

**PENGEMBANGAN WEBSITE E-COMMERCE MENGGUNAKAN
PAYMENT GATEWAY MIDTRANS (STUDI KASUS : BUTIK
KEBAYA IFFAH)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Velia Wilda Ifanah

NIM: 175150707111007



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



PENGESAHAN

PENGEMBANGAN *WEBSITE E-COMMERCE* MENGGUNAKAN *PAYMENT GATEWAY*
MIDTRANS (STUDI KASUS : BUTIK KEBAYA IFFAH)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Velia Wilda Ifanah
NIM: 175150707111007

Skripsi ini telah diuji dan dinyatakan lulus pada
5 Juli 2021

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing 1



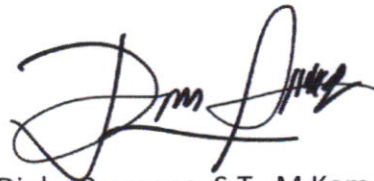
Digitally signed by
Widhy H_N_Putra
DN: C=ID, OU=FILKOM,
O=UB, CN=Widhy H_N_
Putra,
E=widhy@ub.ac.id
Reason: I am approving
this document
Date: 2021-07-17 10:18:12

Widhy Hayuhardhika Nugraha Putra,

S.Kom., M.Kom

NIK: 201712 870409 2 001

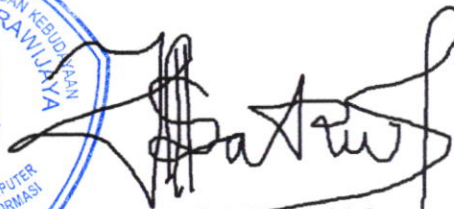
Dosen Pembimbing 2



Djoko Pramono, S.T., M.Kom

NIP: 19780108 200501 1 002

Mengetahui
Ketua Jurusan Sistem Informasi



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

NIP: 19830922 201212 1 003

PERNYATAAN ORISINALITAS

PERNYATAAN ORISINALITAS

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

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat 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, 25 Juni 2021



Velia Wilda Ifanah

NIM: 175150707111007

ABSTRAK

Velia Wilda Ifanah, Pengembangan Website E-commerce Menggunakan Payment Gateway Midtrans (Studi Kasus : Butik Kebaya Iffah)

Pembimbing: Widhy Hayuhardhika Nugraha Putra, S.Kom., M.Kom dan Djoko Pramono, S.T., M.Kom

Butik Kebaya Iffah merupakan brand yang bergerak pada bisnis *fashion* yang menjual berbagai macam kebaya hasil produksi sendiri. Proses bisnis Butik Kebaya Iffah masih dilakukan dengan dua metode yaitu *offline* datang langsung ke toko dan *online* melalui *media social* untuk mempromosikan produk. Dari kedua metode tersebut didapati permasalahan pada metode *online* dirasa pemilik kurang efektif dan aman bagi pelanggan untuk melakukan transaksi serta belum memberikan dampak promosi yang besar, juga untuk mekanisme pembelian produk custom ukuran, pelanggan harus datang ke toko untuk dilakukan pengukuran. Dalam masalah pembayaran, banyak permintaan pilihan metode pembayaran, juga bagi admin masih mengecek status pembayaran secara manual dan tidak *real-time*. Sehingga dari masalah yang ada, dikembangkan *Website E-commerce* Butik Kebaya Iffah yang diintegrasikan menggunakan *payment gateway* midtrans. Pengembangan sistem dilakukan dengan menggunakan metode waterfall. Hasil dari penelitian ini didapatkan 18 fungsionalitas pada analisis kebutuhan yang akan dibuat dengan kebutuhan dan aktor yang berbeda. Fitur yang menjadi pembeda pada *e-commerce* ini adalah dapat dilakukan *custom order* kebaya Sistem pada penelitian ini dibangun menggunakan *framework codeigniter* dan *bootstrap* sebagai bantuan pengembangan sistem. Setelah dirancang dan diimplmentasikan dilakukan pengujian dengan metode *black box testing* dengan hasil 100 % valid, yang berarti fungsi-fungsi dari sitem berjalan dengan baik, *compatibility testing* yang tidak didapati *minor* maupun *major issues* yang berarti sistem dapat berjalan pada lingkungan yang berbeda, dan *user acceptance testing* (UAT) dengan memperoleh hasil persentase sebesar 90,3% yang berarti sistem dibuat sesuai dengan kebutuhan user dan diterima dengan hasil sangat setuju.

Kata kunci: butik kebaya iffah, *e-commerce*, *payment gateway*, *Midtrans*, *codeigniter*, *user acceptance testing*

ABSTRACT

Velia Wilda Ifanah, Development of an E-commerce Website Using Midtrans Payment Gateway (Case Study: Kebaya Iffah Boutique)

Supervisors: Widhy Hayuhardhika Nugraha Putra, S.Kom., M.Kom dan Djoko Pramono, S.T., M.Kom

Kebaya Iffah boutique is a brand that is engaged in the fashion business that sells various kinds of kebaya produced by themselves. The Kebaya Iffah Boutique business process is still carried out using two methods, namely offline coming directly to the store and online through social media to promote products. From the two methods, it was found that the problem with the online method was that the owner felt it was less effective and safe for customers to make transactions and had not had a big promotional impact, also for the mechanism for purchasing custom-sized products, customers had to come to the store to take measurements. In terms of payment, there are many requests for payment method choices, and admins are still checking payment status manually and not in real-time. So from the existing problems, the Kebaya Iffah Boutique E-commerce Website was developed which was integrated using the midtrans payment gateway. System development is carried out using the waterfall method. The results of this study obtained 18 functionalities in the needs analysis that will be made with different needs and actors. The distinguishing feature of this e-commerce is that custom orders for kebaya can be made. The system in this study was built using a codeigniter and bootstrap framework as system development assistance. After being designed and implemented, testing is carried out using the black box testing method with 100% valid results, which means that the functions of the system are running well, compatibility testing is not found to be minor or major issues, which means the system can run in a different environment, and user acceptance testing (UAT) by obtaining a percentage result of 90.3% which means the system is made according to user needs and is accepted with the results strongly agree.

Keywords: *kebaya iffah boutique, e-commerce, payment gateway, Midtrans, codeigniter, user acceptance testing*

DAFTAR ISI

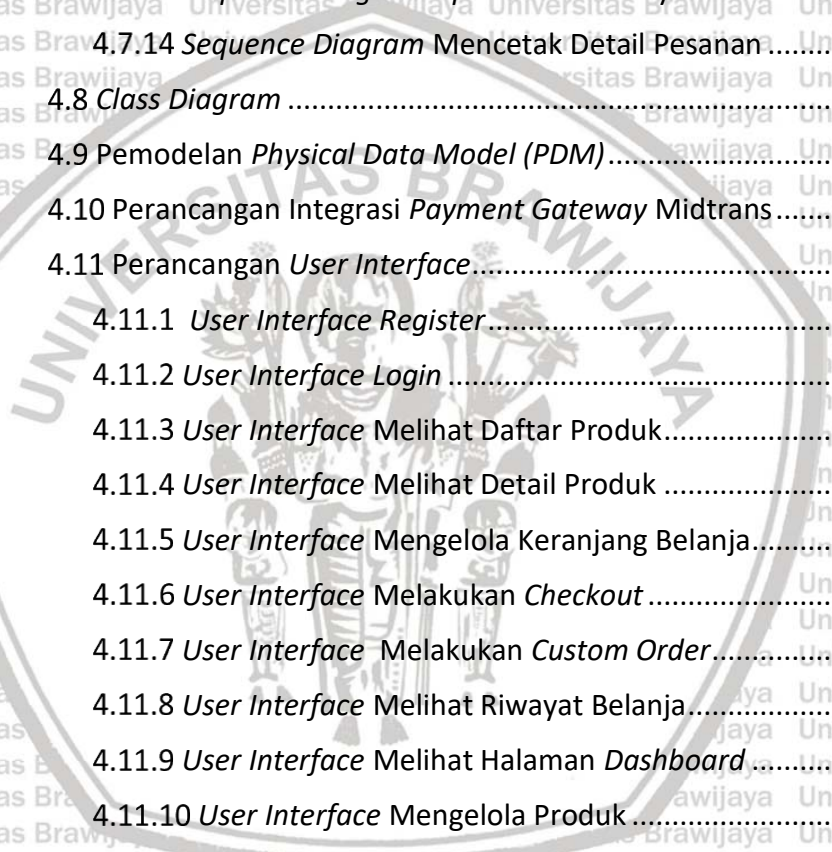
PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xvi
DAFTAR LAMPIRAN	xix
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Manfaat	3
1.5 Batasan Masalah	4
1.6 Sistematika Pembahasan	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Butik Kebaya Iffah	6
2.2 Kajian Pustaka	7
2.3 Sistem Informasi	8
2.3.1 <i>System Development Life Cycle (SDLC)</i>	8
2.4 Website	9
2.5 <i>E-commerce</i>	10
2.5.2 Klasifikasi <i>E-commerce</i>	11
2.6 Bahasa Pemrograman	11
2.6.1 Hypertext Markup Language (HTML)	11
2.6.2 Hypertext Preprocessor (PHP)	12
2.6.3 CSS (<i>Cascading Style Sheet</i>)	12
2.6.4 <i>Javascript</i>	12
2.7 Proses Bisnis	13
2.7.1 <i>Business Process Modeling Notation (BPMN)</i>	13



2.8 UML (Unified Modelling Language)	14
2.8.1 Use case diagram	14
2.8.2 Class diagram	15
2.8.3 Activity diagram	16
2.8.4 Sequence diagram	17
2.9 MySQL	18
2.10 Software Pendukung	18
2.10.1 Visual Studio Code	18
2.10.2 Framework	19
2.10.3 Bootstrap	20
2.11 Application Programming Interface (API)	20
2.12 Payment Gateway	20
2.12.1 Midtrans	21
2.12.2 Mendapatkan <i>Snap Token</i> Transaksi di <i>Backend</i>	21
2.12.3 Menampilkan Halaman Pembayaran <i>Snap</i> di <i>Frontend</i>	22
2.12.4 Melakukan Uji Pembayaran	23
2.12.5 Menangani Setelah Pembayaran	23
2.13 Pengujian Perangkat Lunak	23
2.13.1 Black Box	23
2.13.2 Compatibility Testing	24
2.13.3 User Acceptance Testing	24
BAB 3 METODOLOGI	27
3.1 Tipe Penelitian	27
3.2 Strategi Penelitian	27
3.2.1 Studi Kepustakaan	28
3.2.2 Analisis Kebutuhan	29
3.2.3 Perancangan Sistem	29
3.2.4 Implementasi	29
3.2.5 Pengujian	30
3.2.6 Penutup	30
BAB 4 ANALISIS KEBUTUHAN DAN PERANCANGAN SISTEM	31
4.1 Deskripsi Umum Sistem	31



