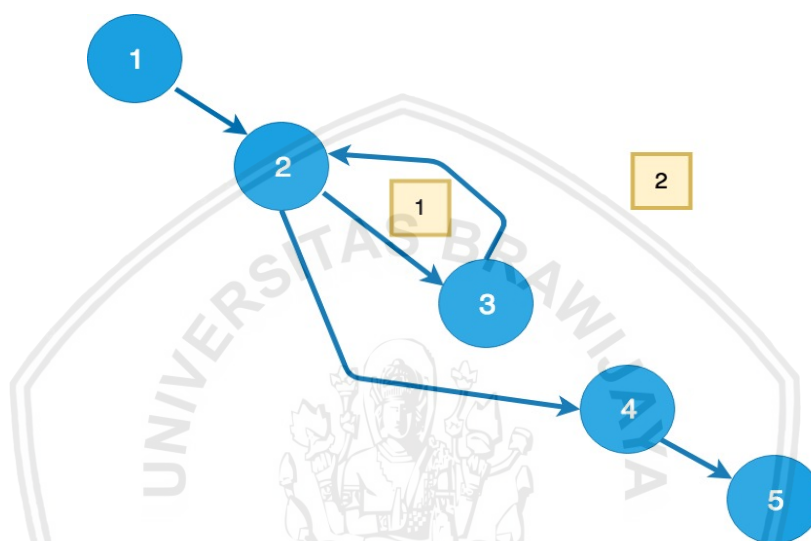
Tabel 6.4 Pseudocode Algoritme method Prediction_Data (Lanjutan)

Pseudocode algoritme method Prediction_Data	
7	End for <u>4</u>
8	Return estimate in Json <u>5</u>

2. Basis Path Testing

2.1. Flow Graph

Dari algoritme yang dipaparkan akan menjadi dasar pembuatan *flowgraph* untuk method *Prediction_Data* yang ditunjukkan dalam Gambar 6.2.



Gambar 6.2 Flowgraph method Prediction_Data

2.2. Cyclomatic Complexity

Dari Gambar 6.2 akan diketahui jumlah jalur independent melalui kalkulasi *cyclomatic complexity*. Berikut merupakan hasil kalkulasi *cyclomatic complexity* dari *flowgraph method Prediction_Data*.

- $V(G) = \text{jumlah region} = 2$
- $V(G) = \text{jumlah edge} - \text{jumlah node} + 2 = 2 - 1 + 1 = 2$
- $V(G) = \text{jumlah predicate node} + 1 = 1 + 1 = 2$

2.3. Independent Path

- Jalur 1: 1-2-4-5
- Jalur 2: 1-2-3-4-2-4-5

Jalur *independent* yang didapat akan dijadikan dasar pembuatan kasus uji. Hasil dari pengujian unit method *Prediction_Data* ditunjukkan pada Tabel 6.5.

Tabel 6.5 Hasil pengujian unit *method Prediction_Data*

Jalur	Prosedur uji	Hasil pengujian yang diharapkan	Hasil pengujian	Status
1	<i>Class Driver</i> memanggil <i>method Prediction_Data ()</i> dengan jumlah data pada <i>table Prediction</i> adalah <i>null</i>	Menjalankan <i>method Prediction_Data ()</i> dan menampilkan pesan pada <i>console</i> “Loop Selesai”	Menjalankan <i>method Prediction_Data ()</i> dan menampilkan pesan pada <i>console</i> “Loop Selesai”	Valid
2	<i>Class Driver</i> memanggil <i>method Prediction_Data ()</i> dengan jumlah data pada <i>table Prediction</i> adalah 4	Menjalankan <i>method Prediction_Data ()</i> dan menampilkan pesan pada <i>console</i> “Prediksi 0 di dapatkan” “Prediksi 1 di dapatkan” “Prediksi 2 di dapatkan” “Prediksi 3 di dapatkan” “Loop Selesai”	Menjalankan <i>method Prediction_Data ()</i> dan menampilkan pesan pada <i>console</i> “Prediksi 0 di dapatkan” “Prediksi 1 di dapatkan” “Prediksi 2 di dapatkan” “Prediksi 3 di dapatkan” “Loop Selesai”	Valid

6.1.3 Pengujian Unit *Method Recommendation_Data*

1. *Pseudocodes*

Pengujian unit pada *method Recommendation_Data* dari *class SupervisorController* akan dijelaskan dalam algoritme yang ditunjukkan pada Pseudocode Tabel 6.6.

Tabel 6.6 Pseudocode Algoritme *method Recommendation_Data*

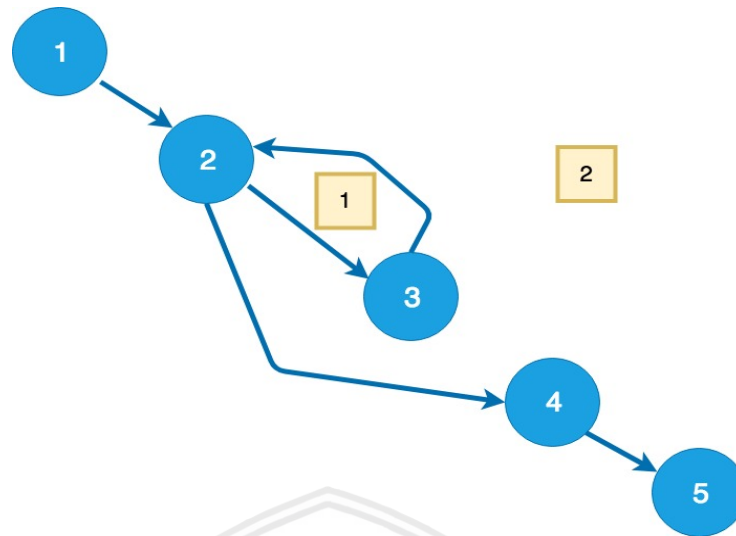
Pseudocode algoritme <i>method Recommendation_Data</i>	
1	db -> Initiate EddsWaffleEntites database
2	var showrecommendation -> Get Recommendation database in array
3	show -> Initiate the list of Recommendation database
4	For int i from 0 until number of Recommendation database
5	row count 2
6	show add showrecommendation in each element 3
	End for 4
7	Return recommendation in Json 5

2. *Basis Path Testing*

2.1. *Flow Graph*

Dari algoritme yang dipaparkan akan menjadi dasar pembuatan *flowgraph* untuk *method Recommendation_Data* yang ditunjukkan dalam Gambar 6.3.





Gambar 6.3 Flowgraph method Recommendation_Data

2.2. Cyclomatic Complexity

Dari Gambar 6.3 akan diketahui jumlah jalur independent melalui kalkulasi *cyclomatic complexity*. Berikut merupakan hasil kalkulasi *cyclomatic complexity* dari *flowgraph method Recommendation_Data*.

- $V(G) = \text{jumlah region} = 2$
- $V(G) = \text{jumlah edge} - \text{jumlah node} + 2 = 5 - 5 + 2 = 2$
- $V(G) = \text{jumlah predicate node} + 1 = 1 + 1 = 2$

2.3. Independent Path

- Jalur 1: 1-2-4-5
- Jalur 2: 1-2-3-2-4-5

Jalur *independent* yang didapat akan dijadikan dasar pembuatan kasus uji. Hasil dari pengujian unit *method Recommendation_Data* ditunjukkan pada Tabel 6.7 dan Tabel 6.8.

Tabel 6.7 Hasil pengujian unit method Recommendation_Data

Jalur	Prosedur uji	Hasil pengujian yang diharapkan	Hasil pengujian	Status
1	Class Driver memanggil <i>method Recommendation_Data()</i> jumlah data pada <i>table Recommendation</i> adalah <i>null</i>	Menjalankan <i>method Recommendation_Data ()</i> dan menampilkan pesan pada <i>console "Looping Selesai"</i>	Menjalankan <i>method Recommendation_Data ()</i> dan menampilkan pesan pada <i>console "Looping Selesai"</i>	Valid

Tabel 6.8 Hasil pengujian unit *method Recommendation_Data* (Lanjutan)

Jalur	Prosedur uji	Hasil pengujian yang diharapkan	Hasil pengujian	Status
2	<i>Class Driver</i> memanggil <i>method Recommendation_Data()</i> jumlah data pada <i>table Recommendation</i> adalah 666	Menjalankan <i>method Recommendation_Data ()</i> dan menampilkan pesan pada <i>console</i> "Kombinasi di dapatkan" sebanyak 666 kali "Looping Selesai"	Menjalankan <i>method Recommendation_Data ()</i> dan menampilkan pesan pada <i>console</i> "Kombinasi di dapatkan" sebanyak 666 kali "Looping Selesai"	Valid

6.2 Pengujian Integrasi

Pengujian Integrasi pada penelitian ini bertujuan untuk memastikan bahwa interaksi antar *class* berjalan dengan baik. Pada pengujian Integrasi menggunakan *basis path testing*. Dalam pelaksanaan pengujian integrasi ini diperlukan sebuah *class test* yang digunakan untuk menjalankan *method class* yang ingin diuji. Pengujian integrasi penelitian ini dilakukan untuk menguji interaksi *method Read ()* yang ada pada *class SalesMonitoring* dengan *method Prediction_Data ()* yang ada pada *class SupervisorController*. Dalam interaksi ini, *method Prediction_Data ()* pada *class SupervisorController* akan memanggil *method Read()* pada *SetProduct*.

1. Pseudocodes

Pengujian integrasi pada *method Prediction_Data* dari *class SupervisorController* akan dijelaskan algoritme yang ditunjukkan pada Tabel 6.9.

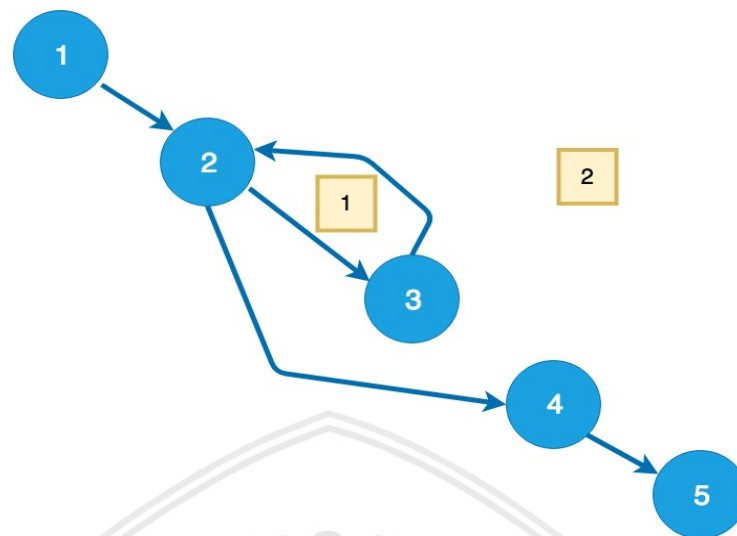
Tabel 6.9 Pseudocode Algoritme *method Prediction_Data*

Pseudocode algoritme <i>method Prediction_Data</i>	
1	db -> Initiate EddsWaffleEntites database
2	Var Forecasting -> Get Prediction database in array
3	Estimate -> Initiate the list of Prediction database
4	For int i from 0 until number of Prediction database row count <u>2</u>
5	Estimate add Forecasting in each element <u>3</u>
6	End for <u>4</u>
7	Return estimate in Json <u>5</u>

2. Basis Path Testing

2.1. Flow Graph

Dari algoritme yang dipaparkan akan menjadi dasar pembuatan *flowgraph* untuk *method AddProduct* yang ditunjukkan dalam Gambar 6.4.



Gambar 6.4 Flowgraph *method Prediction_Data*

2.2. Cyclomatic Complexity

Dari Gambar 6.4 akan diketahui jumlah jalur independent melalui kalkulasi *cyclomatic complexity*. Berikut merupakan hasil kalkulasi *cyclomatic complexity* dari *flowgraph method Prediction_Data*.

- $V(G) = \text{jumlah region} = 2$
- $V(G) = \text{jumlah edge} - \text{jumlah node} + 2 = 5 - 5 + 2 = 2$
- $V(G) = \text{jumlah predicate node} + 1 = 1 + 1 = 2$

2.3. Independent Path

- Jalur 1: 1-2-4-5
- Jalur 2: 1-2-3-2-4-5

Jalur *independent* yang didapat akan dijadikan dasar pembuatan kasus uji. Kasus uji dan hasil dari pengujian integrasi *method Prediction_Data* dengan *method Read* ditunjukkan pada Tabel 6.10. Pada pengujian integrasi ini, di dapatkan 2 jalur *independent* dikarenakan terdapatnya seleksi kondisi *loop*. Jalur *independent* tidak melewati seleksi *loop* apabila tidak terdapat data yang terdapat pada *table prediction*, sehingga perulangan tidak dilakukan karena memiliki batas atau *limit*. Sedangkan jalur *independent* melewati *loop* apabila terdapat data di dalam *table prediction*, hal ini dikarenakan batas atau *limit* telah di definisikan sehingga proses mendapatkan data prediksi dapat dilakukan.