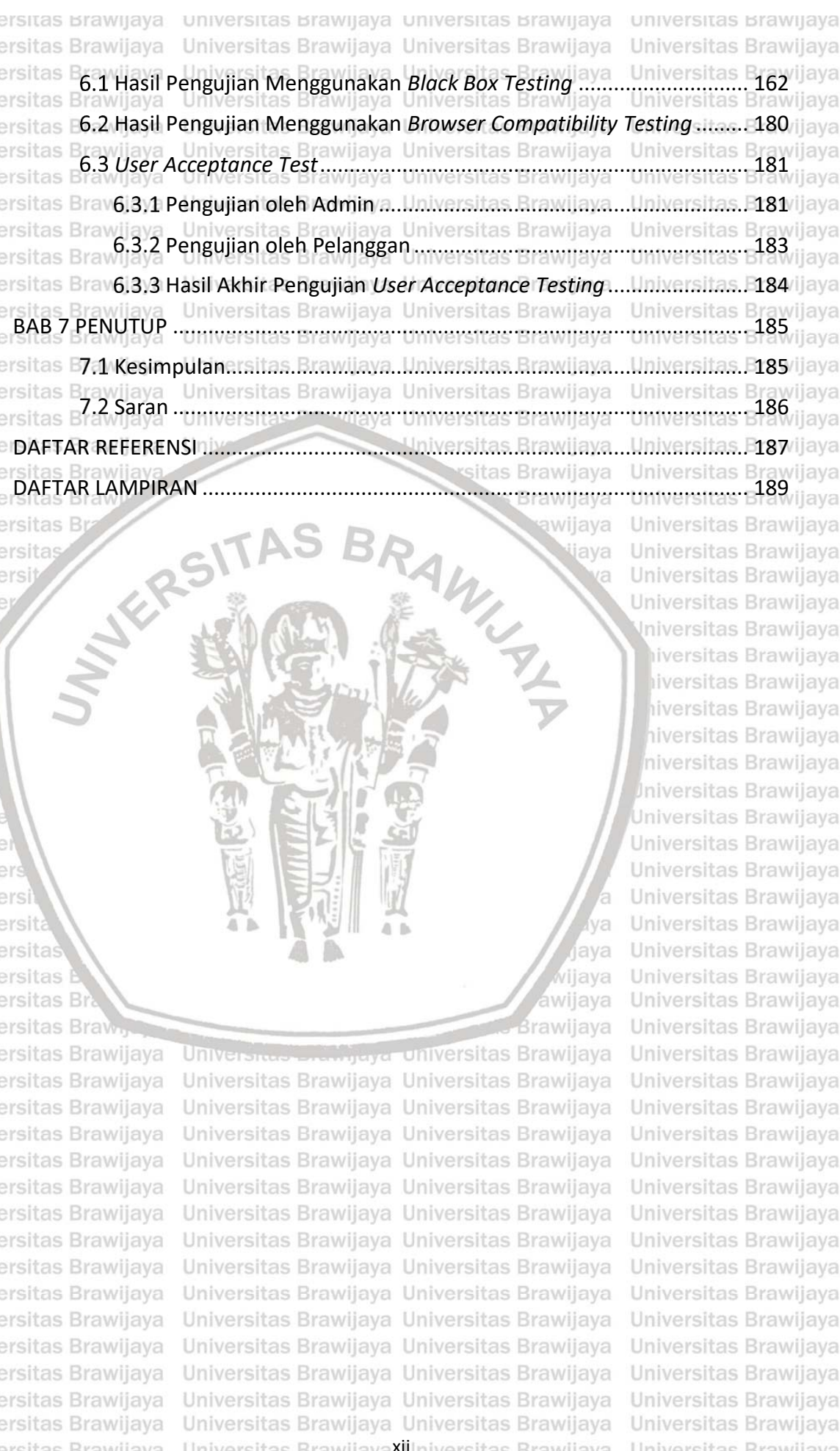
4.2	Arsitektur Website E-commerce Butik Kebaya Iffah	32
4.3	Pemodelan Proses Bisnis	32
4.1.1	Pemodelan Proses Bisnis As-Is.....	33
4.1.2	Pemodelan Proses Bisnis <i>To-Be</i>	35
4.4	Analisis Kebutuhan	37
4.4.1	Identifikasi Aktor	37
4.4.2	Analisis Kebutuhan Fungsional	38
4.4.3	Analisis Kebutuhan Non Fungsional.....	42
4.4.4	<i>Use Case Diagram</i>	42
4.5	Skenario <i>Use Case</i>	43
4.6	<i>Activity Diagram</i>	58
4.6.1	<i>Activity Diagram Register</i>	58
4.6.2	<i>Activity Diagram Login</i>	59
4.6.3	<i>Activity Diagram Logout</i>	61
4.6.4	<i>Activity Diagram</i> Melihat Daftar Produk.....	62
4.6.5	<i>Activity Diagram</i> Mengelola Keranjang Belanja	62
4.6.6	<i>Activity Diagram</i> Melakukan <i>Custom Order</i>	63
4.6.7	<i>Activity Diagram</i> Melakukan <i>Checkout</i>	64
4.6.8	<i>Activity Diagram</i> Melakukan Pembayaran	65
4.6.9	<i>Activity Diagram</i> Melihat Riwayat Transaksi	66
4.6.10	<i>Activity Diagram Dashboard</i>	67
4.6.11	<i>Activity Diagram</i> Mengelola Produk.....	67
4.6.12	<i>Activity Diagram</i> Melihat Data Pesanan.....	68
4.6.13	<i>Activity Diagram</i> Melihat Data <i>Custom Order</i>	69
4.6.14	<i>Activity Diagram Update</i> Status Bayar.....	70
4.6.15	<i>Activity Diagram</i> Mengelola Admin	70
4.6.16	<i>Activity Diagram</i> Mencetak Detail Pesanan	71
4.7	<i>Sequence Diagram</i>	72
4.7.1	<i>Sequence Diagram Register</i>	72
4.7.2	<i>Sequence Diagram Login</i>	72
4.7.3	<i>Sequence Diagram logout</i>	74
4.7.4	<i>Sequence Diagram</i> Melihat Daftar Produk	75



4.7.5 Sequence Diagram Mengelola Keranjang Belanja	75
4.7.6 Sequence Diagram Melakukan Custom Order	76
4.7.7 Sequence Diagram Melakukan Checkout	77
4.7.8 Sequence Diagram Melakukan Pembayaran	78
4.7.9 Sequence Diagram Melihat Dashboard	79
4.7.10 Sequence Diagram Mengelola Produk	80
4.7.11 Sequence Diagram Melihat Data Pesanan	81
4.7.12 Sequence Diagram Melihat Data Custom Order	81
4.7.13 Sequence Diagram Update Status Bayar	82
4.7.14 Sequence Diagram Mencetak Detail Pesanan	83
4.8 Class Diagram	83
4.9 Pemodelan Physical Data Model (PDM)	85
4.10 Perancangan Integrasi Payment Gateway Midtrans	86
4.11 Perancangan User Interface	86
4.11.1 User Interface Register	87
4.11.2 User Interface Login	88
4.11.3 User Interface Melihat Daftar Produk	89
4.11.4 User Interface Melihat Detail Produk	90
4.11.5 User Interface Mengelola Keranjang Belanja	90
4.11.6 User Interface Melakukan Checkout	91
4.11.7 User Interface Melakukan Custom Order	92
4.11.8 User Interface Melihat Riwayat Belanja	93
4.11.9 User Interface Melihat Halaman Dashboard	93
4.11.10 User Interface Mengelola Produk	94
4.11.11 User Interface Melihat Data Pesanan	95
4.11.12 User Interface Melihat Data Custom Order	96
4.11.13 User Interface Pembayaran	97
BAB 5 IMPLEMENTASI	98
5.1 Spesifikasi Lingkungan Implementasi	98
5.1.1 Spesifikasi Perangkat Keras	98
5.1.2 Spesifikasi Perangkat Lunak	98
5.2 Konfigurasi API sandbox Payment Gateway Midtrans	99



5.3 Implementasi <i>User Interface</i>	99
5.3.1 <i>User Interface Register</i>	99
5.3.2 <i>User Interface Login Pelanggan</i>	100
5.3.3 <i>User Interface Melihat Daftar Produk</i>	101
5.3.4 <i>User Interface Melihat Detail Produk</i>	102
5.3.5 <i>User Interface Keranjang Belanja</i>	102
5.3.6 <i>User Interface Custom Order</i>	103
5.3.7 <i>User Interface Checkout</i>	104
5.3.8 <i>User Interface Pembayaran</i>	105
5.3.9 <i>User Interface Login Admin</i>	106
5.3.10 <i>User Interface Halaman Dashboard Admin</i>	106
5.3.11 <i>User Interface Halaman Mengelola Produk</i>	107
5.3.12 <i>User Interface Halaman Melihat Data Pesanan dan Update Status</i>	108
5.3.13 <i>User Interface Halaman Melihat Data Custom Order dan Update Status</i>	109
5.4 Implementasi <i>Database</i>	110
5.4.1 Implementasi Table Pelanggan	110
5.4.2 Implementasi Table Admin	111
5.4.3 Implementasi Tabel Produk	111
5.4.4 Implementasi Tabel Header Transaksi Pesanan	112
5.4.5 Implementasi Tabel Transaksi Pesanan	112
5.4.6 Implementasi Tabel Header Transaksi Custom.....	113
5.4.7 Implementasi Tabel Custom	114
5.4.8 Implementasi Tabel Kategori	114
5.4.9 Implementasi Tabel Gambar.....	115
5.5 Implementasi Kode Program	115
5.5.1 Kode Program Pengaturan Umum Midtrans	115
5.5.2 Kode Program Produk.....	115
5.5.3 Kode Program Custom	129
5.5.4 Kode Program Checkout	140
5.5.5 Kode Program Pembayaran	149
BAB 6 PENGUJIAN	162



6.1 Hasil Pengujian Menggunakan <i>Black Box Testing</i>	162
6.2 Hasil Pengujian Menggunakan <i>Browser Compatibility Testing</i>	180
6.3 <i>User Acceptance Test</i>	181
6.3.1 Pengujian oleh Admin	181
6.3.2 Pengujian oleh Pelanggan	183
6.3.3 Hasil Akhir Pengujian <i>User Acceptance Testing</i>	184
BAB 7 PENUTUP	185
7.1 Kesimpulan	185
7.2 Saran	186
DAFTAR REFERENSI	187
DAFTAR LAMPIRAN	189

DAFTAR TABEL

Tabel 2.1 Komponen-komponen pada <i>use case diagram</i>	14
Tabel 2.2 Komponen-komponen pada <i>class diagram</i>	15
Tabel 2.3 Komponen-komponen pada <i>activity diagram</i>	16
Tabel 2.4 Komponen-komponen pada <i>sequence diagram</i>	17
Tabel 4. 1 Identifikasi Aktor	37
Tabel 4. 2 Kebutuhan Fungsional Pelanggan	39
Tabel 4. 3 Kebutuhan Fungsional Admin	40
Tabel 4. 4 Kebutuhan Fungsional Midtrans	41
Tabel 4. 5 Kebutuhan Non Fungsional	42
Tabel 4. 6 Skenario <i>Use Case</i> Melakukan Registrasi	43
Tabel 4. 7 Skenario <i>Use Case</i> Login Pelanggan	44
Tabel 4. 8 Skenario <i>Use Case</i> Logout	45
Tabel 4. 9 Skenario <i>Use Case</i> melihat daftar produk	46
Tabel 4. 10 Skenario <i>Use Case</i> Mengelola keranjang belanja	46
Tabel 4. 11 Skenario <i>Use Case</i> Melakukan <i>Custom Order</i>	48
Tabel 4. 12 Skenario <i>Use Case</i> Melakukan <i>checkout</i>	49
Tabel 4. 13 Skenario <i>Use Case</i> Melakukan Pembayaran	49
Tabel 4. 14 Skenario <i>Use Case</i> Melihat Riwayat transaksi	50
Tabel 4. 15 Skenario <i>Use Case</i> Login Admin	51
Tabel 4. 16 Skenario <i>Use Case</i> Logout Admin	52
Tabel 4. 17 Skenario <i>Use Case</i> Melihat <i>Dashboard</i>	53
Tabel 4. 18 Skenario <i>Use Case</i> Mengelola produk	53
Tabel 4. 19 Skenario <i>Use Case</i> Melihat Data Pesanan	55
Tabel 4. 20 Skenario <i>Use Case</i> Melihat Data <i>Custom Order</i>	55
Tabel 4. 21 Skenario <i>Use Case</i> Update Status Bayar	56
Tabel 4. 22 Skenario <i>Use Case</i> Mengelola Data admin	57
Tabel 4. 23 Skenario <i>Use Case</i> Mencetak Detail Pesanan	58
Tabel 5. 1 Spesifikasi Lingkungan Perangkat Keras	98
Tabel 5. 2 Spesifikasi Lingkungan Perangkat Lunak	98
Tabel 5. 3 Implementasi Tabel Pelanggan	110
Tabel 5. 4 Implementasi Tabel Admin	111



Tabel 5. 5 Implementasi Tabel Produk.....	111
Tabel 5. 6 Implementasi Tabel Header Transaksi.....	112
Tabel 5. 7 Implmentasi Tabel Transaksi.....	113
Tabel 5. 8 Implementasi Tabel Header Transaksi Custom.....	113
Tabel 5. 9 Implementasi Tabel Custom.....	114
Tabel 5. 10 Implementasi Tabel Kategori.....	114
Tabel 5. 11 Implementasi Tabel Gambar.....	115
Tabel 5. 12 Kode Program Pengaturan Umum Midtrans.....	115
Tabel 5. 13 Kode Program Controller Produk.....	116
Tabel 5. 14 Kode Program Controller Mengelola Produk.....	120
Tabel 5. 15 Kode Program Model Produk.....	125
Tabel 5. 16 Kode Program Controller Custom.....	130
Tabel 5. 17 Kode program Controller Data Custom Order.....	135
Tabel 5. 18 Kode Program Model Custom.....	136
Tabel 5. 19 Kode Program Model Header Transaksi Custom.....	138
Tabel 5. 20 Kode Program Controller Belanja.....	140
Tabel 5. 21 Kode Program Controller Data Pesanan.....	144
Tabel 5. 22 Kode Program Model Header Transaksi.....	146
Tabel 5. 23 Kode Program Model transaksi.....	147
Tabel 5. 24 Kode Program Controller Pembayaran.....	149
Tabel 5. 25 Kode Program <i>View Checkout</i>	155
Tabel 6. 1 Pengujian pada Proses Melakukan Registrasi.....	162
Tabel 6. 2 Pengujian pada Proses Login.....	164
Tabel 6. 3 Pengujian pada Proses Logout.....	165
Tabel 6. 4 Pengujian pada Proses Melihat daftar produk.....	165
Tabel 6. 5 Pengujian pada Proses Mengelola Keranjang Belanja.....	166
Tabel 6. 6 Pengujian pada Proses Melakukan <i>Custom Order</i>	167
Tabel 6. 7 Pengujian pada Proses Melakukan <i>Checkout</i>	169
Tabel 6. 8 Pengujian pada Proses Melakukan Pembayaran.....	171
Tabel 6. 9 Pengujian pada Proses Melihat Riwayat Transaksi.....	171
Tabel 6. 10 Pengujian pada Proses <i>Login Admin</i>	172
Tabel 6. 11 Pengujian pada Proses <i>Logout Admin</i>	173

Tabel 6. 12 Pengujian pada Proses Melihat <i>Dashboard</i>	173
Tabel 6. 13 Pengujian pada Proses Mengelola Produk	174
Tabel 6. 14 Pengujian pada Proses Melihat Data Pesanan	176
Tabel 6. 15 Pengujian pada Proses Melihat Data <i>Custom Order</i>	176
Tabel 6. 16 Pengujian pada Proses Update Status Bayar	176
Tabel 6. 17 Pengujian pada Proses Mengelola Admin	177
Tabel 6. 18 Pengujian pada Proses Mencetak Detail Pesanan	179
Tabel 6. 19 Browser Compatibility Testing	180
Tabel 6. 20 Hasil Kuesioner UAT <i>User Admin</i>	181
Tabel 6. 21 Tabel Perhitungan Skala Likert <i>User Admin</i>	182
Tabel 6. 22 Hasil Kuesioner UAT <i>User Pelanggan</i>	183
Tabel 6. 23 Tabel Perhitungan Skala Likert <i>User Pelanggan</i>	183



DAFTAR GAMBAR

Gambar 2. 1 Logo Butik Kebaya Iffah	6
Gambar 2. 2 Tahapan Metode <i>SDLC Waterfall</i>	9
Gambar 2. 3 Gambaran Arsitektur ModelViewController (MVC)	19
Gambar 2. 4 Contoh <i>Request SNAP Token</i> Midtrans	22
Gambar 2. 5 Menampilkan <i>Snap</i> Halaman Pembayaran	22
Gambar 4. 1 Arsitektur Website E-commerce Butik Kebaya Iffah	32
Gambar 4. 3 Proses Bisnis Saat Ini Pada Butik Kebaya Iffah Pemesanan Melalui Whatsapp	34
Gambar 4. 2 Proses Bisnis Saat Ini Pada Butik Kebaya Iffah Datang Langsung ke Toko	34
Gambar 4. 4 Proses Bisnis Usulan Pembelian Produk Pada Butik Kebaya Iffah ...	36
Gambar 4. 5 Penomoran Identifikasi Aktor	37
Gambar 4. 6 Penomoran Kebutuhan Fungsional	38
Gambar 4. 7 Penomoran Kebutuhan Non-Fungsional	42
Gambar 4. 8 Penomoran Diagram <i>Use Case</i>	42
Gambar 4. 9 <i>Use Case Diagram E-commerce</i> Butik Kebaya Iffah	43
Gambar 4. 10 <i>Activity Diagram Register</i>	59
Gambar 4. 11 <i>Activity Diagram Login</i> Pelanggan	60
Gambar 4. 12 <i>Activity Diagram Login</i> Admin	61
Gambar 4. 13 <i>Activity Diagram Logout</i> pada pelanggan dan admin	61
Gambar 4. 14 <i>Activity Diagram</i> Melihat Daftar Produk	62
Gambar 4. 15 <i>Activity Diagram</i> Mengelola Keranjang Belanja	63
Gambar 4. 16 <i>Activity Diagram</i> Melakukan <i>Custom Order</i>	64
Gambar 4. 17 <i>Activity Diagram</i> Melakukan <i>Checkout</i>	65
Gambar 4. 18 <i>Activity Diagram</i> Melakukan Pembayaran	66
Gambar 4. 19 <i>Activity Diagram</i> Melihat Riwayat Transaksi	66
Gambar 4. 20 <i>Activity Diagram Dashboard</i> Admin	67
Gambar 4. 21 <i>Activity Diagram</i> Mengelola Produk	68
Gambar 4. 22 <i>Activity Diagram</i> Melihat Data Pesanan	69
Gambar 4. 23 <i>Activity Diagram</i> Melihat Data <i>Custom Order</i>	69
Gambar 4. 24 <i>Activity Diagram Update</i> Status Bayar	70

Gambar 4. 25 <i>Activity Diagram</i> Mengelola Admin	71
Gambar 4. 26 <i>Activity Diagram</i> Mencetak Detail Pesanan.....	71
Gambar 4. 27 <i>Sequence Diagram</i> Register	72
Gambar 4. 28 <i>Sequence Diagram</i> Login Pelanggan	73
Gambar 4. 29 <i>Sequence Diagram</i> Login Admin	73
Gambar 4. 30 <i>Sequence Diagram</i> Logout Pelanggan dan Admin	74
Gambar 4. 31 <i>Sequence Diagram</i> Logout Admin.....	74
Gambar 4. 32 <i>Sequence Diagram</i> melihat daftar produk.....	75
Gambar 4. 33 <i>Sequence Diagram</i> Mengelola Keranjang Belanja	76
Gambar 4. 34 <i>Sequence Diagram</i> Melakukan <i>Custom Order</i>	77
Gambar 4. 35 <i>Sequence Diagram</i> Melakukan <i>Checkout</i>	78
Gambar 4. 36 <i>Sequence Diagram</i> Pembayaran	79
Gambar 4. 37 <i>Sequence Diagram</i> Melihat <i>Dashboard</i>	79
Gambar 4. 38 <i>Sequence Diagram</i> Mengelola Produk.....	80
Gambar 4. 39 <i>Sequence Diagram</i> Melihat Data Pesanan.....	81
Gambar 4. 40 <i>Sequence Diagram</i> Melihat Data <i>Custom Order</i>	82
Gambar 4. 41 <i>Sequence Diagram</i> Update Status Bayar.....	82
Gambar 4. 42 <i>Sequence Diagram</i> Mencetak Detail Pesanan	83
Gambar 4. 43 <i>Class Diagram</i> Website <i>E-commerce</i> Butik Kebaya Iffah	84
Gambar 4. 44 <i>Physical Data Model</i> Website <i>E-commerce</i> Butik Kebaya Iffah	85
Gambar 4. 45 Perancangan Integrasi <i>Payment Gateway</i> Midtrans	86
Gambar 4. 46 Perancangan <i>User Interface</i> Registrasi Pelanggan.....	87
Gambar 4. 47 Perancangan <i>User Interface</i> Login Pelanggan.....	88
Gambar 4. 48 Perancangan <i>User Interface</i> Login Admin.....	88
Gambar 4. 49 Perancangan <i>User Interface</i> Melihat Daftar Produk.....	89
Gambar 4. 50 Perancangan <i>User Interface</i> Melihat Detail Produk.....	90
Gambar 4. 51 Perancangan <i>User Interface</i> Keranjang Belanja	90
Gambar 4. 52 Perancangan <i>User Interface</i> Melakukan <i>Checkout</i>	91
Gambar 4. 53 Perancangan <i>User Interface</i> Melakukan <i>Custom Order</i>	92
Gambar 4. 54 Perancangan <i>User Interface</i> Melihat Riwayat Belanja.....	93
Gambar 4. 55 Perancangan <i>User Interface</i> Melihat <i>Dashboard</i>	93
Gambar 4. 56 Perancangan <i>User Interface</i> Mengelola Produk	94