Tabel 6.10 Hasil pengujian integrasi

Jalur	Prosedur uji	Hasil pengujian yang diharapkan	Hasil pengujian	Status
1	<i>Class Driver</i> memanggil <i>method Prediction_Data ()</i> yang memanggil <i>method Read()</i> dengan jumlah data pada <i>table Prediction</i> adalah <i>null</i>	Menjalankan <i>method Prediction_Data ()</i> yang memanggil <i>method Read()</i> dan menampilkan pesan pada <i>console</i> "Loop Selesai"	Menjalankan <i>method Prediction_Data ()</i> yang memanggil <i>method Read()</i> dan menampilkan pesan pada <i>console</i> "Loop Selesai"	Valid
2	<i>Class Driver</i> memanggil <i>method Prediction_Data ()</i> yang memanggil <i>method Read()</i> dengan jumlah data pada <i>table Prediction</i> adalah 4	Menjalankan <i>method Prediction_Data ()</i> yang memanggil <i>method Read()</i> dan menampilkan pesan pada <i>console</i> "Prediksi 0 di dapatkan" "Prediksi 1 di dapatkan" "Prediksi 2 di dapatkan" "Prediksi 3 di dapatkan" "Loop Selesai"	Menjalankan <i>method Prediction_Data ()</i> yang memanggil <i>method Read()</i> dan menampilkan pesan pada <i>console</i> "Prediksi 0 di dapatkan" "Prediksi 1 di dapatkan" "Prediksi 2 di dapatkan" "Prediksi 3 di dapatkan" "Loop Selesai"	Valid

6.3 Pengujian Validasi

Pada pengujian validasi akan dilakukan pengujian pada setiap kebutuhan yang telah didefinisikan sebelumnya. Pengujian ini digunakan untuk memastikan bahwa semua kebutuhan telah diterapkan sesuai skenario yang dibuat. Pengujian validasi akan menguji seluruh kebutuhan baik kebutuhan fungsional dan kebutuhan non-fungsional. Pengujian validasi menggunakan metode *Blackbox*. Pada penelitian ini akan menguji lima puluh kebutuhan fungsional beserta alur alternatif. Pengujian validasi pada penelitian ini dijelaskan pada sub bab 6.3.1 sampai 6.3.50.

6.3.1 Login

Pengujian validasi *login* akan menguji kebutuhan fungsional POSEW-F-01. Pada pengujian ini, dilakukan validasi terhadap sistem agar dapat mengidentifikasi segmen pengguna. Pada pengujian validasi jalur alternatif akan memastikan bahwa sistem tidak dapat mengidentifikasi segmen pengguna apabila *username* maupun *password* belum terdaftar pada basis data. Pengujian validasi *login* dipaparkan pada Tabel 6.11 dan jalur alternatif dipaparkan pada Tabel 6.12.

Tabel 6.11 Pengujian validasi Login

Nama kasus uji	<i>Login</i>
Objek uji	Kebutuhan fungsional POSEW-F-01
Tujuan pengujian	Memastikan bahwa sistem mampu mengidentifikasi segmentasi pengguna
Prosedur uji	<ol style="list-style-type: none"> 1. Mengakses halaman utama sistem yaitu halaman <i>login</i> 2. Mengisi <i>username</i> dan <i>password</i> 3. Klik tombol <i>Login</i>
Hasil pengujian yang diharapkan	Sistem menampilkan halaman utama <i>Supervisor, Cashier, atau Administrator.</i>
Hasil pengujian	Sistem menampilkan halaman utama <i>Supervisor, Cashier, atau Administrator.</i>
Status	Valid

Tabel 6.12 Pengujian validasi Login - Alternatif

Nama kasus uji	<i>Login - alternatif</i>
Objek uji	Kebutuhan fungsional POSEW-F-01
Tujuan pengujian	Memastikan bahwa sistem tidak memberikan akses apabila akun dari setiap segmen pengguna belum terdaftar
Prosedur uji	<ol style="list-style-type: none"> 1. Mengakses halaman utama sistem yaitu halaman <i>login</i> 2. Mengisi <i>username</i> dan <i>password</i> 3. Klik tombol <i>Login</i>
Hasil pengujian yang diharapkan	Sistem menampilkan peringatan " <i>Incorrect Username / Password – Please Try Again</i> ".
Hasil pengujian	Sistem menampilkan peringatan " <i>Incorrect Username / Password – Please Try Again</i> ".
Status	Valid

6.3.2 Logout

Pengujian validasi *logout* akan menguji kebutuhan fungsional POSEW-F-02. Pada pengujian ini, dilakukan validasi terhadap sistem agar dapat mengeluarkan *Cashier, Supervisor, dan Administrator* dari *role* nya pada sistem. Pengujian validasi *logout* dipaparkan pada Tabel 6.13 dan Tabel 6.14.

Tabel 6.13 Pengujian validasi Logout

Nama kasus uji	<i>Logout</i>
Objek uji	Kebutuhan fungsional POSEW-F-02
Tujuan pengujian	Memastikan bahwa sistem dapat mengeluarkan <i>Supervisor, Cashier, dan Administrator</i> dari <i>role</i> nya pada sistem.

Tabel 6.14 Pengujian validasi Logout (Lanjutan)

Prosedur uji	1. Klik tombol <i>Sign Out</i>
Hasil pengujian yang diharapkan	Sistem menampilkan halaman <i>login</i> .
Hasil pengujian	Sistem menampilkan halaman <i>login</i> .
Status	Valid

6.3.3 Tambah Transaksi

Pengujian validasi tambah transaksi akan menguji kebutuhan fungsional POSEW-F-03. Pada pengujian ini, dilakukan validasi terhadap sistem agar dapat menambahkan transaksi. Pada pengujian validasi jalur alternatif akan memastikan bahwa sistem tidak dapat menambahkan transaksi apabila terdapat kekosongan pada data yang di butuhkan, jumlah uang tunai yang lebih kecil di banding total harga, dan penambahan diskon. Pengujian validasi tambah transaksi dipaparkan pada Tabel 6.15 dan jalur alternatif dipaparkan pada Tabel 6.15 hingga Tabel 6.23.

Tabel 6.15 Pengujian validasi Tambah Transaksi

Nama kasus uji	Tambah transaksi
Objek uji	Kebutuhan fungsional POSEW-F-03
Tujuan pengujian	Memastikan bahwa sistem dapat menambahkan transaksi
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Transaction</i> 2. Mengisi <i>Product Code</i> dan <i>Product Quantity</i> 3. Klik tombol <i>Add To Cart</i> 4. Mengisi <i>Paid</i> serta memilih <i>Payment Method</i> 5. Klik tombol <i>Check Out</i>
Hasil pengujian yang diharapkan	Sistem menampilkan peringatan " <i>Transaction Successfully Added !</i> "
Hasil pengujian	Sistem menampilkan peringatan " <i>Transaction Successfully Added !</i> "
Status	Valid

Tabel 6.16 Pengujian validasi Tambah Transaksi – Alternatif 1

Nama kasus uji	Tambah transaksi
Objek uji	Kebutuhan fungsional POSEW-F-03
Tujuan pengujian	Memastikan bahwa sistem dapat menolak penambahan <i>item</i> pada transaksi bila kode produk tidak diisi

Tabel 6.17 Pengujian validasi Tambah Transaksi – Alternatif 1 (Lanjutan)

Prosedur uji	1. Klik menu <i>transaction</i> 2. Mengisi <i>Product Code</i> 3. Klik tombol <i>Add To Cart</i>
Hasil pengujian yang diharapkan	Sistem menampilkan peringatan " <i>Please fill the Product Code correctly</i> "
Hasil pengujian	Sistem menampilkan peringatan " <i>Please fill the Product Code correctly</i> "
Status	Valid

Tabel 6.18 Pengujian validasi Tambah Transaksi – Alternatif 2

Nama kasus uji	Tambah transaksi
Objek uji	Kebutuhan fungsional POSEW-F-03
Tujuan pengujian	Memastikan bahwa sistem dapat menolak penambahan <i>item</i> pada transaksi bila kuantitas produk tidak diisi
Prosedur uji	1. Klik menu <i>transaction</i> 2. Mengisi <i>Product Code</i> 3. Klik tombol <i>Add To Cart</i>
Hasil pengujian yang diharapkan	Sistem menampilkan peringatan " <i>Please fill the Product Quantity correctly</i> "
Hasil pengujian	Sistem menampilkan peringatan " <i>Please fill the Product Quantity correctly</i> "
Status	Valid

Tabel 6.19 Pengujian validasi Tambah Transaksi – Alternatif 3

Nama kasus uji	Tambah transaksi
Objek uji	Kebutuhan fungsional POSEW-F-03
Tujuan pengujian	Memastikan bahwa sistem dapat menolak penambahan <i>item</i> pada transaksi bila kode produk tidak sesuai dengan kode produk yang telah terdaftar.
Prosedur uji	1. Klik menu <i>transaction</i> 2. Mengisi <i>Product Code</i> dan <i>Product Quantity</i> 3. Klik tombol <i>Add To Cart</i>
Hasil pengujian yang diharapkan	Sistem menampilkan peringatan " <i>Item not found</i> "
Hasil pengujian	Sistem menampilkan peringatan " <i>Item not found</i> "
Status	Valid

Tabel 6.20 Pengujian validasi Tambah Transaksi – Alternatif 4

Nama kasus uji	Tambah transaksi
Objek uji	Kebutuhan fungsional POSEW-F-03
Tujuan pengujian	Memastikan bahwa sistem dapat menolak penambahan transaksi apabila uang yang dibayarkan lebih kecil dibanding total harga belanja
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>transaction</i> 2. Mengisi <i>Product Code</i> dan <i>Product Quantity</i> 3. Klik tombol <i>Add To Cart</i> 4. Mengisi <i>Discount</i> dan <i>Paid</i> serta memilih <i>Payment Method</i> 5. Klik tombol <i>Check Out</i>
Hasil pengujian yang diharapkan	Sistem menampilkan peringatan " <i>Insufficient Money Entered – Please Make Total Paid Is Bigger Than Sub Total</i> "
Hasil pengujian	Sistem menampilkan peringatan " <i>Insufficient Money Entered – Please Make Total Paid Is Bigger Than Sub Total</i> "
Status	Valid

Tabel 6.21 Pengujian validasi Tambah Transaksi – Alternatif 5

Nama kasus uji	Tambah transaksi
Objek uji	Kebutuhan fungsional POSEW-F-03
Tujuan pengujian	Memastikan bahwa sistem dapat menolak penambahan transaksi apabila tidak ada item yang ditambahkan
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>transaction</i> 2. Klik tombol <i>Add To Cart</i>
Hasil pengujian yang diharapkan	Sistem menampilkan peringatan " <i>No Item Cart Found – Please Do A Transaction Correctly</i> "
Hasil pengujian	Sistem menampilkan peringatan " <i>No Item Cart Found – Please Do A Transaction Correctly</i> "
Status	Valid

Tabel 6.22 Pengujian validasi Tambah Transaksi – Alternatif 6

Nama kasus uji	Tambah transaksi
Objek uji	Kebutuhan fungsional POSEW-F-03
Tujuan pengujian	Memastikan bahwa sistem dapat melakukan kalkulasi total harga terkini apabila diskon di tambahkan
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Transaction</i> 2. Mengisi <i>Product Code</i> dan <i>Product Quantity</i> 3. Klik tombol <i>Add To Cart</i> 4. Mengisi <i>Discount</i>

Tabel 6.23 Pengujian validasi Tambah Transaksi – Alternatif 6 (Lanjutan)

Hasil pengujian yang diharapkan	Sistem melakukan kalkulasi perkalian antara diskon dan total harga serta menampilkan <i>Sub Total</i> terkini (<i>textfield</i>)
Hasil pengujian	Sistem melakukan kalkulasi perkalian antara diskon dan total harga serta menampilkan <i>Sub Total</i> terkini (<i>textfield</i>)
Status	Valid

6.3.4 Hapus *Item* Transaksi

Pengujian validasi hapus transaksi *akan* menguji kebutuhan fungsional POSEW-F-04. Pada pengujian ini, dilakukan penghapusan pada *item* yang telah di tambahkan. Dengan menghapus *item*, maka sistem akan melakukan kalkulasi pengurangan pada *Sub Total*. Selain itu, uang kembalian atau *Change* akan di kalkulasi ulang. Pengujian validasi hapus *item* transaksi dipaparkan pada Tabel 6.24.

Tabel 6.24 Pengujian validasi Hapus *Item* Transaksi

Nama kasus uji	Hapus <i>item</i> transaksi
Objek uji	Kebutuhan fungsional POSEW-F-04
Tujuan pengujian	Memastikan bahwa sistem dapat menghapus <i>item</i> yang telah ditambahkan pada transaksi
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Transaction</i> 2. Mengisi <i>Product Code</i> dan <i>Product Quantity</i> 3. Klik tombol <i>Add To Cart</i> 4. Klik tombol <i>Remove</i> dalam tabel pada <i>item</i> yang telah ditambahkan
Hasil pengujian yang diharapkan	Sistem menghapus <i>item</i> yang telah dipilih aktor dan melakukan kalkulasi pengurangan sejumlah total harga produk pada <i>Sub Total</i> (<i>textfield</i>) dan kalkulasi ulang pada <i>Change</i> (<i>textfield</i>).
Hasil pengujian	Sistem menghapus <i>item</i> yang telah dipilih aktor dan melakukan kalkulasi pengurangan sejumlah total harga produk pada <i>Sub Total</i> (<i>textfield</i>) dan kalkulasi ulang pada <i>Change</i> (<i>textfield</i>).
Status	Valid

6.3.5 Reset Transaksi

Pengujian validasi hapus transaksi akan menguji kebutuhan fungsional POSEW-F-05. Pada pengujian ini, dilakukan proses ulang transaksi. Dengan melakukan *reset* pada transaksi, maka sistem akan mengosongkan seluruh *field*. Selain itu, tabel yang terdapat pada halaman transaksi juga di kosongkan. Pengujian validasi *reset* transaksi dipaparkan pada Tabel 6.25.

Tabel 6.25 Pengujian validasi Reset Transaksi

Nama kasus uji	Reset transaksi
Objek uji	Kebutuhan fungsional POSEW-F-05
Tujuan pengujian	Memastikan bahwa sistem dapat mengulang proses penambahan transaksi
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Transaction</i> 2. Mengisi <i>Product Code</i> dan <i>Product Quantity</i> 3. Klik tombol <i>Add To Cart</i> 4. Klik tombol <i>Reset</i>
Hasil pengujian yang diharapkan	Sistem mengosongkan seluruh <i>field</i> dan tabel yang terdapat pada halaman transaksi.
Hasil pengujian	Sistem mengosongkan seluruh <i>field</i> dan tabel yang terdapat pada halaman transaksi.
Status	Valid

6.3.6 Lihat Referensi Produk

Pengujian validasi lihat referensi produk akan menguji kebutuhan fungsional POSEW-F-06. Pada pengujian ini, dilakukan proses menampilkan referensi produk. Dengan adanya referensi produk, memungkinkan aktor untuk menemukan produk yang telah terdaftar. Pengujian validasi lihat referensi produk dipaparkan pada Tabel 6.26.