Gambar 4. 57 Perancangan <i>User Interface</i> Mengelola Produk (Form Tambah) ..	94
Gambar 4. 58 Perancangan <i>User Interface</i> Melihat Data Pesanan dan Cek Status	95
Gambar 4. 59 Perancangan <i>User Interface</i> Melihat Data <i>Custom Order</i> dan Cek Status.....	96
Gambar 4. 60 Perancangan <i>User Interface</i> Melihat Riwayat Belanja.....	96
Gambar 4. 61 Perancangan <i>User Interface</i> Pembayaran Midtrans	97
Gambar 5. 1 Konfigurasi Server-Key	99
Gambar 5. 2 <i>Frontend Reuirement Client-Key</i>	99
Gambar 5. 3 Implementasi <i>User Interface</i> Halaman Registrasi	100
Gambar 5. 4 Implementasi <i>User Interface</i> Halaman <i>Login</i> Pelanggan	101
Gambar 5. 5 Implementasi <i>User Interface</i> Halaman Melihat Daftar Produk	101
Gambar 5. 6 Implementasi <i>User Interface</i> Halaman Melihat Detail Produk	102
Gambar 5. 7 Implementasi <i>User Interface</i> Halaman Keranjang Belanja	103
Gambar 5. 8 Implementasi <i>User Interface</i> Halaman <i>Custom Order</i>	103
Gambar 5. 9 Implementasi <i>User Interface</i> Halaman <i>Checkout</i>	104
Gambar 5. 10 Implementasi <i>User Interface</i> Halaman Pembayaran Midtrans....	106
Gambar 5. 11 Implementasi <i>User Interface</i> Halaman <i>Login</i> Admin	106
Gambar 5. 12 Implementasi <i>User Interface</i> Halaman <i>Dashboard</i> Admin	107
Gambar 5. 13 Implementasi <i>User Interface</i> Halaman Mengelola Produk.....	107
Gambar 5. 14 Implementasi <i>User Interface</i> Halaman Mengelola Produk (Form Tambah)	108
Gambar 5. 15 Implementasi <i>User Interface</i> Halaman Melihat Data Pesanan	108
Gambar 5. 16 Implementasi <i>User Interface</i> Halaman Melihat <i>Data Custom Order</i>	109
Gambar 5. 17 Implementasi <i>Physical Data Model Database Website E-commerce</i> Butik Kebaya Iffah	110
Gambar 6. 1 Hasil <i>Compatibility Testing</i>	180

DAFTAR LAMPIRAN

Lampiran 1 Transkrip Wawancara	189
Lampiran 2 Transkrip Wawancara Lembar Ke-2	190
Lampiran 3 Transkrip Wawancara Lembar Ke-3	191
Lampiran 4 Hasil Pengujian Black-Box Testing oleh <i>Owner</i> Lembar Ke-1.....	192
Lampiran 5 Hasil Pengujian Black-Box Testing oleh <i>Owner</i> Lembar Ke-2.....	193
Lampiran 6 Hasil Pengujian Black-Box Testing oleh <i>Owner</i> Lembar Ke-3.....	194
Lampiran 7 Hasil Pengujian Black-Box Testing oleh <i>Owner</i> Lembar Ke-4.....	195
Lampiran 8 Hasil Pengujian Black-Box Testing oleh <i>Owner</i> Lembar Ke-5.....	196
Lampiran 9 Hasil Pengujian Black-Box Testing oleh <i>Owner</i> Lembar Ke-6.....	197
Lampiran 10 Hasil Pengujian Black-Box Testing oleh Pelanggan Lembar Ke-1..	198
Lampiran 11 Hasil Pengujian Black-Box Testing oleh Pelanggan Lembar Ke-2..	199
Lampiran 12 Hasil Pengujian Black-Box Testing oleh Pelanggan Lembar Ke-3..	200
Lampiran 13 Hasil Pengujian Black-Box Testing oleh Pelanggan Lembar Ke-4..	201
Lampiran 14 Hasil Pengujian UAT Responden Admin 1.....	202
Lampiran 15 Hasil Pengujian UAT Responden Pelanggan 1.....	203
Lampiran 16 Hasil Pengujian UAT Responden Pelanggan 2.....	204
Lampiran 17 Hasil Pengujian UAT Responden Pelanggan 3.....	205

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Segala hal di era modern dan serba digital seperti sekarang ini, tidak dipungkiri memberikan dampak dalam kehidupan bermasyarakat, apalagi dengan adanya peran teknologi dan internet. Terutama dalam bidang bisnis, peran teknologi informasi menjadi sebuah hal wajib yang harus diterapkan pada suatu bisnis. Salah satu implementasi teknologi informasi online adalah dengan menggunakan Electronic Commerce (e-commerce) yang dapat digunakan untuk bisnis. E-commerce adalah aktivitas belanja online yang menggunakan internet, dan metode transaksinya melalui transfer mata uang digital. (Kalakota dan Whinston, 1997). Dengan adanya penerapan e-commerce dapat membantu pengembangan proses bisnis menjadi lebih efektif dan efisien, meningkatkan kredibilitas, meluaskan jangkauan target market, sarana promosi produk, dan juga dapat meningkatkan pelayanan bisnis, yang mempermudah pelanggan untuk mendapatkan informasi mengenai produk yang ditawarkan, pembelian, pembayaran bahkan dalam penyampaian ulasan secara *online* di manapun dan kapanpun, sehingga dapat dibandingkan dengan bisnis offline, proses bisnis online dapat berjalan lebih luas.

Meskipun e-commerce di Indonesia sudah berkembang pesat, masih terdapat beberapa masalah yang dihadapi, salah satunya transaksi pembayaran. Metode pembayaran merupakan komponen penting dalam penggunaan *e-commerce*, dikarenakan beberapa pelanggan memiliki pertimbangan dalam melakukan transaksi online agar tidak terdapat penipuan. Kebanyakan e-commerce masih menawarkan metode secara konvensional. Oleh karena itu, diperlukan suatu metode pembayaran yang dapat memudahkan pengguna dalam melakukan transaksi pembayaran yaitu dengan menerapkan *payment gateway*. *Payment gateway* adalah layanan konfirmasi pembayaran perantara antara situs web penjualan dan sistem online di mana pihak ketiga atau bank langsung memproses transaksi online. Midtrans menjadi salah satu *payment gateway* terbaik yang dimiliki Indonesia, dengan berbagai metode pembayaran, antara lain pembayaran dengan kartu, transfer bank, direct debit, e-wallet, counter, dll, yang dapat memenuhi kebutuhan bisnis online.

Semakin dapat dilihat bisnis butik terus menunjukkan kemajuan dan perkembangannya, ditambah dengan minat masyarakat Indonesia yang tinggi akan keinginan mengikuti trend fashion membuat meningkatnya minat untuk berbelanja pakaian. Bisnis butik akan menjadi peluang bisnis dengan pangsa pasar yang luas, salah satunya bisnis butik kebaya iffah, butik ini merupakan butik kebaya yang memiliki toko di jalan lasem no.5 Gresik kota baru Kabupaten Gresik Jawa Timur butik ini menjual berbagai macam kebaya, mulai dari kebaya pesta hingga kebaya wisuda hasil produksi sendiri. Menurut wawancara dengan pemilik butik, iffah Maulidah menjelaskan bahwa proses bisnis kebaya iffah masih dilakukan dengan cara konvensional yaitu kebaya diproduksi, kemudian

dipromosikan, kemudian dilakukan transaksi pembelian, dan barang dikirim bila ada permintaan. Media sosial Instagram dan WhatsApp masih menjadi sarana pemasaran dan penjualan kebaya miliknya, dilakukan usaha dengan menambahkan *value* pada *feed Instagram* namun hanya dapat meningkatkan sebesar 15% penjualan berdasarkan data laporan keuangan bulanan selama satu tahun terakhir ini. Yang tercatat dalam satu minggu, Butik Kebaya Iffah dapat menjual sebanyak 125 pcs dengan kurang lebih 500 pcs dapat terjual dalam sebulan. Dari situlah dirasakan meningkatnya pendapatan dan jangkauan bisnis jika terus dikembangkan dengan pemanfaatan teknologi yang lebih seperti *E-commerce*, karena upaya tersebut dirasa masih kurang efektif dan belum memberikan dampak promosi yang besar pada bisnisnya. Selain itu dari hasil wawancara didapati selain menjual produk ready butik kebaya Iffah juga menjual produk custom ukuran sesuai yang diinginkan pelanggan, namun hal ini juga berjalan kurang efektif, dikarenakan untuk melakukan pemesanan pelanggan harus datang ke toko untuk dilakukan pengukuran. Dalam masalah pembayaran pesanan pun, banyak permintaan pelanggan agar dapat dilakukan lebih banyak pilihan metode pembayaran, juga bagi admin proses pembayaran diharapkan dapat berjalan lebih aman, karena agar tidak didapati penipuan yang dikarenakan admin tidak dapat memantau secara *real-time* pembayaran yang dilakukan pelanggan. Pemilik merasakan bahwa belum secara maksimal memanfaatkan teknologi informasi agar dapat bersaing dengan butik lainnya.

Terdapat penelitian terdahulu terkait pengembangan *e-commerce* dengan pemanfaatan *payment gateway*. Penelitian pertama yang telah dilakukan oleh Calvin Jardinia Tanuwidjaja, Alexander Setiawan (2017), yang mana pada penelitian tersebut dibangun website *e-commerce* pada toko aksesoris komputer yang dibangun menggunakan *framework Laravel* dan implementasikan *payment gateway Midtrans* sebagai sarana transaksi pembayaran. Sistem ini bertujuan dapat menjangkau customer yang jauh dari toko, sehingga dapat melakukan pembelian dan pembayaran secara *online*. Penelitian lain yang menggunakan kombinasi metode serupa namun objek permasalahan berbeda seperti yang dilakukan, oleh Yuninda Eka Nisrina, Widhy Hayuhardhika Nugraha Putra, dan Buce Trias Hanggara (2019), Sistem dibangun dengan menggunakan *payment gateway Midtrans* namun studi kasus pada penelitian ini dilakukan pada bidang pemesanan tiket. Dengan adanya sistem pemesanan tiket, yang sebelumnya manual dengan memberikan bukti nota menjadi lebih efektif dan efisien.