Tabel 6.26 Pengujian validasi Lihat Referensi Produk

Nama kasus uji	Lihat referensi produk
Objek uji	Kebutuhan fungsional POSEW-F-06
Tujuan pengujian	Memastikan bahwa sistem dapat menampilkan referensi produk
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>References</i>
Hasil pengujian yang diharapkan	Sistem menampilkan daftar produk yang telah terdaftar dengan informasi yang disajikan adalah kode produk dan nama produk.
Hasil pengujian	Sistem menampilkan daftar produk yang telah terdaftar dengan informasi yang disajikan adalah kode produk dan nama produk.
Status	Valid

6.3.7 Cari Referensi Produk

Pengujian validasi lihat referensi produk akan menguji kebutuhan fungsional POSEW-F-07. Pada pengujian ini, dilakukan proses melakukan pencarian pada referensi produk. Dengan adanya pencarian pada referensi produk,

memungkinkan aktor untuk mencari produk yang telah terdaftar. Pengujian validasi cari referensi produk dipaparkan pada Tabel 6.27.

Tabel 6.27 Pengujian validasi Cari Referensi Produk

Nama kasus uji	Cari referensi produk
Objek uji	Kebutuhan fungsional POSEW-F-07
Tujuan pengujian	Memastikan bahwa sistem menampilkan data referensi produk sesuai dengan masukan pada pencarian
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>References</i> 2. Mengisi <i>search area</i>
Hasil pengujian yang diharapkan	Sistem menampilkan hasil pencarian berdasarkan masukan data dari aktor.
Hasil pengujian	Sistem menampilkan hasil pencarian berdasarkan masukan data dari aktor.
Status	Valid

6.3.8 Lihat Total Transaksi Per Pekan

Pengujian validasi lihat total transaksi per pekan akan menguji kebutuhan fungsional POSEW-F-08. Pada pengujian ini, dilakukan proses menampilkan total transaksi dalam skala waktu setiap pekan. Melalui proses ini, memungkinkan aktor untuk melihat perkembangan total transaksi dalam visualisasi *line chart*. Pengujian validasi lihat total transaksi per pekan dipaparkan pada Tabel 6.28.

Tabel 6.28 Pengujian validasi Lihat Total Transaksi Per Pekan

Nama kasus uji	Lihat total transaksi per pekan
Objek uji	Kebutuhan fungsional POSEW-F-08
Tujuan pengujian	Memastikan bahwa sistem menampilkan jumlah total transaksi per pekan.
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Sales Monitoring</i> 2. Klik <i>Weeks</i> 3. Klik <i>Number of Transaction</i>
Hasil pengujian yang diharapkan	Sistem menampilkan jumlah total transaksi di setiap pekan dalam visualisasi plot pada grafik <i>line chart</i> .
Hasil pengujian	Sistem menampilkan jumlah total transaksi di setiap pekan dalam visualisasi plot pada grafik <i>line chart</i> .
Status	Valid

6.3.9 Lihat Total Transaksi Per Bulan

Pengujian validasi lihat total transaksi per bulan akan menguji kebutuhan fungsional POSEW-F-09. Pada pengujian ini, dilakukan proses menampilkan total transaksi dalam skala waktu setiap bulan. Melalui proses ini, memungkinkan aktor untuk melihat perkembangan total transaksi dalam visualisasi *line chart*. Pengujian validasi lihat total transaksi per bulan dipaparkan pada Tabel 6.29.

Tabel 6.29 Pengujian validasi Lihat Total Transaksi Per Bulan

Nama kasus uji	Lihat total transaksi per bulan
Objek uji	Kebutuhan fungsional POSEW-F-09
Tujuan pengujian	Memastikan bahwa sistem menampilkan jumlah total transaksi per bulan.
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Sales Monitoring</i> 2. Klik <i>Months</i> 3. Klik <i>Number of Transaction</i>
Hasil pengujian yang diharapkan	Sistem menampilkan jumlah total transaksi di setiap bulan dalam visualisasi plot pada grafik <i>line chart</i> .
Hasil pengujian	Sistem menampilkan jumlah total transaksi di setiap bulan dalam visualisasi plot pada grafik <i>line chart</i> .
Status	Valid

6.3.10 Lihat Total Pendapatan Transaksi Per Pekan

Pengujian validasi lihat total pendapatan transaksi per pekan akan menguji kebutuhan fungsional POSEW-F-10. Pada pengujian ini, dilakukan proses menampilkan total pendapatan transaksi dalam skala waktu setiap pekan. Melalui proses ini, memungkinkan aktor untuk melihat perkembangan total pendapatan transaksi dalam visualisasi *line chart*. Pengujian validasi lihat total pendapatan transaksi per pekan dipaparkan pada Tabel 6.30 dan Tabel 6.31.

Tabel 6.30 Pengujian validasi Lihat Total Pendapatan Transaksi Per Pekan

Nama kasus uji	Lihat total pendapatan transaksi per pekan
Objek uji	Kebutuhan fungsional POSEW-F-10
Tujuan pengujian	Memastikan bahwa sistem menampilkan total pendapatan transaksi per pekan.
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Sales Monitoring</i> 2. Klik label <i>Weeks</i> 3. Klik label <i>Total Revenue</i>
Hasil pengujian yang diharapkan	Sistem menampilkan jumlah total pendapatan transaksi di setiap pekan dalam visualisasi plot pada grafik <i>line chart</i> .

Tabel 6.31 Pengujian validasi Lihat Total Pendapatan Transaksi Per Pekan (Lanjutan)

Hasil pengujian	Sistem menampilkan jumlah total pendapatan transaksi di setiap pekan dalam visualisasi plot pada grafik <i>line chart</i> .
Status	Valid

6.3.11 Lihat Total Pendapatan Transaksi Per Bulan

Pengujian validasi lihat total pendapatan transaksi per bulan akan menguji kebutuhan fungsional POSEW-F-11. Pada pengujian ini, dilakukan proses menampilkan total pendapatan transaksi dalam skala waktu setiap bulan. Melalui proses ini, memungkinkan aktor untuk melihat perkembangan total pendapatan transaksi dalam visualisasi *line chart*. Pengujian validasi lihat total pendapatan transaksi per bulan dipaparkan pada Tabel 6.32.

Tabel 6.32 Pengujian validasi Lihat Total Pendapatan Transaksi Per Bulan

Nama kasus uji	Lihat total pendapatan transaksi per bulan
Objek uji	Kebutuhan fungsional POSEW-F-11
Tujuan pengujian	Memastikan bahwa sistem menampilkan total pendapatan transaksi per bulan.
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Sales Monitoring</i> 2. Klik label <i>Months</i> 3. Klik label <i>Total Revenue</i>
Hasil pengujian yang diharapkan	Sistem menampilkan jumlah pendapatan terendah dalam satu transaksi di setiap bulan dalam visualisasi plot pada grafik <i>line chart</i> .
Hasil pengujian	Sistem menampilkan jumlah pendapatan terendah dalam satu transaksi di setiap bulan dalam visualisasi plot pada grafik <i>line chart</i> .
Status	Valid

6.3.12 Lihat Pendapatan Transaksi Tertinggi Per Pekan

Pengujian validasi lihat pendapatan transaksi tertinggi per pekan akan menguji kebutuhan fungsional POSEW-F-12. Pada pengujian ini, dilakukan proses menampilkan pendapatan transaksi tertinggi dalam skala waktu setiap pekan. Melalui proses ini, memungkinkan aktor untuk melihat perkembangan pendapatan transaksi tertinggi dalam visualisasi *line chart*. Pengujian validasi lihat pendapatan transaksi tertinggi per pekan dipaparkan pada Tabel 6.33.

Tabel 6.33 Pengujian validasi Lihat Pendapatan Transaksi Tertinggi Per Pekan

Nama kasus uji	Lihat total pendapatan transaksi tertinggi per pekan
Objek uji	Kebutuhan fungsional POSEW-F-12
Tujuan pengujian	Memastikan bahwa sistem menampilkan jumlah pendapatan tertinggi dari transaksi yang telah terjadi dengan skala waktu setiap pekan.
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Sales Monitoring</i> 2. Klik label <i>Weeks</i> 3. Klik label <i>Maximum Transaction</i>
Hasil pengujian yang diharapkan	Sistem menampilkan rata-rata pendapatan transaksi di setiap pekan dalam visualisasi plot pada grafik <i>line chart</i> .
Hasil pengujian	Sistem menampilkan rata-rata pendapatan transaksi di setiap pekan dalam visualisasi plot pada grafik <i>line chart</i> .
Status	Valid

6.3.13 Lihat Pendapatan Transaksi Tertinggi Per Bulan

Pengujian validasi lihat pendapatan transaksi tertinggi per bulan akan menguji kebutuhan fungsional POSEW-F-13. Pada pengujian ini, dilakukan proses menampilkan pendapatan transaksi tertinggi dalam skala waktu setiap bulan. Melalui proses ini, memungkinkan aktor untuk melihat perkembangan pendapatan transaksi tertinggi dalam visualisasi *line chart*. Pengujian validasi lihat pendapatan transaksi tertinggi per bulan dipaparkan pada Tabel 6.34.

Tabel 6.34 Pengujian validasi Lihat Pendapatan Transaksi Tertinggi Per Bulan

Nama kasus uji	Lihat total pendapatan transaksi tertinggi per bulan
Objek uji	Kebutuhan fungsional POSEW-F-13
Tujuan pengujian	Memastikan bahwa sistem menampilkan jumlah pendapatan tertinggi dari transaksi yang telah terjadi dengan skala waktu setiap bulan.
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Sales Monitoring</i> 2. Klik label <i>Months</i> 3. Klik label <i>Maximum Transaction</i>
Hasil pengujian yang diharapkan	Sistem menampilkan jumlah pendapatan tertinggi dalam satu transaksi di setiap bulan dalam visualisasi plot pada grafik <i>line chart</i> .
Hasil pengujian	Sistem menampilkan jumlah pendapatan tertinggi dalam satu transaksi di setiap bulan dalam visualisasi plot pada grafik <i>line chart</i> .
Status	Valid



6.3.14 Lihat Pendapatan Transaksi Terendah Per Pekan

Pengujian validasi lihat pendapatan transaksi terendah per pekan akan menguji kebutuhan fungsional POSEW-F-14. Pada pengujian ini, dilakukan proses menampilkan pendapatan transaksi terendah dalam skala waktu setiap pekan. Melalui proses ini, memungkinkan aktor untuk melihat perkembangan pendapatan transaksi terendah dalam visualisasi *line chart*. Pengujian validasi lihat pendapatan transaksi terendah per pekan dipaparkan pada Tabel 6.35.

Tabel 6.35 Pengujian validasi Lihat Pendapatan Transaksi Terendah Per Pekan

Nama kasus uji	Lihat total pendapatan transaksi terendah per pekan
Objek uji	Kebutuhan fungsional POSEW-F-14
Tujuan pengujian	Memastikan bahwa sistem menampilkan jumlah pendapatan terendah dari transaksi yang telah terjadi dengan skala waktu setiap pekan.
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Sales Monitoring</i> 2. Klik label <i>Weeks</i> 3. Klik label <i>Minimum Transaction</i>
Hasil pengujian yang diharapkan	Sistem menampilkan jumlah pendapatan terendah dalam satu transaksi di setiap pekan dalam visualisasi plot pada grafik <i>line chart</i> .
Hasil pengujian	Sistem menampilkan jumlah pendapatan terendah dalam satu transaksi di setiap pekan dalam visualisasi plot pada grafik <i>line chart</i> .
Status	Valid

6.3.15 Lihat Pendapatan Transaksi Terendah Per Bulan

Pengujian validasi lihat pendapatan transaksi terendah per bulan akan menguji kebutuhan fungsional POSEW-F-15. Pada pengujian ini, dilakukan proses menampilkan pendapatan transaksi terendah dalam skala waktu setiap bulan. Melalui proses ini, memungkinkan aktor untuk melihat perkembangan pendapatan transaksi terendah dalam visualisasi *line chart*. Pengujian validasi lihat pendapatan transaksi terendah per bulan dipaparkan pada Tabel 6.36 dan Tabel 6.37.

Tabel 6.36 Pengujian validasi Lihat Pendapatan Transaksi Terendah Per Bulan

Nama kasus uji	Lihat total pendapatan transaksi terendah per bulan
Objek uji	Kebutuhan Fungsional POSEW-F-15
Tujuan pengujian	Memastikan bahwa sistem menampilkan jumlah pendapatan terendah dari transaksi yang telah terjadi dengan skala waktu setiap bulan.
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Sales Monitoring</i> 2. Klik label <i>Months</i> 3. Klik label <i>Minimum Transaction</i>

Tabel 6.37 Pengujian validasi Lihat Pendapatan Transaksi Terendah Per Bulan (Lanjutan)

Hasil pengujian yang diharapkan	Sistem menampilkan jumlah pendapatan terendah dalam satu transaksi di setiap bulan dalam visualisasi plot pada grafik <i>line chart</i> .
Hasil pengujian	Sistem menampilkan jumlah pendapatan terendah dalam satu transaksi di setiap bulan dalam visualisasi plot pada grafik <i>line chart</i> .
Status	Valid

6.3.16 Lihat Rata-Rata Pendapatan Transaksi Per Pekan

Pengujian validasi lihat rata-rata transaksi per pekan akan menguji kebutuhan fungsional POSEW-F-16. Pada pengujian ini, dilakukan proses menampilkan rata-rata pendapatan transaksi dalam skala waktu setiap pekan. Melalui proses ini, memungkinkan aktor untuk melihat perkembangan rata-rata pendapatan transaksi dalam visualisasi *line chart*. Pengujian validasi lihat rata-rata pendapatan transaksi per pekan dipaparkan pada Tabel 6.38.

Tabel 6.38 Pengujian validasi Lihat Rata-Rata Pendapatan Transaksi Per Pekan

Nama kasus uji	Lihat rata-rata pendapatan transaksi per pekan
Objek uji	Kebutuhan Fungsional POSEW-F-16
Tujuan pengujian	Memastikan bahwa sistem menampilkan rata-rata pendapatan per pekan dari transaksi yang telah terjadi.
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Sales Monitoring</i> 2. Klik label <i>Weeks</i> 3. Klik label <i>AverageTransaction</i>
Hasil pengujian yang diharapkan	Sistem menampilkan rata-rata pendapatan transaksi di setiap pekan dalam visualisasi plot pada grafik <i>line chart</i> .
Hasil pengujian	Sistem menampilkan rata-rata pendapatan transaksi di setiap pekan dalam visualisasi plot pada grafik <i>line chart</i> .
Status	Valid

6.3.17 Lihat Rata-Rata Pendapatan Transaksi Per Bulan

Pengujian validasi lihat rata-rata transaksi per bulan akan menguji kebutuhan fungsional POSEW-F-17. Pada pengujian ini, dilakukan proses menampilkan rata-rata pendapatan transaksi dalam skala waktu setiap bulan. Melalui proses ini, memungkinkan aktor untuk melihat perkembangan rata-rata pendapatan transaksi dalam visualisasi *line chart*. Pengujian validasi lihat rata-rata pendapatan transaksi per bulan dipaparkan pada Tabel 6.39.

Tabel 6.39 Pengujian validasi Lihat Rata-Rata Pendapatan Transaksi Per Bulan

Nama kasus uji	Lihat rata-rata pendapatan transaksi per bulan
Objek uji	Kebutuhan fungsional POSEW-F-17
Tujuan pengujian	Memastikan bahwa sistem menampilkan rata-rata pendapatan per bulan dari transaksi yang telah terjadi.
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Sales Monitoring</i> 2. Klik label <i>Months</i> 3. Klik label <i>AverageTransaction</i>
Hasil pengujian yang diharapkan	Sistem menampilkan rata-rata pendapatan transaksi di setiap bulan dalam visualisasi plot pada grafik <i>line chart</i> .
Hasil pengujian	Sistem menampilkan rata-rata pendapatan transaksi di setiap bulan dalam visualisasi plot pada grafik <i>line chart</i> .
Status	Valid

6.3.18 Lihat Prediksi Pendapatan

Pengujian validasi lihat prediksi pendapatan akan menguji kebutuhan fungsional POSEW-F-18. Pada pengujian ini, dilakukan proses menampilkan prediksi pendapatan untuk empat pekan yang akan datang. Melalui proses ini, memungkinkan aktor untuk melihat prediksi pendapatan dalam visualisasi *line chart*. Pengujian validasi lihat prediksi pendapatan dipaparkan pada Tabel 6.40.

Tabel 6.40 Pengujian validasi Lihat Prediksi Pendapatan

Nama kasus uji	Lihat prediksi pendapatan
Objek uji	Kebutuhan fungsional POSEW-F-18
Tujuan pengujian	Memastikan bahwa sistem menampilkan prediksi pendapatan.
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Sales Monitoring</i>
Hasil pengujian yang diharapkan	Sistem menampilkan prediksi pendapatan untuk empat pekan yang akan datang dalam visualisasi grafik <i>line chart</i> .
Hasil pengujian	Sistem menampilkan prediksi pendapatan untuk empat pekan yang akan datang dalam visualisasi grafik <i>line chart</i> .
Status	Valid

6.3.19 Lihat 10 Hari Dengan Pendapatan Tertinggi

Pengujian validasi lihat prediksi pendapatan akan menguji kebutuhan fungsional POSEW-F-19. Pada pengujian ini, dilakukan proses menampilkan 10 hari dengan pendapatan tertinggi. Melalui proses ini, memungkinkan aktor untuk melihat 10 hari dengan pendapatan tertinggi dalam visualisasi *area chart*. Pengujian validasi lihat 10 hari dengan pendapatan tertinggi dipaparkan pada Tabel 6.41.

Tabel 6.41 Pengujian validasi Lihat 10 Hari Dengan Pendapatan Tertinggi

Nama kasus uji	Lihat 10 hari dengan pendapatan tertinggi
Objek uji	Kebutuhan fungsional POSEW-F-19
Tujuan pengujian	Memastikan bahwa sistem menampilkan 10 hari dengan pendapatan tertinggi.
Prosedur uji	1. Klik menu <i>Sales Monitoring</i>
Hasil pengujian yang diharapkan	Sistem menampilkan 10 hari dengan pendapatan tertinggi dengan informasi yang disajikan adalah tanggal dan total pendapatan dari transaksi yang terjadi pada tanggal tersebut dalam visualisasi grafik <i>area chart</i> .
Hasil pengujian	Sistem menampilkan 10 hari dengan pendapatan tertinggi dengan informasi yang disajikan adalah tanggal dan total pendapatan dari transaksi yang terjadi pada tanggal tersebut dalam visualisasi grafik <i>area chart</i> .
Status	Valid

6.3.20 Lihat 10 Hari Dengan Pendapatan Terendah

Pengujian validasi lihat prediksi pendapatan akan menguji kebutuhan fungsional POSEW-F-20. Pada pengujian ini, dilakukan proses menampilkan 10 hari dengan pendapatan terendah. Melalui proses ini, memungkinkan aktor untuk melihat 10 hari dengan pendapatan terendah dalam visualisasi *area chart*. Pengujian validasi lihat 10 hari dengan pendapatan terendah dipaparkan pada Tabel 6.42.

Tabel 6.42 Pengujian validasi Lihat 10 Hari Dengan Pendapatan Terendah

Nama kasus uji	Lihat 10 hari dengan pendapatan terendah
Objek uji	Kebutuhan fungsional POSEW-F-20
Tujuan pengujian	Memastikan bahwa sistem menampilkan 10 hari dengan pendapatan terendah.
Prosedur uji	1. Klik menu <i>Sales Monitoring</i>
Hasil pengujian yang diharapkan	Sistem menampilkan 10 hari dengan pendapatan terendah dengan informasi yang disajikan adalah tanggal dan total pendapatan dari transaksi yang terjadi pada tanggal tersebut dalam visualisasi grafik <i>area chart</i> .
Hasil pengujian	Sistem menampilkan 10 hari dengan pendapatan terendah dengan informasi yang disajikan adalah tanggal dan total pendapatan dari transaksi yang terjadi pada tanggal tersebut dalam visualisasi grafik <i>area chart</i> .
Status	Valid



6.3.21 Lihat Riwayat Transaksi

Pengujian validasi lihat riwayat transaksi akan menguji kebutuhan fungsional POSEW-F-21. Pada pengujian ini, dilakukan proses menampilkan riwayat transaksi. Melalui proses ini, memungkinkan aktor untuk melihat kumpulan transaksi yang pernah terjadi. Pengujian validasi lihat riwayat transaksi dipaparkan pada Tabel 6.43.

Tabel 6.43 Pengujian validasi Lihat Riwayat Transaksi

Nama kasus uji	Lihat riwayat transaksi
Objek uji	Kebutuhan fungsional POSEW-F-21
Tujuan pengujian	Memastikan bahwa sistem menampilkan riwayat transaksi
Prosedur uji	1. Klik menu <i>History</i>
Hasil pengujian yang diharapkan	Sistem menampilkan riwayat transaksi dengan informasi yang disajikan adalah kode transaksi, tanggal transaksi, diskon pada transaksi, metode pembayaran transaksi, dan total biaya transaksi, total transaksi, dan total pendapatan dalam representasi tabel.
Hasil pengujian	Sistem menampilkan riwayat transaksi dengan informasi yang disajikan adalah kode transaksi, tanggal transaksi, diskon pada transaksi, metode pembayaran transaksi, dan total biaya transaksi, total transaksi, dan total pendapatan dalam representasi tabel., dan total pendapatan dalam representasi tabel.
Status	Valid

6.3.22 Lihat Rincian Riwayat Transaksi

Pengujian validasi lihat rincian riwayat transaksi akan menguji kebutuhan fungsional POSEW-F-22. Pada pengujian ini, dilakukan proses menampilkan rincian riwayat transaksi. Melalui proses ini, memungkinkan aktor untuk melihat rincian dari kumpulan transaksi yang pernah terjadi. Pengujian validasi lihat riwayat transaksi dipaparkan pada Tabel 6.44 dan Tabel 6.45.

Tabel 6.44 Pengujian validasi Lihat Riwayat Transaksi

Nama kasus uji	Lihat riwayat transaksi
Objek uji	Kebutuhan fungsional POSEW-F-22
Tujuan pengujian	Memastikan bahwa sistem menampilkan rincian dari riwayat transaksi
Prosedur uji	1. Klik menu <i>History</i> 2. Klik icon <i>expand</i> pada daftar riwayat transaksi

Tabel 6.45 Pengujian validasi Lihat Riwayat Transaksi (Lanjutan)

Hasil pengujian yang diharapkan	Sistem menampilkan rincian dari riwayat transaksi dengan informasi yang disajikan adalah daftar nama produk, jumlah produk, dan harga produk pada transaksi yang bersangkutan.
Hasil pengujian	Sistem menampilkan rincian dari riwayat transaksi dengan informasi yang disajikan adalah daftar nama produk, jumlah produk, dan harga produk pada transaksi yang bersangkutan.
Status	Valid

6.3.23 Hapus Transaksi

Pengujian validasi hapus transaksi akan menguji kebutuhan fungsional POSEW-F-23. Pada pengujian ini, dilakukan proses menghapus transaksi yang pernah terjadi. Pengujian validasi hapus transaksi dipaparkan pada Tabel 6.46 sedangkan pengujian validasi untuk proses alternatif ditampilkan pada Tabel 6.47 dan Tabel 6.48.

Tabel 6.46 Pengujian validasi Hapus Transaksi

Nama kasus uji	Hapus transaksi
Objek uji	Kebutuhan fungsional POSEW-F-23
Tujuan pengujian	Memastikan bahwa sistem dapat menghapus transaksi yang telah terjadi.
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>History</i> 2. Klik tombol <i>Delete</i> pada transaksi yang ingin dihapus 3. Klik tombol <i>Yes</i> pada peringatan "<i>Are you sure to delete this record ?</i>"
Hasil pengujian yang diharapkan	Sistem menghapus transaksi yang dipilih dan menampilkan data riwayat transaksi terkini.
Hasil pengujian	Sistem menghapus transaksi yang dipilih dan menampilkan data riwayat transaksi terkini.
Status	Valid

Tabel 6.47 Pengujian validasi Hapus Transaksi – Alternatif 1

Nama kasus uji	Hapus transaksi
Objek uji	Kebutuhan fungsional POSEW-F-23
Tujuan pengujian	Memastikan bahwa sistem dapat membatalkan penghapusan transaksi
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>History</i> 2. Klik tombol <i>Delete</i> pada transaksi yang ingin dihapus 3. Klik tombol <i>No</i> pada peringatan "<i>Are you sure to delete this record ?</i>"

Tabel 6.48 Pengujian validasi Hapus Transaksi – Alternatif 1 (Lanjutan)

Hasil pengujian yang diharapkan	Sistem tidak menghapus transaksi yang dipilih dan menampilkan data riwayat transaksi terkini
Hasil pengujian	Sistem tidak menghapus transaksi yang dipilih dan menampilkan data riwayat transaksi terkini
Status	Valid

6.3.24 Cari Transaksi

Pengujian validasi cari transaksi akan menguji kebutuhan fungsional POSEW-F-24. Pada pengujian ini, dilakukan proses pencarian transaksi yang pernah terjadi. Pengujian validasi cari transaksi dipaparkan pada Tabel 6.49.

Tabel 6.49 Pengujian validasi Cari Transaksi

Nama kasus uji	Cari transaksi
Objek uji	Kebutuhan fungsional POSEW-F-24
Tujuan pengujian	Memastikan bahwa sistem menampilkan data transaksi sesuai dengan masukan pada pencarian
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>History</i> 2. Klik <i>box-area</i> pada kolom riwayat transaksi. 3. Mengisi kolom pencarian 4. Klik filter
Hasil pengujian yang diharapkan	Sistem menampilkan data yang sesuai dengan masukan data yang diberikan.
Hasil pengujian	Sistem menampilkan data yang sesuai dengan masukan data yang diberikan.
Status	Valid

6.3.25 Urut Transaksi

Pengujian validasi urut transaksi akan menguji kebutuhan fungsional POSEW-F-25. Pada pengujian ini, dilakukan proses pengurutan transaksi yang pernah terjadi. Pengujian validasi urut transaksi dipaparkan pada Tabel 6.50 sedangkan pengujian validasi untuk proses alternatif ditampilkan pada Tabel 6.51 dan Tabel 6.52.

Tabel 6.50 Pengujian validasi Urut Transaksi

Nama kasus uji	Urut transaksi
Objek uji	Kebutuhan fungsional POSEW-F-25
Tujuan pengujian	Memastikan bahwa sistem menampilkan data transaksi sesuai dengan urutan dari posisi terendah hingga tertinggi
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>History</i> 2. Klik pada salah satu kolom riwayat transaksi

Tabel 6.51 Pengujian validasi Urut Transaksi (Lanjutan)

Hasil pengujian yang diharapkan	Sistem menampilkan data dengan urutan yang dimulai dari posisi terendah hingga tertinggi.
Hasil pengujian	Sistem menampilkan data dengan urutan yang dimulai dari posisi terendah hingga tertinggi.
Status	Valid

Tabel 6.52 Pengujian validasi Urut Transaksi – Alternatif 1

Nama kasus uji	Urut transaksi
Objek uji	Kebutuhan fungsional POSEW-F-25
Tujuan pengujian	Memastikan bahwa sistem menampilkan data transaksi sesuai dengan urutan dari posisi tertinggi hingga terendah
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>History</i> 2. Klik sebanyak dua kali pada salah satu kolom riwayat transaksi
Hasil pengujian yang diharapkan	Sistem menampilkan data transaksi sesuai dengan urutan dari posisi tertinggi hingga terendah
Hasil pengujian	Sistem menampilkan data transaksi sesuai dengan urutan dari posisi tertinggi hingga terendah
Status	Valid

6.3.26 Lihat Kombinasi Produk

Pengujian validasi lihat kombinasi produk akan menguji kebutuhan fungsional POSEW-F-26. Pada pengujian ini, dilakukan proses menampilkan kombinasi produk. Pengujian validasi lihat kombinasi produk dipaparkan pada Tabel 6.53 dan Tabel 6.54.