Dalam penelitian ini, akan dibangun *e-commerce* butik kebaya Iffah yang dapat menghubungkan antara pengelola toko dengan calon pembeli. Untuk mempermudah transaksi pembayaran sistem akan terintegrasi dengan *Payment Gateway* untuk memberikan pengalaman bertransaksi yang lebih efektif dan efisien, dengan menggunakan layanan sistem pembayaran *Midtrans*. Pengembangan sistem pada penelitian ini akan dilakukan dengan metode *waterfall* dan diimplementasikan menggunakan bahasa pemrograman PHP dengan menggunakan *framework Codeigniter 3.1.11*. Dalam menguji hasil akhir sistem dari penelitian ini, pengujian akan dilakukan dengan metode *black box testing* untuk menguji fungsi-fungsi dari sistem agar tidak terdapat kesalahan.

compatibility testing untuk memastikan sistem dapat berjalan pada lingkungan yang berbeda, dan pengujian *user acceptance testing* agar dapat dipastikan bahwa sistem dibuat sesuai dengan kebutuhan user. Diharapkan dengan adanya *e-commerce* butik kebaya iffah yang terintegrasi dengan *payment gateway* midtrans akan meningkatkan penghasilan bisnis pada butik kebaya iffah, memudahkan konsumen dalam berbelanja dalam masa pandemi COVID-19 ini serta dapat meningkatkan kredibilitas bisnis.

Berdasarkan uraian permasalahan yang sudah dijelaskan di atas, akan dikembangkan penelitian dengan judul “Pengembangan Website E-commerce Menggunakan Payment Gateway Midtrans (Studi Kasus : Butik Kebaya Iffah)”.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan di atas, rumusan masalah yang ada dalam pelaksanaan penelitian ini dapat diuraikan sebagai berikut:

1. Bagaimana hasil analisis kebutuhan dan rancangan *website e-commerce* butik kebaya iffah menggunakan *payment gateway* midtrans?
2. Bagaimana mengimplementasi dan mengintegrasikan *website e-commerce* butik kebaya iffah dengan sistem pembayaran *payment gateway* midtrans?
3. Bagaimana hasil pengujian dari *website e-commerce* butik kebaya iffah dengan *payment gateway* midtrans?

1.3 Tujuan

Tujuan yang ingin dicapai pada penelitian ini adalah sebagai berikut :

1. Mendapatkan analisis kebutuhan dan rancangan *website e-commerce* butik kebaya iffah menggunakan *payment gateway* midtrans
2. Mendapatkan implementasi dan Mengintegrasikan *website e-commerce* butik kebaya iffah dengan sistem pembayaran *payment gateway* midtrans
3. Mendapatkan hasil pengujian *website e-commerce* butik kebaya iffah dengan *payment gateway* midtrans

1.4 Manfaat

Dengan terwujudnya tujuan dari penelitian ini diharapkan memberikan manfaat bagi semua pihak, Manfaat yang diharapkan dari pelaksanaan penelitian ini adalah, *website e-commerce* dapat membantu menaikkan penjualan pada butik kebaya iffah pada masa pandemi COVID-19 , memudahkan butik untuk merespon dengan cepat dan berinteraksi langsung dengan pelanggan, memudahkan pembeli untuk mengakses dan melakukan transaksi, mengurangi kesalahan transaksi pembelian, dan meningkatkan kemudahan dan keamanan transaksi pembayaran pada e-commerce butik kebaya iffah.

1.5 Batasan Masalah

Adapun hal-hal pada penelitian ini yang menjadi batasan masalah, yaitu :

1. Sistem e-commerce yang dibangun berbasis web
2. Sistem terIntegrasi menggunakan API payment gateway midtrans terbatas pada lingkup *sandbox*
3. Hanya melakukan pelayanan pemesanan produk untuk user yang telah melakukan registrasi.

1.6 Sistematika Pembahasan

Untuk mempermudah Penyusunan Laporan agar lebih jelas dan sistematis, maka penulis membaginya dalam sistematika penulisan yang terdiri dari beberapa bab pembahasan dengan urutan sebagai berikut :

BAB 1 Pendahuluan

Bab ini membahas mengenai latar belakang, rumusan masalah, tujuan, manfaat, dan batasan masalah serta sistematika penulisan yang digunakan dalam penyusunan skripsi ini.

BAB 2 Landasan Kepustakaan

Pada bab kedua menjelaskan mengenai kumpulan teori yang terkait dengan pengerjaan skripsi ini untuk membantu dalam penulisan mulai dari penelitian terdahulu sampai definisi istilah yang dipakai.

BAB 3 Metodologi

Bab ini akan membahas metodologi yang digunakan dalam penyusunan laporan dan pengembangan sistem dalam skripsi ini.

BAB 4 Analisis Kebutuhan dan Perancangan Sistem

Bab ini membahas hal-hal terkait proses penggalan kebutuhan sistem dan identifikasi aktor dalam pengembangan sistem pada skripsi ini. Kemudian akan dilanjutkan untuk proses perancangan sistem berdasarkan analisis yang telah dibuat

BAB 5 Implementasi

Bab ini membahas tentang implementasi dari perancangan sistem informasi dalam bentuk *database* , kode program dan

user interface berdasarkan rancangan yang telah dilakukan sebelumnya.

BAB 6 Pengujian

Bab ini akan memuat tentang pengujian sistem dikembangkan dengan menggunakan 3 metode yaitu *Black Box Testing*, *Compatibility Testing*, dan *User Acceptance Testing (UAT)*.

BAB 7 Penutup

Bab ini akan menjelaskan uraian kesimpulan dan saran dari keseluruhan penulisan skripsi ini.



BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini akan dijelaskan beberapa uraian mengenai teori dan penelitian yang telah dilakukan sebelumnya demi mendukung pengerjaan penelitian ini, yang berkaitan dengan pembangunan sistem e-commerce dan pengimplementasian payment gateway. Selain itu pada bab ini juga dilakukan pembahasan mengenai pengertian pengertian yang berhubungan dengan isi penelitian, dan menjadi bahan landasan awal.

2.1 Butik Kebaya Iffah

Kebaya Iffah adalah brand yang terbentuk mulai tahun 2015 yang dimulai oleh seorang owner bernama iffah maulidiyah, brand ini merupakan brand yang bergerak pada bisnis fashion terutama kebaya masa kini. Butik Kebaya iffah memiliki toko yang berada Di jalan lasem no.5 Gresik kota baru Kabupaten Gresik Jawa Timur. Butik ini menjual berbagai macam kebaya, mulai dari kebaya pesta,kebaya lamaran, hingga kebaya wisuda hasil produksi jahitan sendiri. Oleh karena itu butik ini memiliki penawaran tipe pembelian produk dimana selain produk *ready*, pelanggan dapat memesan produk butik secara *custom* sesuai ukuran bahkan mulai tahun 2017 Butik ini merambah tidak hanya menjual kebaya saja tetapi juga menjual, hijab, *dress couple*, dll. Target utama pembeli butik kebaya iffah sendiri adalah perempuan dewasa dan anak remaja yang hendak wisuda, dan juga Ibu Ibu. Sampai kini bisnisnya terus berkembang dan dapat dikirim ke seluruh Indonesia. Pada Gambar 2.1 merupakan logo dari Butik Kebaya Iffah.



Gambar 2. 1 Logo Butik Kebaya Iffah

Sumber : (Iffah Maulidiyah,2020)

2.2 Kajian Pustaka

Terdapat penelitian terdahulu terkait pengembangan *e-commerce* dengan pemanfaatan *payment gateway*. Penelitian pertama adalah penelitian yang dilakukan oleh Erick Febriyanto, Untung Rahardja, Niko Alnabawi pada tahun 2018 dengan judul “Penerapan Midtrans sebagai Sistem Verifikasi Pembayaran pada Website iPanda”. Sebelumnya, sistem pembayaran pada website iPanda, masih menggunakan transaksi tunai di kasir dan tidak fleksibel dan efisien, maka dibangunlah sistem ini yang terintegrasi dengan *payment gateway* Midtrans. Penelitian ini menggunakan metode analisis SWOT dan Literature Review. Penelitian ini bertujuan memudahkan pembeli untuk membayar dan memudahkan admin dalam menyiapkan laporan pembelian dan penjualan.

Penelitian yang kedua berjudul “Perancangan dan Pembuatan Website E-Commerce pada Toko Aksesori Komputer di Surabaya” yang ditulis oleh Calvine Jardinia Tanuwidjaja, Alexander Setiawan pada tahun 2017 sistem *e-commerce* dibangun dengan menggunakan *framework* laravel dan implementasikan *payment gateway* Midtrans sebagai sarana transaksi pembayaran. Sistem ini dirancang untuk menjangkau pelanggan yang jauh dari toko aksesoris komputer, sehingga bisa melakukan pembelian dan pembayaran secara *online*.

Penelitian ketiga berjudul “Perancangan Pemesanan Paket Wisata dengan Pembayaran Online Menggunakan Payment Gateway pada aplikasi Android (Studi Kasus: CV Ambarawa Raya Travel Agent)” oleh Fajar Ardhanta Hendraswara, Suprihadi, S.Si, M.Kom, Charitas Fibriani, S.Kom M.Eng pada tahun 2016. Tujuan penelitian ini merancang sebuah aplikasi Android untuk Sistem Pariwisata Kabupaten Semarang, yang dapat memesan kamar hotel, produk perjalanan, atau paket perjalanan. dengan dilengkapi Google maps API dan pembayaran online menggunakan *payment gateway* veritrans. Penelitian ini menggunakan metode *prototype*. Data dikumpulkan dari objek wisata beserta fasilitas pendukung pariwisata. Diharapkan sistem ini dapat mempromosikan dengan efektif agar menarik lebih banyak wisatawan.

Penelitian keempat berjudul “Pengembangan E - Commerce Dengan Pemanfaatan Sistem Payment Gateway (Studi Kasus: Wisata Kampung Sapi Adventure)” oleh Yuninda Eka Nisrina, Widhy Hayuhardhika Nugraha Putra, dan Buce Trias Hanggara pada tahun 2019. Penelitian ini bertujuan membangun *e-commerce* pembelian tiket pada kampung wisata sapi dengan menggunakan pembayaran online. Sistem dibangun dengan menggunakan *payment gateway* midtrans. Dengan adanya sistem pemesanan tiket yang sebelumnya manual dengan memberikan bukti nota menjadi lebih efektif dan efisien.

Penelitian kelima berjudul “Implementasi Payment Gateway Menggunakan Midtrans Pada Marketplace Travnesia.com” oleh Tri May Mega Puspasari, Dina Maulina pada tahun 2019. Penelitian ini dilakukan dengan tujuan meningkatkan performa layanan pada website travnesia.com yang sudah ada sebelumnya namun masih menggunakan metode pembayaran secara manual sehingga dibangun fitur yang dapat memudahkan customer dalam melakukan transaksi

pada travnesia.com menggunakan layanan payment gateway midtrans, agar customer dengan mudah memesan paket wisata.