Tabel 6.53 Pengujian validasi Lihat Kombinasi Produk

Nama kasus uji	Lihat kombinasi produk
Objek uji	Kebutuhan fungsional POSEW-F-26
Tujuan pengujian	Memastikan bahwa sistem menampilkan kombinasi produk
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Trend - Analysis</i>
Hasil pengujian yang diharapkan	Sistem menampilkan kombinasi produk yang bertemu dari transaksi yang telah terjadi dengan informasi yang disajikan adalah nama produk yang di kombinasi kan dan jumlah kombinasi produk tersebut bertemu.

Tabel 6.54 Pengujian validasi Lihat Kombinasi Produk (Lanjutan)

Hasil pengujian	Sistem menampilkan kombinasi produk yang bertemu dari transaksi yang telah terjadi dengan informasi yang disajikan adalah nama produk yang di kombinasi kan dan jumlah kombinasi produk tersebut bertemu.
Status	Valid

6.3.27 Cari Kombinasi Produk

Pengujian validasi cari kombinasi produk akan menguji kebutuhan fungsional POSEW-F-27. Pada pengujian ini, dilakukan proses pencarian kombinasi produk yang pernah bertemu dalam satu transaksi. Pengujian validasi cari kombinasi produk dipaparkan pada Tabel 6.55.

Tabel 6.55 Pengujian validasi Cari Kombinasi Produk

Nama kasus uji	Cari kombinasi produk
Objek uji	Kebutuhan fungsional POSEW-F-27
Tujuan pengujian	Memastikan bahwa sistem menampilkan data kombinasi produk sesuai dengan masukan pada pencarian
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Trends-Analysis</i> 2. Mengisi kolom pencarian
Hasil pengujian yang diharapkan	Sistem menampilkan data yang sesuai dengan masukan data yang diberikan.
Hasil pengujian	Sistem menampilkan data yang sesuai dengan masukan data yang diberikan.
Status	Valid

6.3.28 Urut Kombinasi Produk

Pengujian validasi urut kombinasi produk akan menguji kebutuhan fungsional POSEW-F-28. Pada pengujian ini, dilakukan proses pengurutan kombinasi produk yang pernah bertemu dalam satu transaksi. Pengujian validasi urut kombinasi produk dipaparkan pada Tabel 6.56 dan Tabel 6.57 sedangkan pengujian validasi untuk proses alternatif ditampilkan pada Tabel 6.58.

Tabel 6.56 Pengujian validasi Urut Kombinasi Produk

Nama kasus uji	Urut transaksi
Objek uji	Kebutuhan fungsional POSEW-F-28
Tujuan pengujian	Memastikan bahwa sistem menampilkan data transaksi sesuai dengan urutan dari posisi terendah hingga tertinggi
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Trend - Analysis</i> 2. Klik pada salah satu kolom kombinasi produk

Tabel 6.57 Pengujian validasi Urut Kombinasi Produk (Lanjutan)

Hasil pengujian yang diharapkan	Sistem menampilkan data dengan urutan yang dimulai dari posisi terendah hingga tertinggi.
Hasil pengujian	Sistem menampilkan data dengan urutan yang dimulai dari posisi terendah hingga tertinggi.
Status	Valid

Tabel 6.58 Pengujian validasi Urut Kombinasi Produk – alternatif 1

Nama kasus uji	Urut transaksi
Objek uji	Kebutuhan fungsional POSEW-F-28
Tujuan pengujian	Memastikan bahwa sistem menampilkan data transaksi sesuai dengan urutan dari posisi tertinggi hingga terendah
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Trend – Analysis</i> 2. Klik sebanyak dua kali pada salah satu kolom kombinasi produk
Hasil pengujian yang diharapkan	Sistem menampilkan data kombinasi produk sesuai dengan urutan dari posisi tertinggi hingga terendah
Hasil pengujian	Sistem menampilkan data kombinasi produk sesuai dengan urutan dari posisi tertinggi hingga terendah
Status	Valid

6.3.29 Lihat Jumlah Produk Terjual

Pengujian validasi lihat jumlah produk terjual akan menguji kebutuhan fungsional POSEW-F-29. Pada pengujian ini, dilakukan proses menampilkan jumlah produk terjual. Pengujian validasi lihat jumlah produk terjual dipaparkan pada Tabel 6.59.

Tabel 6.59 Pengujian validasi Lihat Jumlah Produk Terjual

Nama kasus uji	Lihat jumlah produk terjual
Objek uji	Kebutuhan fungsional POSEW-F-29
Tujuan pengujian	Memastikan bahwa sistem menampilkan jumlah produk terjual
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Trend - Analysis</i>
Hasil pengujian yang diharapkan	Sistem menampilkan jumlah produk yang telah terjual dengan informasi yang disajikan adalah nama produk dan jumlah produk terjual.
Hasil pengujian	Sistem menampilkan jumlah produk yang telah terjual dengan informasi yang disajikan adalah nama produk dan jumlah produk terjual.
Status	Valid

6.3.30 Cari Jumlah Produk Terjual

Pengujian validasi cari jumlah produk terjual akan menguji kebutuhan fungsional POSEW-F-30. Pada pengujian ini, dilakukan proses pencarian kombinasi produk yang pernah bertemu dalam satu transaksi. Pengujian validasi cari kombinasi produk dipaparkan pada Tabel 6.60.

Tabel 6.60 Pengujian validasi Cari Jumlah Produk Terjual

Nama kasus uji	Cari jumlah produk terjual
Objek uji	Kebutuhan fungsional POSEW-F-30
Tujuan pengujian	Memastikan bahwa sistem menampilkan data jumlah produk terjual sesuai dengan masukan pada pencarian
Prosedur uji	1. Klik menu <i>Trends-Analysis</i> 2. Mengisi kolom pencarian
Hasil pengujian yang diharapkan	Sistem menampilkan data produk yang sesuai dengan masukan data yang diberikan.
Hasil pengujian	Sistem menampilkan data produk yang sesuai dengan masukan data yang diberikan.
Status	Valid

6.3.31 Urut Jumlah Produk Terjual

Pengujian validasi urut jumlah produk terjual akan menguji kebutuhan fungsional POSEW-F-31. Pada pengujian ini, dilakukan proses pengurutan jumlah produk terjual. Pengujian validasi urut jumlah produk terjual dipaparkan pada Tabel 6.61 sedangkan pengujian validasi untuk proses alternatif ditampilkan pada Tabel 6.62.

Tabel 6.61 Pengujian validasi Urut Jumlah Produk Terjual

Nama kasus uji	Urut jumlah produk terjual
Objek uji	Kebutuhan fungsional POSEW-F-31
Tujuan pengujian	Memastikan bahwa sistem menampilkan data jumlah produk terjual sesuai dengan urutan dari posisi terendah hingga tertinggi
Prosedur uji	1. Klik menu <i>Trend - Analysis</i> 2. Klik <i>title</i> pada salah satu kolom jumlah produk terjual.
Hasil pengujian yang diharapkan	Sistem mengurutkan data jumlah produk terjual dengan urutan yang dimulai dari posisi terendah hingga tertinggi.
Hasil pengujian	Sistem mengurutkan data jumlah produk terjual dengan urutan yang dimulai dari posisi terendah hingga tertinggi.
Status	Valid

Tabel 6.62 Pengujian validasi Urut Jumlah Produk Terjual – Alternatif 1

Nama kasus uji	Urut jumlah produk terjual
Objek uji	Kebutuhan fungsional POSEW-F-31
Tujuan pengujian	Memastikan bahwa sistem menampilkan data jumlah produk terjual sesuai dengan urutan dari posisi tertinggi hingga terendah
Prosedur uji	1. Klik menu <i>Trend - Analysis</i> 2. Klik <i>title</i> pada salah satu kolom jumlah produk terjual.
Hasil pengujian yang diharapkan	Sistem mengurutkan data jumlah produk terjual dengan urutan yang dimulai dari posisi tertinggi hingga terendah.
Hasil pengujian	Sistem mengurutkan data jumlah produk terjual dengan urutan yang dimulai dari posisi tertinggi hingga terendah.
Status	Valid

6.3.32 Lihat Daftar Produk

Pengujian validasi lihat daftar produk akan menguji kebutuhan fungsional POSEW-F-32. Pada pengujian ini, dilakukan proses menampilkan daftar produk yang menampilkan produk-produk yang telah terdaftar. Pengujian validasi lihat daftar produk dipaparkan pada Tabel 6.63.

Tabel 6.63 Pengujian validasi Lihat Daftar Produk

Nama kasus uji	Lihat daftar produk
Objek uji	Kebutuhan fungsional POSEW-F-32
Tujuan pengujian	Memastikan bahwa sistem dapat menampilkan daftar produk
Prosedur uji	1. Klik menu <i>Products</i>
Hasil pengujian yang diharapkan	Sistem menampilkan daftar produk yang telah terdaftar dengan informasi yang disajikan adalah kode produk, nama produk, kategori produk, dan harga produk.
Hasil pengujian	Sistem menampilkan daftar produk yang telah terdaftar dengan informasi yang disajikan adalah kode produk, nama produk, kategori produk, dan harga produk.
Status	Valid

6.3.33 Cari Produk

Pengujian validasi cari produk akan menguji kebutuhan fungsional POSEW-F-33. Pada pengujian ini, dilakukan proses pencarian produk yang telah terdaftar. Pengujian validasi cari produk dipaparkan pada Tabel 6.64.

Tabel 6.64 Pengujian validasi Cari Produk

Nama kasus uji	Cari produk
Objek uji	Kebutuhan fungsional POSEW-F-33
Tujuan pengujian	Memastikan bahwa sistem dapat menampilkan data produk yang telah terdaftar sesuai dengan masukan pada pencarian
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Products</i> 2. Mengisi kolom pencarian
Hasil pengujian yang diharapkan	Sistem menampilkan hasil pencarian data produk yang telah terdaftar berdasarkan masukan data yang dimasukkan.
Hasil pengujian	Sistem menampilkan hasil pencarian data produk yang telah terdaftar berdasarkan masukan data yang dimasukkan.
Status	Valid

6.3.34 Tambah Produk

Pengujian validasi tambah produk akan menguji kebutuhan fungsional POSEW-F-34. Pada pengujian ini, dilakukan proses penambahan produk. Pengujian validasi tambah produk dipaparkan pada Tabel 6.65 sedangkan pengujian validasi untuk proses alternatif ditampilkan pada Tabel 6.66 - 6.70.

Tabel 6.65 Pengujian validasi Tambah Produk

Nama kasus uji	Tambah produk
Objek uji	Kebutuhan fungsional POSEW-F-34
Tujuan pengujian	Memastikan bahwa sistem dapat menambahkan produk
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Products</i> 2. Klik tombol <i>Add Product</i> 3. Mengisi <i>Product Code</i>, <i>Product Name</i>, <i>Product Category</i>, dan <i>Product Price</i> 4. Klik tombol <i>Add Product</i>
Hasil pengujian yang diharapkan	Sistem menampilkan peringatan " <i>New Product Successfully Added !</i> ".
Hasil pengujian	Sistem menampilkan peringatan " <i>New Product Successfully Added !</i> ".
Status	Valid

Tabel 6.66 Pengujian validasi Tambah Produk – Alternatif 1

Nama kasus uji	Tambah produk
Objek uji	Kebutuhan fungsional POSEW-F-34
Tujuan pengujian	Memastikan bahwa sistem tidak dapat menambahkan produk apabila kode produk telah terdaftar

Tabel 6.67 Pengujian validasi Tambah Produk – Alternatif 1 (Lanjutan)

Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Products</i> 2. Klik tombol <i>Add Product</i> 3. Mengisi <i>Product Code</i> yang telah terdaftar, <i>Product Name</i>, <i>Product Category</i>, dan <i>Product Price</i> 4. Klik tombol <i>Add Product</i>
Hasil pengujian yang diharapkan	Sistem menampilkan peringatan " <i>Adding Product Failed (Product Code Already Exist) – Please Try Again</i> "
Hasil pengujian	Sistem menampilkan peringatan " <i>Adding Product Failed (Product Code Already Exist) – Please Try Again</i> "
Status	Valid

Tabel 6.68 Pengujian validasi Tambah Produk – Alternatif 2

Nama kasus uji	Tambah produk
Objek uji	Kebutuhan fungsional POSEW-F-34
Tujuan pengujian	Memastikan bahwa sistem tidak dapat menambahkan produk apabila terdapat masukan data yang kosong
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Products</i> 2. Klik tombol <i>Add Product</i> 3. Mengisi <i>Product Code</i>, <i>Product Name</i>, dan <i>Product Category</i> tanpa mengisi <i>Product Price</i> 4. Klik tombol <i>Add Product</i>
Hasil pengujian yang diharapkan	Sistem menampilkan peringatan " <i>Error – Please Try Again and Fill The Form Completely</i> "
Hasil pengujian	Sistem menampilkan peringatan " <i>Error – Please Try Again and Fill The Form Completely</i> "
Status	Valid

Tabel 6.69 Pengujian validasi Tambah Produk – Alternatif 3

Nama kasus uji	Tambah produk
Objek uji	Kebutuhan fungsional POSEW-F-34
Tujuan pengujian	Memastikan bahwa sistem dapat membatalkan proses penambahan produk
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Products</i> 2. Klik tombol <i>Add Product</i> 3. Mengisi <i>Product Code</i>, <i>Product Name</i>, dan <i>Product Category</i> tanpa mengisi <i>Product Price</i> 4. Klik tombol <i>Cancel</i>
Hasil pengujian yang diharapkan	Sistem menutup <i>form Add Product</i> .

Tabel 6.70 Pengujian validasi Tambah Produk – Alternatif 3 (Lanjutan)

Hasil pengujian	Sistem menutup <i>form Add Product</i> .
Status	Valid

6.3.35 Hapus Produk

Pengujian validasi hapus produk akan menguji kebutuhan fungsional POSEW-F-35. Pada pengujian ini, dilakukan proses menghapus produk yang telah terdaftar. Pengujian validasi hapus produk dipaparkan pada Tabel 6.71 sedangkan pengujian validasi untuk proses alternatif ditampilkan pada Tabel 6.72.

Tabel 6.71 Pengujian validasi Hapus Produk

Nama kasus uji	Hapus produk
Objek uji	Kebutuhan fungsional POSEW-F-35
Tujuan pengujian	Memastikan bahwa sistem dapat menghapus produk yang telah terdaftar
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Products</i> 2. Klik tombol <i>Delete</i> pada produk yang ingin dihapus 3. Klik tombol <i>Yes</i> pada peringatan "<i>Are you sure you want to delete this record ?</i>"
Hasil pengujian yang diharapkan	Sistem menghapus data produk yang dipilih dan menampilkan data produk terkini.
Hasil pengujian	Sistem menghapus data produk yang dipilih dan menampilkan data produk terkini.
Status	Valid

Tabel 6.72 Pengujian validasi Hapus Produk – Alternatif 1

Nama kasus uji	Hapus produk
Objek uji	Kebutuhan fungsional POSEW-F-35
Tujuan pengujian	Memastikan bahwa sistem dapat membatalkan penghapusan produk
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Products</i> 2. Klik tombol <i>Delete</i> pada produk yang ingin dihapus 3. Klik tombol <i>No</i> pada peringatan "<i>Are you sure you want to delete this record ?</i>"
Hasil pengujian yang diharapkan	Sistem menampilkan data produk terkini dengan masih tersedianya produk yang ingin dihapus sebelumnya.
Hasil pengujian	Sistem menampilkan data produk terkini dengan masih tersedianya produk yang ingin dihapus sebelumnya.
Status	Valid



6.3.36 Lihat Daftar Kategori

Pengujian validasi lihat daftar kategori akan menguji kebutuhan fungsional POSEW-F-36. Pada pengujian ini, dilakukan proses menampilkan daftar kategori. Pengujian validasi lihat daftar kategori dipaparkan pada Tabel 6.73.

Tabel 6.73 Pengujian validasi Lihat Daftar Kategori

Nama kasus uji	Lihat daftar kategori
Objek uji	Kebutuhan fungsional POSEW-F-36
Tujuan pengujian	Memastikan bahwa sistem menampilkan daftar kategori
Prosedur uji	1. Klik menu <i>Category</i>
Hasil pengujian yang diharapkan	Sistem menampilkan daftar kategori yang telah terdaftar dengan informasi yang disajikan adalah nama kategori.
Hasil pengujian	Sistem menampilkan daftar kategori yang telah terdaftar dengan informasi yang disajikan adalah nama kategori.
Status	Valid

6.3.37 Cari Kategori

Pengujian validasi cari kategori akan menguji kebutuhan fungsional POSEW-F-37. Pada pengujian ini, dilakukan proses pencarian kategori yang telah terdaftar. Pengujian validasi cari kategori dipaparkan pada Tabel 6.74.

Tabel 6.74 Pengujian validasi Cari Kategori

Nama kasus uji	Cari kategori
Objek uji	Kebutuhan fungsional POSEW-F-37
Tujuan pengujian	Memastikan bahwa sistem menampilkan data kategori yang telah terdaftar sesuai dengan masukan pada pencarian
Prosedur uji	1. Klik menu <i>Category</i> 2. Mengisi kolom pencarian
Hasil pengujian yang diharapkan	Sistem menampilkan data kategori yang sesuai dengan masukan data yang diberikan.
Hasil pengujian	Sistem menampilkan data kategori yang sesuai dengan masukan data yang diberikan.
Status	Valid

6.3.38 Tambah Kategori

Pengujian validasi tambah produk akan menguji kebutuhan fungsional POSEW-F-38. Pada pengujian ini, dilakukan proses penambahan kategori. Pengujian validasi tambah kategori dipaparkan pada Tabel 6.75 sedangkan pengujian validasi untuk proses alternatif ditampilkan pada Tabel 6.76 - 6.78.

Tabel 6.75 Pengujian validasi Tambah Kategori

Nama kasus uji	Tambah kategori
Objek uji	Kebutuhan fungsional POSEW-F-38
Tujuan pengujian	Memastikan bahwa sistem dapat menambahkan kategori
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Menu - Category</i> 2. Klik tombol <i>Add Category</i> 3. Mengisi <i>Product Category</i> 4. Klik tombol <i>Add Category</i>
Hasil pengujian yang diharapkan	Sistem menampilkan peringatan " <i>New Category Successfully Added !</i> "
Hasil pengujian	Sistem menampilkan peringatan " <i>New Category Successfully Added !</i> "
Status	Valid

Tabel 6.76 Pengujian validasi Tambah Kategori – Alternatif 1

Nama kasus uji	Tambah kategori
Objek uji	Kebutuhan fungsional POSEW-F-38
Tujuan pengujian	Memastikan bahwa sistem tidak dapat menambahkan kategori apabila nama kategori telah terdaftar
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Menu - Category</i> 2. Klik tombol <i>Add Category</i> 3. Mengisi <i>Category Name</i> dengan kategori yang telah terdaftar 4. Klik tombol <i>Add Category</i>
Hasil pengujian yang diharapkan	Sistem menampilkan peringatan " <i>Category Is Already Exist – Please Try Again</i> "
Hasil pengujian	Sistem menampilkan peringatan " <i>Category Is Already Exist – Please Try Again</i> "
Status	Valid

Tabel 6.77 Pengujian validasi Tambah Produk – Alternatif 2

Nama kasus uji	Tambah kategori
Objek uji	Kebutuhan fungsional POSEW-F-38
Tujuan pengujian	Memastikan bahwa sistem tidak dapat menambahkan kategori apabila nama kategori di kosongkan
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Menu - Category</i> 2. Klik tombol <i>Add Category</i> 3. Klik tombol <i>Add Category</i>
Hasil pengujian yang diharapkan	Sistem menampilkan peringatan " <i>Error – Please Try Again amd Fill The Category Name Correctly</i> "



Tabel 6.78 Pengujian validasi Tambah Produk – Alternatif 2 (Lanjutan)

Hasil pengujian	Sistem menampilkan peringatan “ <i>Error – Please Try Again and Fill The Category Name Correctly</i> ”
Status	Valid

6.3.39 Hapus Kategori

Pengujian validasi hapus kategori akan menguji kebutuhan fungsional POSEW-F-39. Pada pengujian ini, dilakukan proses menghapus kategori yang telah terdaftar. Pengujian validasi hapus kategori dipaparkan pada Tabel 6.79 sedangkan pengujian validasi untuk proses alternatif ditampilkan pada Tabel 6.80.

Tabel 6.79 Pengujian validasi Hapus Kategori

Nama kasus uji	Hapus kategori
Objek uji	Kebutuhan fungsional POSEW-F-39
Tujuan pengujian	Memastikan bahwa sistem dapat menghapus produk yang telah terdaftar
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Products</i> 2. Klik tombol <i>Delete</i> pada produk yang ingin dihapus 3. Klik tombol <i>Yes</i> pada peringatan “<i>Are you sure to delete this record ?</i>”
Hasil pengujian yang diharapkan	Sistem menghapus data kategori yang dipilih dan menampilkan data kategori terkini.
Hasil pengujian	Sistem menghapus data kategori yang dipilih dan menampilkan data kategori terkini.
Status	Valid

Tabel 6.80 Pengujian validasi Hapus Produk – Alternatif 1

Nama kasus uji	Hapus produk
Objek uji	Kebutuhan fungsional POSEW-F-39
Tujuan pengujian	Memastikan bahwa sistem dapat membatalkan penghapusan produk
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Products</i> 2. Klik tombol <i>Delete</i> pada produk yang ingin dihapus 3. Klik tombol <i>No</i> pada peringatan “<i>Are you sure you want to delete this record ?</i>”
Hasil pengujian yang diharapkan	Sistem menampilkan data produk terkini dengan masih tersedianya produk yang ingin dihapus sebelumnya.
Hasil pengujian	Sistem menampilkan data produk terkini dengan masih tersedianya produk yang ingin dihapus sebelumnya.
Status	Valid



6.3.40 Lihat Daftar Akun

Pengujian validasi lihat daftar akun akan menguji kebutuhan fungsional POSEW-F-40. Pada pengujian ini, dilakukan proses menampilkan daftar akun. Pengujian validasi lihat daftar akun dipaparkan pada Tabel 6.81.

Tabel 6.81 Pengujian validasi Lihat Daftar Akun

Nama kasus uji	Lihat daftar akun
Objek uji	Kebutuhan fungsional POSEW-F-40
Tujuan pengujian	Memastikan bahwa sistem menampilkan daftar akun
Prosedur uji	1. Klik menu <i>Accounts</i>
Hasil pengujian yang diharapkan	Sistem menampilkan daftar akun yang telah terdaftar dengan informasi yang disajikan adalah <i>username</i> dan <i>role</i> .
Hasil pengujian	Sistem menampilkan daftar akun yang telah terdaftar dengan informasi yang disajikan adalah <i>username</i> dan <i>role</i> .
Status	Valid

6.3.41 Cari Akun

Pengujian validasi cari akun akan menguji kebutuhan fungsional POSEW-F-41. Pada pengujian ini, dilakukan proses pencarian akun yang telah terdaftar. Pengujian validasi cari akun dipaparkan pada Tabel 6.82.

Tabel 6.82 Pengujian validasi Cari Akun

Nama kasus uji	Cari akun
Objek uji	Kebutuhan fungsional POSEW-F-41
Tujuan pengujian	Memastikan bahwa sistem dapat mencari akun yang telah terdaftar.
Prosedur uji	1. Klik menu <i>Accounts</i> 2. Mengisi kolom pencarian
Hasil pengujian yang diharapkan	Sistem menampilkan data akun yang sesuai dengan masukan data yang diberikan.
Hasil pengujian	Sistem menampilkan data akun yang sesuai dengan masukan data yang diberikan.
Status	Valid

6.3.42 Tambah Akun

Pengujian validasi tambah akun akan menguji kebutuhan fungsional POSEW-F-42. Pada pengujian ini, dilakukan proses penambahan akun. Pengujian validasi tambah akun dipaparkan pada Tabel 6.83 sedangkan pengujian validasi untuk proses alternatif ditampilkan pada Tabel 6.84 hingga Tabel 6.86.

Tabel 6.83 Pengujian validasi Tambah Akun

Nama kasus uji	Tambah akun
Objek uji	Kebutuhan fungsional POSEW-F-42
Tujuan pengujian	Memastikan bahwa sistem dapat menambahkan akun
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Accounts</i> 2. Klik tombol <i>Add New Account</i> 3. Mengisi <i>Username, Password</i>, dan memilih <i>Role</i> 4. Klik tombol <i>Add Account</i>
Hasil pengujian yang diharapkan	Sistem menampilkan peringatan " <i>New Account Successfully Added</i> "
Hasil pengujian	Sistem menampilkan peringatan " <i>New Account Successfully Added</i> "
Status	Valid

Tabel 6.84 Pengujian validasi Tambah Akun – Alternatif 1

Nama kasus uji	Tambah akun
Objek uji	Kebutuhan fungsional POSEW-F-42
Tujuan pengujian	Memastikan bahwa sistem tidak dapat menambahkan akun apabila masukan data <i>username</i> telah terdaftar
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Accounts</i> 2. Klik tombol <i>Add New Account</i> 3. Mengisi <i>Username</i> yang telah terdaftar, <i>Password</i>, dan memilih <i>Role</i> 4. Klik tombol <i>Add Account</i>
Hasil pengujian yang diharapkan	Sistem menampilkan peringatan " <i>Username Is Already Exist – Please Try Again</i> "
Hasil pengujian	Sistem menampilkan peringatan " <i>Username Is Already Exist – Please Try Again</i> "
Status	Valid

Tabel 6.85 Pengujian validasi Tambah Produk – Alternatif 2

Nama kasus uji	Tambah akun
Objek uji	Kebutuhan fungsional POSEW-F-42
Tujuan pengujian	Memastikan bahwa sistem tidak dapat menambahkan akun apabila terdapat masukan data pada <i>form</i> yang tidak di isi
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Accounts</i> 2. Klik tombol <i>Add New Account</i> 3. Mengisi <i>Username</i> dan memilih <i>Role</i> 4. Klik tombol <i>Add Account</i>



Tabel 6.86 Pengujian validasi Tambah Produk – Alternatif 2 (Lanjutan)

Hasil pengujian yang diharapkan	Sistem menampilkan peringatan <i>“Username / Password Must Be Filled – Please Try Again”</i>
Hasil pengujian	Sistem menampilkan peringatan <i>“Username / Password Must Be Filled – Please Try Again”</i>
Status	Valid

6.3.43 Hapus Akun

Pengujian validasi hapus akun akan menguji kebutuhan fungsional POSEW-F-43. Pada pengujian ini, dilakukan proses menghapus akun yang telah terdaftar. Pengujian validasi hapus akun dipaparkan pada Tabel 6.87 sedangkan pengujian validasi untuk proses alternatif ditampilkan pada Tabel 6.88 dan Tabel 6.89.