2.3 Sistem Informasi

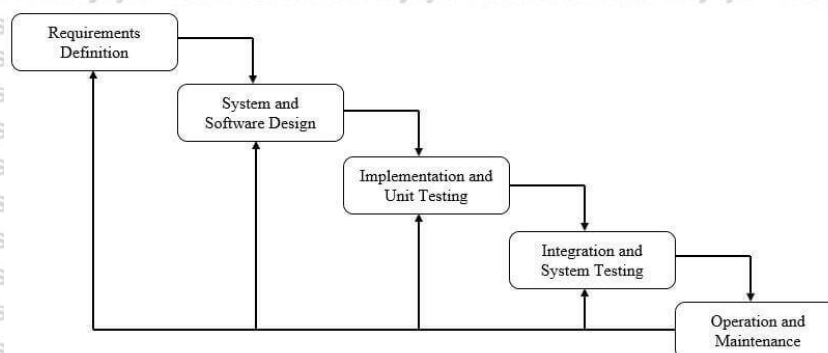
Sistem informasi merupakan kumpulan orang, prosedur, dan sumber daya tim yang berinteraksi untuk mencapai sebuah tujuan yang ditetapkan, elemen tersebut saling berinteraksi untuk melakukan aktivitas mengumpulkan, melakukan manipulasi, menyimpan, menyebarkan data dan informasi (Janry Haposan U. P. Simanungkalit, S.Si., 2012). Sedangkan Sistem informasi berbasis komputer (computer-based information system - CBIS) merupakan sistem informasi yang menggunakan teknologi komputer dengan komponen pendukung yang terdiri dari perangkat keras, perangkat lunak, database, telekomunikasi, orang dan prosedur yang saling terikat.

2.3.1 System Development Life Cycle (SDLC)

SDLC merupakan tahapan kerja yang dilakukan oleh analis sistem dan pemrogram dalam pembangunan sistem informasi (Dwanoko, 2016). SDLC adalah metode umum yang sering dipergunakan dalam pengembangan sistem informasi meliputi berbagai tahapan mulai dari fase perencanaan, analisis, perancangan, implementasi hingga pemeliharaan sistem. SDLC berkonsep sebagai dasar model pengembangan lain, agar terbentuk kerangka kerja dalam proses pembuatan sistem informasi maupun perencanaan juga (Susanto & Andriana, 2016).

SDLC merupakan proses proses terstruktur yang dijadikan acuan oleh seorang project manager dan tim dalam Mengembangkan sistem informasi yang dirancang untuk memecahkan masalah secara efektif. agar siste yang dihasilkan memiliki kualitas tinggi yang memenuhi keinginan pelanggan atau kesesuaian tujuan awal pembuatan sistem.

2.3.1.1 Metode Pengembangan Waterfall



Gambar 2. 2 Tahapan Metode SDLC Waterfall

Sumber: (Sommerville, 2011)

Waterfall merupakan SDLC yang sangat sering digunakan dalam melakukan pengembangan sistem. Menurut sommerville (Sommerville, 2011) metode ini terdiri dari 5 tahapan seperti yang dijelaskan pada gambar 2.1.

Model Waterfall atau sering disebut model konvensional adalah model SDLC yang sering digunakan. Model pengembangan ini bersifat linear system, Tahap selanjutnya tidak akan dilanjutkan sampai tahap sebelumnya selesai, dan tahap sebelumnya tidak dapat dikembalikan atau diulangi lagi tahap tersebut (Susanto & Andriana, 2016).

Kelemahan metode ini adalah diperlukan biaya yang tinggi, mudahnya kesalahpahaman komunikasi dengan pengguna, dan metode ini tidak cocok untuk proyek yang harus beradaptasi dengan requirement. Namun, kelebihan dari penggunaan metode ini adalah karena setiap aktivitas dilakukan dalam waktu yang cukup sehingga akan menghasilkan kualitas sistem yang baik.

2.4 Website

Website merupakan layanan dengan menggunakan internet yang dapat diakses secara luas menggunakan browser berisi kumpulan halaman-halaman web yang dapat berisi gambar, video, dan file digital lainnya yang saling berhubungan. Halaman web dapat diakses dari nama domain setelah web tersebut disimpan pada hosting atau web server. Website dapat dilihat melalui perangkat untuk mengakses internet seperti komputer dll. World Wide Web (WWW) ini adalah sistem jaringan berbasis Client-Server menggunakan protokol transfer hypertext atau protokol HTTP serta menggunakan TCP/IP. Website dibagi menjadi dua macam dari segi dinamisnya (Hartono, 2014). Yaitu :

1. Website Statis merupakan halaman web yang menampilkan halaman konten yang tidak berubah-ubah atau menampilkan bentuk yang sama saat diakses oleh user karena sama dengan file halaman web yang disimpan di server, biasanya menggunakan bahasa pemrograman Hypertext Markup Language (HTML) atau pun Cascading Style Sheets (CSS).

2. Website dinamis halaman web yang strukturnya memiliki kemampuan untuk terus menyesuaikan dengan keadaan ketika diakses oleh user, website akan update sesering mungkin karena mengalami pemrosesan atau parsing. Website ini umumnya dikembangkan dengan bahasa pemrograman Perl, PHP, Javascript, Phyton dan dan lain sebagainya.

2.5 E-commerce

E-commemrce atau Electronic commerce Ini adalah proses pembelian dan penjualan barang atau jasa secara online memanfaatkan teknologi informasi dengan menghubungkan pemilik bisnis dan konsumen. E-commerce dapat membantu mengurangi biaya administrasi dan waktu siklus untuk proses bisnis, dan meningkatkan hubungan dengan mitra bisnis dan pelanggan (Pradana, 2015).

E-commerce termasuk bagian dari *e-businness* yang secara umum, konsumen membutuhkan teknologi atau infrastruktur Internet yang digunakan untuk menemukan toko online guna membeli secara online melalui Internet. Dalam istilah , penjual atau penyedia layanan *e-commerce* dapat mengumpulkan informasi, seperti data konsumen, yang umumnya membutuhkan konsumen untuk menjadi anggota sebelum melanjutkan dengan lebih banyak transaksi (Nanda & Maharani, 2018).

Munculnya e-commerce memberikan beberapa dampak positif bagi kegiatan pemasaran, antara lain (Irmawati, 2011):

1. Mempermudah promosi produk dan layanan secara interaktif dan real-time melalui saluran komunikasi langsung di Internet
2. Membuat saluran distribusi baru yang dapat menjangkau lebih banyak pelanggan di hampir seluruh wilayah dunia
3. Menghemat waktu siklus dan tugas manajemen dari pemesanan hingga pengiriman produk
4. Layanan pelanggan lebih responsif dan memuaskan, karena pelanggan dapat memperoleh informasi lebih detail secara online dan merespon dengan cepat
5. Hemat waktu dan uang dalam manajemen pesanan, karena sistem pesanan elektronik dapat mencapai pemrosesan yang lebih cepat dan lebih akurat

Tujuan penggunaan *E-commerce* adalah untuk memudahkan pelanggan untuk memesan dan membayar transaksi. pelanggan dapat melakukan pembelian produk meskipun jauh dari butik, selama ada jaringan. Setelah melakukan pemesanan maupun *custom* ukuran dan membayar nominal transaksi, dan setelah itu admin akan mengirimkan produk. dengan harapan dapat mempermudah dan mempercepat proses pembelian produk.

Melalui e-commerce, perusahaan dapat mengubah daftar harga atau menyesuaikan produk atau layanan yang disediakan, dan dapat dipelajari dengan

cepat melalui situs web. E-commerce juga berguna untuk menghimpun database pelanggan yang lengkap sehingga perusahaan dapat memperoleh informasi tentang pola pesanan pelanggan dan mengelolanya sebagai informasi yang berharga. Ketika menerapkan strategi pemasaran dan promosi, database akan membantu mencapai tujuan bisnis (Irmawati, 2011).

2.5.2 Klasifikasi E-commerce

Klasifikasi jenis situs web e-commerce menurut model bisnisnya (Pradana, 2015), antar lain:

1. Business to Business (B2B)

Kegiatan usaha dilakukan antara perusahaan dengan perusahaan lain. Biasanya jenis ini digunakan oleh produsen dan grosir.

2. Business to Consumer (B2C)

Jenis transaksi ini dilakukan antara produsen atau perusahaan dengan konsumen akhir. Konsumen akhir biasanya berupa komoditi yang tidak dijual kembali. Biasanya beberapa jenis toko online yang menjual berbagai produk.

3. Consumer to consumer (C2C)

Jenis transaksi ini dilakukan antara konsumen dengan bertukar informasi tentang barang dan jasa. Biasanya pembeli tidak menjual kembali, penjual hanya memiliki beberapa barang.

4. Consumer to business (C2B)

Jenis transaksi ini akan berlangsung dari konsumen ke bisnis. Konsumen akan memberikan produk atau jasa kepada pelaku usaha yang membutuhkannya.

5. B2G (Business to Government)

Ini adalah produk turunan B2B, dan transaksi semacam itu akan dilakukan antara pengusaha dan lembaga pemerintah.

6. G2C (Government to Consumer)

Yaitu hubungan atau interaksi antara pemerintah dengan masyarakat. Konsumen dalam hal ini masyarakat dapat dengan mudah berkomunikasi dengan pemerintah untuk mendapatkan pelayanan yang nyaman.

2.6 Bahasa Pemrograman

2.6.1 Hypertext Markup Language (HTML)

Proses menampilkan halaman web di browser melibatkan HTML. *Hypertext Markup Language* (HTML) adalah format yang digunakan untuk membuat dokumen dapat dibaca di Web.

Menurut Solichin (Solichin, 2016) HTML adalah bahasa pemrograman web yang memberi tahu browser web cara membangun dan menyajikan konten di

halaman web. Sedangkan menurut (Arief, 2011) HTML adalah salah satu format yang digunakan untuk membuat dokumen atau aplikasi yang berjalan pada halaman web.

Berdasarkan penjelasan ahli diatas, penulis dapat menyimpulkan bahwa HTML (*Hypertext Markup Language*) adalah bahasa pemrograman yang digunakan sebagai standart pemrograman dalam membuat halam web yang dikenal oleh *browser*.

2.6.2 Hypertext Preprocessor (PHP)

Merupakan bahasa pemrograman server-side scripting yang dirancang untuk pengembangan dan desain website ini juga digunakan sebagai script untuk mempercantik layar website . Berikut beberapa pengertian dari hypertext preprocessor (php).

Menurut Supono dan Putratama (Supono, 2016) bahwa *Hypertext Preprocessor* sebagai bahasa pemrograman dapat dipergunakan untuk mengubah baris kode program menjadi kode mesin yang dapat dipahami oleh komputer sisi server, serta dapat ditambahkan ke dalam HTML. Sedangkan menurut (Arief, 2011) PHP (*Hypertext Preprocessor*) adalah bahasa skrip sisi server yang menggabungkan HTML untuk membuat halaman web dinamis.

Penulis menyimpulkan bahwa PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman yang menggunakan server untuk mengeksekusi seluruh proses dari sintaks dan perintah yang diberikan, atau skrip sisi server, yang terintegrasi dengan HTML untuk membuat sebuah halaman web dinamis.

2.6.3 CSS (*Cascading Style Sheet*)

CSS atau bahasa pemrograman cascading style sheet diusulkan oleh Hakon Wilum Lie pada tahun 1994 dan distandarisasi oleh W3C untuk meningkatkan tampilan Web (Solichin, 2016). CSS adalah skrip yang dapat digunakan untuk mengubah halaman web menjadi bentuk web yang lebih indah dan menarik. Cascading Style Sheets (CSS) digunakan untuk menampilkan halaman web yang menarik, memperindah tampilan halaman web, dan mudah digunakan. Berisi sekumpulan instruksi yang menentukan cara menampilkan teks pada halaman web.