Tabel 6.87 Pengujian validasi Hapus Akun

Nama kasus uji	Hapus akun
Objek uji	Kebutuhan fungsional POSEW-F-43
Tujuan pengujian	Memastikan bahwa sistem dapat menghapus akun yang telah terdaftar.
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Account</i> 2. Klik tombol <i>Delete</i> pada akun yang ingin dihapus 3. Klik tombol <i>Yes</i> pada peringatan <i>“Are you sure you want to delete this record ?”</i>
Hasil pengujian yang diharapkan	Sistem menghapus data akun yang dipilih dan menampilkan daftar akun yang telah terdaftar saat ini.
Hasil pengujian	Sistem menghapus data akun yang dipilih dan menampilkan daftar akun yang telah terdaftar saat ini.
Status	Valid

Tabel 6.88 Pengujian validasi Hapus Akun – Alternatif 1

Nama kasus uji	Hapus akun
Objek uji	Kebutuhan fungsional POSEW-F-43
Tujuan pengujian	Memastikan bahwa sistem dapat menghapus akun yang telah terdaftar.
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Accounts</i> 2. Klik tombol <i>Delete</i> pada akun yang ingin dihapus 3. Klik tombol <i>Yes</i> pada peringatan <i>“Are you sure you want to delete this record ?”</i>
Hasil pengujian yang diharapkan	Sistem menghapus data akun yang dipilih dan menampilkan daftar akun yang telah terdaftar saat ini.



Tabel 6.89 Pengujian validasi Hapus Akun – Alternatif 1 (Lanjutan)

Hasil pengujian	Sistem menghapus data akun yang dipilih dan menampilkan daftar akun yang telah terdaftar saat ini.
Status	Valid

6.3.44 Atur Nilai Parameter Prediksi

Pengujian validasi atur nilai parameter prediksi akan menguji kebutuhan fungsional POSEW-F-44. Pengujian validasi atur nilai parameter prediksi dipaparkan pada Tabel 6.90 sedangkan proses alternatif ditampilkan pada Tabel 6.91 dan Tabel 6.92.

Tabel 6.90 Pengujian validasi Atur Nilai Parameter Prediksi

Nama kasus uji	Atur nilai parameter prediksi
Objek uji	Kebutuhan fungsional POSEW-F-44
Tujuan pengujian	Memastikan bahwa sistem dapat menetapkan nilai <i>alpha</i> dan <i>beta</i> .
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Mining – Predict</i> 2. Klik tombol <i>Set Parameter Value</i> 3. Memilih nilai <i>alpha</i> dan <i>beta</i> 4. Klik tombol <i>Add Parameter Value</i>
Hasil pengujian yang diharapkan	Sistem menampilkan peringatan " <i>New Parameter Value Submitted!</i> "
Hasil pengujian	Sistem menampilkan peringatan " <i>New Parameter Value Submitted!</i> "
Status	Valid

Tabel 6.91 Pengujian validasi Atur Nilai Parameter Prediksi - Alternatif 1

Nama kasus uji	Atur nilai parameter prediksi
Objek uji	Kebutuhan fungsional POSEW-F-44
Tujuan pengujian	Memastikan bahwa sistem tidak dapat menetapkan nilai <i>alpha</i> dan <i>beta</i> apabila nilai parameter tidak pada <i>range</i> yang tepat.
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Mining – Predict</i> 2. Klik tombol <i>Set Parameter Value</i> 3. Memilih nilai <i>alpha</i> 0 dan <i>beta</i> 0 4. Klik tombol <i>Add Parameter Value</i>
Hasil pengujian yang diharapkan	Sistem menampilkan peringatan " <i>Parameter Range only 0.1 to 1 – Please Try Again</i> "
Hasil pengujian	Sistem menampilkan peringatan " <i>Parameter Range only 0.1 to 1 – Please Try Again</i> "
Status	Valid

Tabel 6.92 Pengujian validasi Atur Nilai Parameter Prediksi - Alternatif 2

Nama kasus uji	Atur nilai parameter prediksi
Objek uji	Kebutuhan fungsional POSEW-F-44
Tujuan pengujian	Memastikan bahwa sistem dapat menetapkan nilai <i>alpha</i> dan <i>beta</i> .
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Mining - Predict</i> 2. Klik tombol <i>Set Parameter Value</i> 3. Klik <i>Cancel</i>
Hasil pengujian yang diharapkan	Sistem menutup <i>form</i> untuk menetapkan nilai <i>alpha</i> dan <i>beta</i>
Hasil pengujian	Sistem menutup <i>form</i> untuk menetapkan nilai <i>alpha</i> dan <i>beta</i>
Status	Valid

6.3.45 Lihat Informasi Prediksi Pendapatan

Pengujian validasi lihat informasi prediksi pendapatan akan menguji kebutuhan fungsional POSEW-F-45. Pada pengujian ini, dilakukan proses menampilkan informasi prediksi pendapatan. Pengujian validasi lihat informasi prediksi pendapatan dipaparkan pada Tabel 6.93.

Tabel 6.93 Pengujian validasi Lihat Informasi Prediksi Pendapatan

Nama kasus uji	Lihat informasi prediksi pendapatan
Objek uji	Kebutuhan fungsional POSEW-F-45
Tujuan pengujian	Memastikan bahwa sistem menampilkan nilai <i>alpha</i> dan <i>beta</i> saat ini serta visualisasi prediksi pendapatan
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Mining - Predict</i>
Hasil pengujian yang diharapkan	Sistem menampilkan nilai <i>alpha</i> dan <i>beta</i> saat ini serta visualisasi prediksi pendapatan untuk 4 pekan yang akan datang dalam visualisasi grafik <i>area chart</i>
Hasil pengujian	Sistem menampilkan nilai <i>alpha</i> dan <i>beta</i> saat ini serta visualisasi prediksi pendapatan untuk 4 pekan yang akan datang dalam visualisasi grafik <i>area chart</i>
Status	Valid

6.3.46 Atur Nilai Parameter Produk Potensial

Pengujian validasi atur nilai parameter produk potensial akan menguji kebutuhan fungsional POSEW-F-46. Pada pengujian ini, dilakukan proses penetapan nilai *minimum support*. Pengujian validasi atur nilai parameter produk potensial dipaparkan pada Tabel 6.94 sedangkan pengujian validasi untuk proses alternatif ditampilkan pada Tabel 6.95 - 6.98.

Tabel 6.94 Pengujian validasi Atur Nilai Parameter Produk Potensial

Nama kasus uji	Atur nilai parameter produk potensial
Objek uji	Kebutuhan fungsional POSEW-F-46
Tujuan pengujian	Memastikan bahwa sistem dapat mengatur nilai <i>minimum support</i> .
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Mining – Trends</i> 2. Klik tombol <i>Set Min. Support Value</i> 3. Mengisi nilai <i>minimum support</i> 4. Klik tombol <i>Add Parameter Value</i>
Hasil pengujian yang diharapkan	Sistem menampilkan peringatan “Sistem menampilkan peringatan “ <i>New Parameter Value Submitted !</i> ”
Hasil pengujian	Sistem menampilkan peringatan “Sistem menampilkan peringatan “ <i>New Parameter Value Submitted !</i> ”
Status	Valid

Tabel 6.95 Pengujian validasi Atur Nilai Parameter Produk Potensial

– Alternatif 1

Nama kasus uji	Atur nilai parameter produk potensial
Objek uji	Kebutuhan fungsional POSEW-F-46
Tujuan pengujian	Memastikan bahwa sistem tidak dapat mengatur nilai <i>minimum support</i> apabila isian data <i>minimum support</i> tidak di isi.
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Mining – Trends</i> 2. Klik tombol <i>Set Min. Support Value</i> 3. Klik tombol <i>Add Parameter Value</i>
Hasil pengujian yang diharapkan	Sistem menampilkan peringatan “ <i>Please Insert The Right Value</i> ”
Hasil pengujian	Sistem menampilkan peringatan “ <i>Please Insert The Right Value</i> ”
Status	Valid

Tabel 6.96 Pengujian validasi Atur Nilai Parameter Produk Potensial

– Alternatif 2

Nama kasus uji	Atur nilai parameter produk potensial
Objek uji	Kebutuhan fungsional POSEW-F-46
Tujuan pengujian	Memastikan bahwa sistem tidak dapat mengatur nilai <i>minimum support</i> apabila isian data <i>minimum support</i> di isi dengan angka 0.
Prosedur uji	<ol style="list-style-type: none"> 1. Klik menu <i>Mining – Trends</i> 2. Klik tombol <i>Set Min. Support Value</i>

**Tabel 6.97 Pengujian validasi Atur Nilai Parameter Produk Potensial
– Alternatif 2 (Lanjutan)**

Prosedur uji	3. Mengisi nilai <i>minimum support</i> dengan angka nol. 4. Klik tombol <i>Add Parameter Value</i>
Hasil pengujian yang diharapkan	Sistem menampilkan peringatan " <i>Parameter Value Must Be Not 0 – Please Try Again !</i> "
Hasil pengujian	Sistem menampilkan peringatan " <i>Parameter Value Must Be Not 0 – Please Try Again !</i> "
Status	Valid

**Tabel 6.98 Pengujian validasi Atur Nilai Parameter Produk Potensial
– Alternatif 3**

Nama kasus uji	Atur nilai parameter produk potensial
Objek uji	Kebutuhan fungsional POSEW-F-46
Tujuan pengujian	Memastikan bahwa sistem tidak dapat mengatur nilai <i>minimum support</i> apabila isian data <i>minimum support</i> di isi dengan angka lebih dari seratus.
Prosedur uji	1. Klik menu <i>Minig - Trends</i> 2. Klik tombol <i>Set Min. Support Value</i> 3. Mengisi nilai <i>minimum support</i> dengan angka lebih dari seratus. 4. Klik tombol <i>Add Parameter Value</i>
Hasil pengujian yang diharapkan	Sistem menampilkan peringatan " <i>Parameter Value Must Be Less Than 100% - Please Try Again !</i> "
Hasil pengujian	Sistem menampilkan peringatan " <i>Parameter Value Must Be Less Than 100% - Please Try Again !</i> "
Status	Valid

6.3.47 Lihat Daftar Produk Potensial

Pengujian validasi lihat daftar produk potensial akan menguji kebutuhan fungsional POSEW-F-47. Pada pengujian ini, dilakukan proses menampilkan daftar produk potensial. Pengujian validasi lihat daftar produk potensial dipaparkan pada Tabel 6.99 dan Tabel 6.100.

Tabel 6.99 Pengujian validasi Lihat Daftar Produk Potensial

Nama kasus uji	Lihat daftar produk potensial
Objek uji	Kebutuhan fungsional POSEW-F-47

Tabel 6.100 Pengujian validasi Lihat Daftar Produk Potensial (Lanjutan)

Tujuan pengujian	Memastikan bahwa sistem menampilkan daftar produk potensial
Prosedur uji	1. Klik menu <i>Mining - Trends</i>
Hasil pengujian yang diharapkan	Sistem menampilkan daftar produk potensial dengan informasi yang disajikan adalah nama produk dan jumlah produk terjual
Hasil pengujian	Sistem menampilkan daftar produk potensial dengan informasi yang disajikan adalah nama produk dan jumlah produk terjual
Status	Valid

6.3.48 Cari Produk Potensial

Pengujian validasi cari produk potensial akan menguji kebutuhan fungsional POSEW-F-48. Pada pengujian ini, dilakukan proses pencarian produk potensial yang telah terdaftar. Pengujian validasi cari produk potensial dipaparkan pada Tabel 6.101.

Tabel 6.101 Pengujian validasi Cari Produk Potensial

Nama kasus uji	Cari produk potensial
Objek uji	Kebutuhan fungsional POSEW-F-48
Tujuan pengujian	Memastikan bahwa sistem menampilkan data produk potensial sesuai dengan masukan pada pencarian
Prosedur uji	1. Klik menu <i>Mining - Trends</i> 2. Mengisi kolom pencarian
Hasil pengujian yang diharapkan	Sistem menampilkan hasil pencarian data produk potensial berdasarkan masukan data yang diberikan.
Hasil pengujian	Sistem menampilkan hasil pencarian data produk potensial berdasarkan masukan data yang diberikan.
Status	Valid

6.3.49 Urut Produk Potensial

Pengujian validasi urut produk potensial akan menguji kebutuhan fungsional POSEW-F-49. Pada pengujian ini, dilakukan proses pengurutan produk potensial. Pengujian validasi urut produk potensial dipaparkan pada Tabel 6.102 dan Tabel 6.103 sedangkan pengujian validasi untuk proses alternatif ditampilkan pada Tabel 6.104.

Tabel 6.102 Pengujian validasi Urut Produk Potensial

Nama kasus uji	Urut produk potensial
Objek uji	Kebutuhan fungsional POSEW-F-49

Tabel 6.103 Pengujian validasi Urut Produk Potensial (Lanjutan)

Tujuan pengujian	Memastikan bahwa sistem menampilkan data produk potensial sesuai dengan urutan dari posisi terendah hingga tertinggi
Prosedur uji	1. Klik menu <i>Mining - Trends</i> 2. Klik <i>title</i> pada salah satu kolom produk potensial.
Hasil pengujian yang diharapkan	Sistem menampilkan data produk potensial dengan urutan dari posisi terendah hingga tertinggi
Hasil pengujian	Sistem menampilkan data produk potensial dengan urutan dari posisi terendah hingga tertinggi
Status	Valid

Tabel 6.104 Pengujian validasi Urut Produk Potensial – Alternatif 1

Nama kasus uji	Urut produk potensial
Objek uji	Kebutuhan fungsional POSEW-F-49
Tujuan pengujian	Memastikan bahwa sistem menampilkan data produk potensial sesuai dengan urutan dari posisi tertinggi hingga terendah
Prosedur uji	1. Klik menu <i>Mining - Trends</i> 2. Klik <i>title</i> sebanyak dua kali pada salah satu kolom produk potensial.
Hasil pengujian yang diharapkan	Sistem menampilkan data produk potensial sesuai dengan urutan dari posisi tertinggi hingga terendah
Hasil pengujian	Sistem menampilkan data produk potensial sesuai dengan urutan dari posisi tertinggi hingga terendah
Status	Valid

6.3.50 Lihat Informasi Produk Potensial

Pengujian validasi lihat produk potensial akan menguji kebutuhan fungsional POSEW-F-50. Pada pengujian ini, dilakukan proses menampilkan informasi produk potensial. Pengujian validasi lihat informasi produk potensial dipaparkan pada Tabel 6.105 dan Tabel 6.106.

Tabel 6.105 Pengujian validasi Lihat Informasi Produk Potensial

Nama kasus uji	Lihat informasi produk potensial
Objek uji	Kebutuhan fungsional POSEW-F-50
Tujuan pengujian	Memastikan bahwa sistem menampilkan menampilkan informasi produk potensial.
Prosedur uji	1. Klik menu <i>Mining - Trends</i>

Tabel 6.106 Pengujian validasi Lihat Informasi Produk Potensial (Lanjutan)

Hasil pengujian yang diharapkan	Sistem menampilkan informasi produk potensial dengan informasi yang disajikan adalah nilai <i>minimum support</i> , nilai batas minimum, dan jumlah produk potensial saat ini.
Hasil pengujian	Sistem menampilkan informasi produk potensial dengan informasi yang disajikan adalah nilai <i>minimum support</i> , nilai batas minimum, dan jumlah produk potensial saat ini.
Status	Valid

6.4 Pengujian Usabilitas

Pengujian usabilitas dilakukan untuk melakukan pengujian sistem dengan cara menguji sistem secara langsung kepada sejumlah responden untuk mengetahui tingkat kemudahan penggunaan sebuah sistem. Kinerja *usability* diukur dengan menggunakan pengujian *post study* dengan metode *System Usability Scale (SUS)* (Brooke, 2013). *SUS* merupakan sebuah pengujian *usability* yang efektif dan andal untuk digunakan pada berbagai produk dan aplikasi (Bangor, Kortum, & Miller, 2009).

6.4.1 Prosedur Pengujian Usabilitas

Prosedur pelaksanaan pengujian Usabilitas ialah melalui pertanyaan melalui kuesioner. *User* dalam hal ini ialah pihak dari Edd's Waffle. Sebelum menjawab setiap butir pertanyaan, sistem dicoba terlebih dahulu oleh *user*. Sistem dicoba oleh tiga individu yang merepresentasikan tiga segmen pengguna yang terdapat pada Edd's Waffle yaitu *Cashier*, *Supervisor*, dan *Administrator*. Setelah proses mencoba sistem selesai, ketiga segmen pengguna tersebut dipersilahkan untuk menjawab pertanyaan-pertanyaan yang telah disiapkan. Daftar pertanyaan *SUS* dapat dilihat pada Tabel 6.107 dan Tabel 6.108.

Tabel 6.107 Daftar Pertanyaan SUS

No.	Pertanyaan
1	Saya dapat mengoperasikan sistem ini dengan mudah
2	Cara melakukan aksi pada sistem cukup sulit
3	Informasi yang diberikan sangat dibutuhkan <i>restaurant</i>
4	Saya merasa perlu pendampingan dari orang yang paham untuk dapat menggunakan sistem ini
5	Bahasa dan instruksi yang diberikan mudah dipahami

Tabel 6.108 Daftar Pertanyaan SUS (Lanjutan)

6	Saya merasa huruf pada aplikasi ini terlalu kecil
7	Saya suka dengan warna yang digunakan
8	Saya merasa tombol terlalu berdekatan sehingga akan sering salah menekan tombol
9	Jika saya akan menjalankan bisnis <i>restaurant</i> saya, saya cenderung akan menggunakan sistem ini
10	Saya tidak akan menggunakan sistem ini

6.4.2 Analisis dan Hasil Pengujian Usabilitas

Setelah proses prosedur pengujian usabilitas dilakukan, maka akan dilakukan analisa dari hasil responden yang didapatkan. Respon dalam skala *likert* atas setiap pertanyaan yang tersedia dapat dilihat pada Tabel 6.109.

Tabel 6.109 Hasil Kuisisioner Skor tiap pertanyaan SUS

Nama			
No	Gledies	Abdullah	Egy
1	5	5	5
2	2	2	2
3	5	5	5
4	3	2	2
5	4	4	4
6	2	2	2
7	4	5	5
8	2	1	1
9	5	5	5
10	1	1	1

Setelah setiap skala *likert* di dapat, akan dilakukan proses konversi skor untuk mendapatkan nilai akhir dari pengujian usabilitas dengan menggunakan metode SUS. Seperti yang telah di jelaskan sebelumnya, penilaian dilakukan dengan pengurangan satu nilai dari respon yang diberikan *user*, untuk pertanyaan ganjil

dan lima dikurangi dengan respon yang diberikan *user* untuk pertanyaan genap. Setelah itu skor SUS didapat dengan cara mengkalikan total skor dengan 2.5 Pada skor SUS dianalisis menggunakan kisaran rating atau penerimaan. Hasil konversi skor tiap pertanyaan SUS dapat dilihat pada Tabel 6.110.

Tabel 6.110 Hasil Konversi Skor tiap pertanyaan SUS

Nama				
	Gledies	Abdullah	Egy	
No				
1	4	4	4	
2	3	3	3	
3	4	4	4	
4	2	3	3	
5	3	3	3	
6	3	3	3	
7	3	4	4	
8	3	4	4	
9	4	4	4	
10	4	4	4	
Jumlah	33	36	36	Rata-Rata
Dikali 2.5	82,5	90	90	87.5

Berdasarkan hasil yang didapatkan dari tiga individu responden, skor *likert* yang diperoleh akan dilakukan analisis Hasil dari pengujian kali ini. Usability diukur menggunakan pengujian post study berdasarkan metode System Usability Scale dengan skor akhir yang di dapat adalah 87.5. Dengan pencapaian skor ini, berarti sistem masuk kedalam kategori **acceptable** yang berarti sistem sudah sesuai dengan kebutuhan pengguna.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil analisis kebutuhan, perancangan, implementasi, dan pengujian yang telah dilakukan, maka dapat disimpulkan bahwa:

1. Pada tahap analisis kebutuhan sistem *Point of Sale* berbasis web pada Edd's Waffle untuk mendukung penjualan *restaurant*, diperoleh lima puluh kebutuhan fungsional dan satu kebutuhan non-fungsional. Kebutuhan fungsional yang telah didefinisikan, dimodelkan dalam bentuk *use case diagram* dan *use case scenario*. Selain itu, terdapat 4 segmentasi pengguna yaitu *Guest*, *Cashier*, *Supervisor*, dan *Administrator*. Hasil analisis diperoleh dari proses studi literatur dan wawancara yang di sesuaikan dengan kebutuhan *client* sehingga menghasilkan gambaran umum sistem yang meliputi transaksi penjualan, pengelolaan produk, riwayat transaksi, *monitoring* performa penjualan, prediksi pendapatan, dan *trend* penjualan.
2. Pada tahap perancangan yang telah dilakukan, diperoleh hasil dari perancangan arsitektur berupa rancangan *sequence diagram* dan *class diagram*, perancangan data yang meliputi *Entity Relationship Diagram* (ERD) dan *Physical Data Model* (PDM), perancangan komponen, dan perancangan antarmuka yang merupakan rancangan *layout* antarmuka sistem.
3. Pada tahap implementasi dihasilkan spesifikasi sistem, implementasi kode program menurut rancangan algoritme, implementasi basis data, dan implementasi antar muka menurut rancangan antarmuka.
4. Pada tahap pengujian yang telah dilakukan diperoleh bahwa pada tiga pengujian unit, satu pengujian integrasi dan lima puluh pengujian validasi menghasilkan nilai 100% valid. Selain itu, didapatkan nilai 87.5 pada pengujian usability dengan menggunakan metode *System Usability Scale* sehingga dikategorikan dalam kategori *acceptable*.

7.2 Saran

Saran yang diberikan untuk pengembangan lanjut sistem *Point of Sale* berbasis web pada Edd's Waffle untuk mendukung penjualan *restaurant* antara lain:

1. Pemanfaatan sistem berbasis *web* dapat di manfaatkan untuk mengadakan pemesanan atau reservasi secara *online*. Selain itu, *customer* juga dapat dilibatkan dengan adanya keanggotaan atau *membership* sehingga dapat memperluas penyebaran informasi untuk meningkatkan performa penjualan dari Edd's Waffle.
2. Pemanfaatan teknik *data mining* dapat digunakan lebih luas untuk mendukung optimalisasi performa *restaurant* seperti melakukan klusterisasi *level* pada transaksi dan prediksi stok barang maupun jumlah konsumen.

3. Pemanfaatan algoritme selain *Apriori* dapat dipertimbangkan mengingat tidak terdapatnya proses pembobotan produk maupun kategori produk pada algoritme *Apriori* sehingga belum dapat menghasilkan informasi yang lebih relevan dalam menghasilkan rekomendasi paket makanan bagi Edd's Waffle.
4. Pemanfaatan basis data *no-SQL* dapat dipertimbangkan mengingat pada perancangan dan implementasi basis data tidak terdapat relasi antar tabel.



DAFTAR REFERENSI

- Adams, J., 2013. *Learning Kendo UI*.
- Brooke John. 2011. Measuring Usability With The System Usability Scale (SUS). <http://www.measuringu.com/sus.php>. Diakses pada tanggal 22 Desember 2017.
- Bangor, A., Kortum, P., & Miller, J. 2009. Determining What Individual SUS Scores Mean : Adding an Adjective Rating Scale. *Journal of Usability Studies*
- Cafaro, M., Epicoco, I. dan Pulimeno, M., 2018. *Data Mining : Mining Frequent Patterns , Associations Rules , and Correlations*. [daring] *Encyclopedia of Bioinformatics and Computational Biology*. Elsevier Ltd. Tersedia pada: <<http://dx.doi.org/10.1016/B978-0-12-809633-8.20472-X>>.
- Caytiles, R. D. & Lee, Sunguk., 2014. *A Review of an MVC Framework based Software Development*. [e-journal]. Tersedia di: <<http://www.earticle.net/article.aspx?sn=233326>> [Diakses 29 Maret 2018]
- Cote, M. dan Mc Carthy, T., 2015. The Power of Point of Sale Improving Growth, Profit and Customer Service in a Retail Business. *Nat Genet*, [daring] 27(1), hal.5–6. Tersedia pada: <http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=11137984>.
- Dian, S., Permana, H., Telematika, F., Trilogi, U. dan Language, U.M., 2015. Analisa dan Perancangan Aplikasi Point Of Sale (POS) Untuk Mendukung Manajemen Hubungan Pelanggan. 2(1), hal.20–28.
- Freeman, A., 2016. *Pro ASP. NET Core MVC. Pro ASP. NET MVC 5 Platform*.
- Gillum, A. dan Rob, M.A., 2011. Issues in Information Systems. XII(2), hal.67–73.
- Harvey, M. 2013. Lakeview Nurseries Clears Plant Inventory In Fall Without Clearances. *Today's Garden Center*, 10(6), hal. 28-32.
- Holt, C. C. (2004). Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting* 20, 5-10.
- Irza, G.P. 2018. Wawancara mengenai Sistem *Point of Sale* pada Edd's Waffle. Jl. Marsan Sejahtera 3-4, Pekanbaru, Riau, Indonesia.
- Jalis, M.H., Kassim, A.R.A. dan Mohamad, N.H., 2012. Identification of Cluster Factors for the Standalone Restaurant. *Procedia - Social and Behavioral Sciences*, [daring] 65(ICIBSoS), hal.293–299. Tersedia pada: <<http://linkinghub.elsevier.com/retrieve/pii/S1877042812051105>>.
- Kabir, A. dan Han, B., 2016. An Improved Usability Evaluation Model for Point-of-Sale Systems An Improved Usability Evaluation Model for Point-of-Sale Systems.

- Lasek, A., Cercone, N. dan Saunders, J., 2016. *Smart restaurants: Survey on customer demand and sales forecasting*. [daring] *Smart Cities and Homes: Key Enabling Technologies*. Elsevier Inc. Tersedia pada: <<http://dx.doi.org/10.1016/B978-0-12-803454-5/00017-1>>.
- Lee, C.H., 2001. Coordinated stocking, clearance sales, and return policies for a supply chain. *European Journal of Operational Research*, 131(3), hal.491–513.
- Li, X. (2013). Comparison and Analysis between Holt Exponential Smoothing and Brown Exponential Smoothing Used for Freight Turnover Forecasts. *Third International Conference on Intelligent System Design and Engineering Applications*.
- Polanz, J. 2011. Your POS Shopping List. *Today's Garden Center*, 8(5), 20.'
- Pressman, R. S., 2001. *Software Engineering*. 5th ed. New York: McGraw-Hill Companies, Inc.
- Pressman, R. S., 2010. *Software Engineering*. 7th ed. New York: McGraw-Hill Companies, Inc.
- Santoso, I., 2016. Peran Kualitas Produk dan Layanan, Harga dan Atmosfer Rumah Makan Cepat Saji terhadap Keputusan Pembelian dan Kepuasan Konsumen. *Jurnal Manajemen Teknologi*, [daring] 15(1), hal.94–109. Tersedia pada: <http://journal.sbm.itb.ac.id/index.php/mantek/article/view/1724/pdf_10%0A>.
- Saxena, A. dan Gaur, N.K., 2015. Frequent Item Set Based Recommendation using Apriori. 4(5).
- Sharma, K. S. D. & Singh, A., 2016. Software Development Life Cycle. *International Journal of Modern Engineering and Research Technology*, p. 5.
- Sommerville, Ian., 2011. *Software engineering*. 9th ed. London: Addison-Wesley.
- Spillner, Andreas, Tilo Linz, and Hans Schaefer. 2014. *Software Testing Foundations - A Study Guide for the Certified Tester Exam*. Edited by Dr. Michael Barabas. *Heidelberg, Germany*. Vol. 16. Santa Barbara, CA.
- Taylor, J.W., 2011. Multi-item sales forecasting with total and split exponential smoothing. *Journal of the Operational Research Society*, 62(3), hal.555–563.
- Whitten, Jeffrey L, and Lonnie D. Bentley. 2007. *Systems Analysis and Design Methods*. McGraw-Hill.
- Zhu, J., 2013. POS Data and Your Demand Forecast. *Procedia Computer Science*, [daring] 17, hal.8–13. Tersedia pada: <<http://dx.doi.org/10.1016/j.procs.2013.05.002>>.