2.6.4 Javascript

JavaScript adalah bahasa pemrograman yang dapat melampirkan dan memproses halaman HTML. Ini juga disebut bahasa scripting , karena program JavaScript dapat mengeksekusi secara langsung tanpa kompilasi. Javascript sering digunakan dalam aplikasi berbasis web untuk menyediakan pesan yang menarik bagi pengguna dan menampilkan dukungan di situs web agar lebih interaktif. JavaScript bersifat open source, siapa pun dapat menggunakan, membuat, atau memodifikasinya secara gratis.

2.7 Proses Bisnis

Proses Bisnis dapat diartikan sebagai rangkaian aktifitas-aktifitas yang dilakukan oleh suatu bisnis. Dimulai dari inisiasi input transformasi informasi hingga output yang akan dihasilkan dan Hal ini dapat digunakan untuk meningkatkan kinerja proses bisnis perusahaan. Suatu organisasi akan mampu untuk meningkatkan proses bisnisnya dan berarti dapat merespon atau beradaptasi terhadap perubahan di lingkungan internal dan eksternal. Terdapat tiga tahapan utama, yaitu Ongoing Process Modeling (As-Is Model), Desired Process Modeling (To-Be Model), dan yang ketiga adalah Execution Process. Tahap pertama dimulai dengan pemodelan proses bisnis yang sedang berjalan (As-Is Model). Kemudian, menganalisis hasil pemodelan sehingga menghasilkan model yang menggambarkan proses perbaikan (To-Be Model). Tahap terakhir adalah untuk mengimplementasikan model Tobe (Nurhayati & Setiadi, 2017)

2.7.1 Business Process Modeling Notation (BPMN)

BPMN adalah singkatan dari Business Process Modeling Notation, sebuah metode dari pemodelan proses bisnis, dan alat desain untuk sistem berbasis pesan. BPMN bertujuan untuk memberikan simbol dengan kemudahan penggunaan dan mudah dipahami oleh *stakeholder* yang terlibat dalam kegiatan berjalannya bisnis (Nurhayati & Setiadi, 2017). ada 4 kategori jenis elemen di BPMN:

1. Flow object :

- Event, berupa lingkaran start,intermediate,atau end
- Activities, berupa persegi panjang dengan sudut radius dan menjelaskan pekerjaan yang dilakukan
- Gateway,berupa belah ketupat dengan isi yang berbeda beda sesuai pilihan aktifitas atau percabangan dan gabungan path yang dijalankan

2. Connecting Objects

- Sequence Flow, berupa garis lurus dengan panah tertutup sebagai petunjuk urutan aktivitas
- Message Flow, berupa garis putus-putus dan panah terbuka sebagai pertukaran pesan
- Association, berupa garis putus-putus sebagai asosiasi artifak,data

3. Swimlanes

- Pool, berupa persegi besar sebagai tempat penggambaran proses
- Lane, bagian dari pool

4. Artifact

- Data Objects, menjelaskan data yang dibutuhkan pada suatu aktivitas

- Group, berupa persegi dengan garis putus-putus sebagai penggabung aktivitas

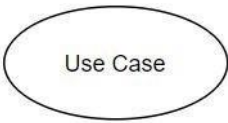


2.8 UML (Unified Modelling Language)

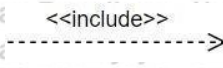

UML (Unified Modeling Language) Ini adalah salah satu bahasa yang paling banyak digunakan untuk mendefinisikan persyaratan, melakukan analisis dan desain, dan menggambarkan desain arsitektur dalam pemrograman berorientasi objek. Menurut Sukamto dan Shalahuddin (Sukamto dan & Shalahuddin(2014:28)., 2014). UML berisikan detail informasi kebutuhan kebutuhan sistem sehingga digunakan sebagai *blue print*. UML, memiliki beberapa macam, diantaranya use case, class, activity dan sequence diagram.

2.8.1 Use case diagram

Use case merupakan pemodelan yang mendeskripsikan satu atau lebih aktor dengan sistem, dan digunakan untuk mengetahui fungsi apa saja yang ada pada sistem (Sukamto dan & Shalahuddin(2014:28)., 2014). Use case menjelaskan behavior dalam sebuah model yang menjelaskan interaksi antara aktor mengenai sistem yang ada. Pada Tabel 2.1 akan dijelaskan komponen-komponen yang ada pada *Use Case Diagram*.

Tabel 2.1 Komponen-komponen pada *use case diagram*

Simbol	Nama	Keterangan
	Use Case	Bertuliskan nama usecase yaitu merupakan gambaran fungsionalitas sistem yang akan dibuat. Biasanya dinyatakan menggunakan kata kerja
	Aktor	Subjek yang berinteraksi dengan sistem yang akan dibuat. Biasanya dinyatakan dengan kata benda
	Asosiasi	Komunikasi antar aktor dan use case maupun sebaliknya.

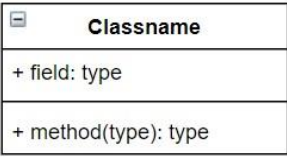




	Include	Relasi antar use case dimana salah satu usecase
	Extend	Relasi antar use case dimana salah satu usecase tersebut dapat berdiri sendiri.

Sumber : (Sukamto dan & Shalahuddin(2014)., 2014)

2.8.2 Class diagram

Diagram kelas merupakan gambaran dari struktur sistem yaitu pendefinisian kelas kelas yang dibuat untuk membangun sistem (Sukamto dan & Shalahuddin(2014:28)., 2014). Diagram bersifat statis, menunjukkan sekumpulan kelas, antarmuka, kolaborasi, dan hubungan dari keseluruhan sistem yang akan dibuat.. Pada Tabel 2.2 akan dijelaskan komponen-komponen dari *Class Diagram*.

Tabel 2.2 Komponen-komponen pada *class diagram*

Simbol	Nama	Keterangan
	<i>Class</i>	Kelas pada struktur sistem
	<i>Association</i>	Hubungan antar kelas yang memiliki makna umum
	<i>Directed Association</i>	Hubungan antar kelas dimana Satu kelas digunakan oleh kelas lain
	<i>Generalization</i>	Hubungan antar kelas yang bermakna umum-khusus
	<i>Aggregation</i>	Hubungan antar kelas dimana satu kelas merupakan bagian dari kelas yang lain

	<i>Dependency</i>	Hubungan antar kelas yang menggambarkan ketergantungan sebuah kelas terhadap kelas lain
--	-------------------	---

Sumber: (Sukamto dan & Shalahuddin(2014:28)., 2014)

2.8.3 Activity diagram

Aktivitas diagram menceritakan langkah alur kerja sebuah kegiatan menu dalam sistem bisnis atau proses atau perangkat lunak. (Sukamto dan & Shalahuddin(2014:28)., 2014).Diagram ini menjelaskan alur aktifitas atau fungsi yang ada pada sistem. Pada Tabel 2.3 akan dijelaskan komponen-komponen dari *Class Diagram*.

Tabel 2.3 Komponen-komponen pada *activity diagram*

Simbol	Nama	Keterangan
	<i>Start Point (Initial Node)</i>	Aktivitas sistem dimulai
	<i>Activity</i>	Aktivitas yang dilakukan oleh sistem. Biasanya dalam bentuk kata kerja
	<i>Edge (Control Flow)</i>	Menghubungkan satu simbol dengan simbol lainnya
	<i>Decision /percabangan</i>	Percabangan seleksi-kondisi, terdapat beberapa pilihan aktivitas
	<i>Join/ penggabungan</i>	Gabungkan dua aktivitas, yang akan digabungkan menjadi satu
	<i>End Point</i>	Activity sistem berakhir


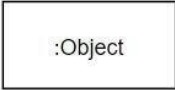


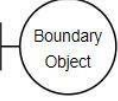
	<p><i>Swimlane</i></p>	<p>Pemisah organisasi bisnis yang bertanggung jawab dari kegiatan yang terjadi</p>
---	------------------------	--

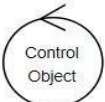


Sumber: (Sukamto dan & Shalahuddin(2014:28)., 2014)

2.8.4 Sequence diagram

Diagram sekuen menjelaskan bagaimana objek berperilaku dalam use case dengan menjelaskan umur objek dan pesan yang dikirim dan diterima antara objek. (Sukamto dan & Shalahuddin(2014:28)., 2014). Pada diagram ini, bersifat dinamis yang menggambarkan pengiriman pesan pada waktu tertentu. Pada Tabel 2.4 akan dijelaskan komponen-komponen dari *Class Diagram*.

Tabel 2.4 Komponen-komponen pada *sequence diagram*

Simbol	Nama	Keterangan
	<p><i>Actor</i></p>	<p>Subjek yang berinteraksi dengan sistem yang dibuat</p>
	<p><i>Object</i></p>	<p>Representasi dari class/object yang berinteraksi pesan</p>
	<p><i>Lifeline</i></p>	<p>Menyatakan umur hidup sebuah object</p>
	<p><i>Activation</i></p>	<p>Menunjukkan bahwa object sedang aktif atau sedang berinteraksi dengan object lain</p>
	<p><i>Boundary Object</i></p>	<p>Batasan sistem (<i>boundary</i>). berupa <i>user interface</i> yang dapat dilihat oleh aktor</p>

	<i>Control Object</i>	Aliran informasi dari skenario akan diatur
	<i>Entity Object</i>	Penyimpanan data atau informasi
	<i>Message Entry</i>	Menunjukkan <i>object</i> sedang berinteraksi dengan <i>object</i> lain
	<i>Message to Self/Return</i>	Mengembalikan <i>object</i> setelah berinteraksi dengan <i>object</i> lain

Sumber: (Sukamto dan & Shalahuddin(2014:28)., 2014)

2.9 MySQL

Menurut Anhar (2010), MYSQL merupakan perangkat lunak SQL (database management system) yang mudah didapatkan secara gratis. Menurut Raharjo (2011), "MySQL adalah RDBMS (atau database server) yang mengelola database dengan cepat menampung jumlah yang sangat besar dan dapat diakses oleh banyak pengguna."

Dapat disimpulkan bahwa MYSQL adalah salah satu server database open source paling terkenal yang digunakan untuk membuat aplikasi web. MYSQL menggunakan SQL (structured query language) yang sering digunakan dalam pemrograman.

Beberapa kelebihan MYSQL, diantaranya :

1. MYSQL mendukung sistem operasi, seperti Windows, Linux, FreeBSD, Mac Os X Server solaris.
2. Bersifat Open Source, dapat dengan mudah didapatkan secara gratis.
3. Sintaksnya lebih mudah dipahami dan tidak rumit.
4. Multi-pengguna, banyak pengguna dapat menggunakan secara bersamaan.
5. Memiliki kemampuan memproses data dengan kecepatan tinggi.

2.10 Software Pendukung

2.10.1 Visual Studio Code

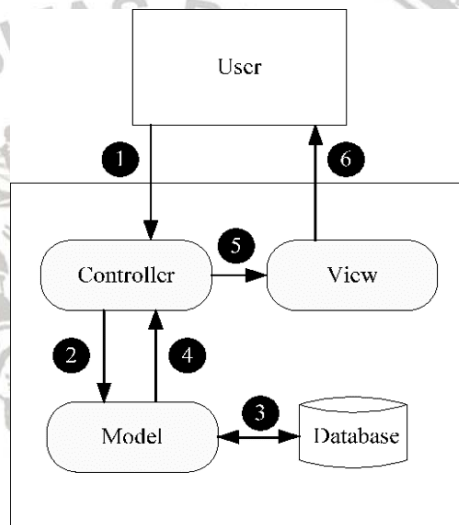
VSCode adalah editor kode sumber yang dapat menggunakan berbagai macam bahasa pemrograman. Karena performanya yang canggih dan software yang ringan Visual Studio Code sering digunakna untuk membuat atau membangun aplikasi web.

2.10.2 Framework

Kerangka kerja adalah kerangka kerja yang digunakan dalam perangkat lunak, yang menyediakan fungsi umum sehingga dapat dimodifikasi oleh kode yang dibuat pengguna, yang akan membantu menyediakan perangkat lunak aplikasi tertentu. (Rahmadiansyah & Irwan, 2012). Dari Penjelasan tersebut dapat disimpulkan Framework berupa kerangka kerja guna mengembangkan aplikasi berbasis desktop atau aplikasi berbasis web agar developer lebih mudah dalam pembuatan aplikasi atau website tanpa perlu membuat fungsi dari awal sehingga lebih cepat dan lebih terstruktur.

2.10.2.1 Model-View-Controller (MVC)

Model View Controller (MVC) metode digunakan membuat aplikasi cara kerjanya dengan memisahkan data dari layar dan cara menanganinya (Rahmadiansyah & Irwan, 2012). Metode MVC membagi aplikasi menjadi tiga bagian yaitu:



Gambar 2. 3 Gambaran Arsitektur ModelViewController (MVC)

Sumber : (Supaartagorn, 2011)

1. Model, bertindak sebagai manajer perilaku dan data dalam domain aplikasi, menanggapi permintaan informasi dan menanggapi instruksi untuk mengubah kondisi (keadaan).
2. View, Mengubah informasi model dalam bentuk yang sesuai untuk interaksi dengan pengguna. Biasanya satu atau lebih elemen *User Interface*.
3. Controller, Menrima input pengguna dan picu respons saat objek model dipanggil.

Gambar 2.3 Arsitektur ModelViewController dijelaskan Pengoperasian suatu proses kerangka kerja dapat dipecah menjadi enam langkah, seperti berikut (Supaartagorn, 2011) :

1. Pengguna mengirimkan permintaan ke controller.

2. Controller menganalisis permintaan dan memanggil model (metode di kelas).
3. Model akan melakukan logika bisnis yang diperlukan dan menghubungkan database.
4. Model mengirimkan hasilnya ke controller.
5. Pengontrol meneruskan permintaan ke tampilan.
6. Permintaan selesai ketika hasilnya merespons pengguna.

2.10.2.2 CodeIgniter

Menurut Hakim (2010), CodeIgniter adalah kerangka kerja PHP yang, dibandingkan dengan menulis semua kode program dari awal, dapat membantu pengembang mempercepat pengembangan aplikasi web berbasis PHP.

Arti dari framework sendiri adalah kumpulan dari fungsi dan class yang dapat digunakan setiap saat sehingga dapat menyelesaikan pekerjaan dengan lebih mudah dan cepat tanpa harus membuatnya dari awal.

CodeIgniter adalah jaringan aplikasi web dengan sumber terbuka sebagai membuat aplikasi PHP yang dinamis, serta Framework PHP yang memiliki konsep pemodelan yang dapat membangun sebuah website lebih dinamis memakai MVC (model, view, controller) , yang dapat mempercepat pembuatan aplikasi Web oleh pengembang.

2.10.3 Bootstrap

Bootstrap adalah kerangka kerja untuk membuat desain web responsif. Dimana halaman web yang dibuat oleh bootstrap akan menampilkan ukuran layar yang sesuai dengan browser yang kita gunakan, baik itu di desktop, tablet, atau perangkat mobile. Penggunaan bootstrap , dapat membantu pembangunan web dinamis atau statis sehinggan menjadikan sesuai,cepat, dan mudah digunakan pada tampilan web kita (Alatas, 2013).

2.11 Application Programming Interface (API)

Application Programming Interface(API) merupakan interface ini berguna dalam proses akses aplikasi atau layanan program. yang digunakan oleh developer untuk mengintegrasikan antar bagian aplikasi yang berbeda secara bersama. Karena pengembangan dapat gunakan fungsi aplikasi lain yang ada, jadi tidak perlu membuat ulang dari awal menjadikan penggunaan API dapat mempercepat proses development.

2.12 Payment Gateway

Payment gateway adalah pintu gerbang atau media transaksi yang disediakan oleh layanan aplikasi e-commerce yang dapat mengotorisasi pemrosesan kartu kredit dan pembayaran langsung dengan pelanggan dalam aktivitas bisnis elektronik / online. Payment gateway ini memudahkan pengusaha dan konsumen dalam melakukan transaksi.

Payment gateway merupakan bagian terpenting dari transaksi antara klien, perusahaan dan institusi perbankan (Damanik, 2012). Beberapa fungsi utama dari gateway pembayaran meliputi:

1. Aplikasi perangkat lunak dirancang untuk perdagangan elektronik, tetapi juga dapat digunakan untuk mengotorisasi pembayaran dalam bisnis fisik tradisional
2. Enkripsi pembayaran dan data pribadi.
3. Komunikasi antara lembaga keuangan, perusahaan, dan klien terkait.
4. Otorisasi pembayaran.

2.12.1 Midtrans

Midtrans merupakan payment gateway yang memenuhi kebutuhan bisnis online dengan menyediakan layanan berbagai metode pembayaran. Layanan ini memungkinkan pelaku industri untuk beroperasi lebih mudah dan meningkatkan penjualan.

Midtrans memiliki 20 metode pembayaran mulai dari bank hingga mini market. Keamanan data perusahaan yang terjamin aman, dan midtrans dapat mendeteksi apabila ada transaksi yang mencurigakan maka transaksi tersebut ditahan dan dilakukan penelusuran melalui telepon oleh pihak midtrans (Midtrans,2021).

2.12.2 Mendapatkan *Snap Token* Transaksi di *Backend*

Menurut dokumentasi di situs resmi Midtrans, perlu dilakukan permintaan API dari *backend merchant* untuk mendapatkan *Snap Token* transaksi. Dimana dalam penelitian ini yaitu *backend* sistem *E-commerce* Butik Kebaya Iffah akan menyediakan informasi pembayaran dan *Server Key*. Setidaknya terdapat tiga komponen yang diperlukan untuk mendapatkan *Snap Token* transaksi, yaitu:

1. *Server Key* : API *server key* digunakan dalam mengakses API Midtrans, didapatkan saat melakukan pendaftaran akun Midtrans. Dapat dilihat pada *dashboard* Midtrans menu *Settings* dan sub-menu *Access Keys*, pada bagian *Server Key*.
2. *Order_id* : ID transaksi bersifat unik, sehingga satu ID hanya dapat digunakan untuk satu kali transaksi. Karakter yang diperbolehkan adalah Alfanumerik, tanda hubung(-), garis bawah(_), tilde (~), dan titik (.) String, maks 50.
3. *Gross_amount* : Jumlah total transaksi yang harus dibayarkan dan harus berupa *integer*.

```

1 // Set your Merchant Server Key
2 \Midtrans\Config::$serverKey = 'YOUR_SERVER_KEY';
3 // Set to Development/Sandbox Environment (default). Set to true for Production
4 \Midtrans\Config::$isProduction = false;
5 // Set sanitization on (default)
6 \Midtrans\Config::$isSanitized = true;
7 // Set 3DS transaction for credit card to true
8 \Midtrans\Config::$is3ds = true;
9
10 $params = array(
11     'transaction_details' => array(
12         'order_id' => rand(),
13         'gross_amount' => 10000,
14     ),
15     'customer_details' => array(
16         'first_name' => 'budi',
17         'last_name' => 'pratama',
18         'email' => 'budi.pra@example.com',
19         'phone' => '0811122333',
20     ),
21 );
22
23 $snapToken = \Midtrans\Snap::getSnapToken($params);

```

Gambar 2. 4 Contoh Request SNAP Token Midtrans

Sumber : (Midtrans, 2021)

2.12.3 Menampilkan Halaman Pembayaran Snap di Frontend

Untuk dapat menampilkan Snap halaman pembayaran, maka disertakan library snap.js pada halaman pembayaran sistem *E-commerce* Butik Kebaya Iffah. Kemudian memasukkan *Client Key* yang didapatkan saat melakukan pendaftaran akun Midtrans sebagai nilai atribut *data-client-key*. Pembayaran dapat dilakukan dengan memanggil fungsi `snap.pay` dan menyertakan token yang diperoleh dari request Snap Token sebelumnya.

```

1 <html>
2 <head>
3 <meta name="viewport" content="width=device-width, initial-scale=1">
4 <script type="text/javascript"
5     src="https://app.sandbox.midtrans.com/snap/snap.js"
6     data-client-key="SET_YOUR_CLIENT_KEY_HERE"></script>
7 <!-- Note: replace with src="https://app.midtrans.com/snap/snap.js" for Production
8 </head>
9
10 <body>
11 <button id="pay-button">Pay!</button>
12 <script type="text/javascript">
13     var payButton = document.getElementById('pay-button');
14     // For example trigger on button clicked, or any time you need
15     payButton.addEventListener('click', function () {
16         window.snap.pay('SNAP_TRANSACTION_TOKEN'); // Replace it with your transaction
17     });
18 </script>
19 </body>
20 </html>

```

Gambar 2. 5 Menampilkan Snap Halaman Pembayaran

Sumber : (Midtrans, 2021)

2.12.4 Melakukan Uji Pembayaran

Untuk memastikan proses mengintegrasikan *Snap* telah berhasil, dapat dilakukan uji pembayaran Dalam mode uji sandbox Midtrans, uji dapat menggunakan simulator pembayaran yang disediakan oleh Midtrans untuk melakukan pembayaran. Simulator Midtrans ini akan membantu dalam menguji pembayaran tanpa harus melakukan pembayaran sebenarnya (Midtrans, 2021). Dalam penelitian ini pembayaran dibatasi pada *EMoney*, transfer bank, dan Kios *Merchant/Minimarket*.

2.12.5 Menangani Setelah Pembayaran

Setelah pembayaran dilakukan, maka status transaksi berubah, pelanggan akan diarahkan ke Redirect URL dan Midtrans mengirimkan notifikasi HTTP ke backend merchant yaitu pada sistem *E-commerce* Butik Kebaya Iffah. Permintaan HTTP POST berupa JSON akan dikirim. Untuk mendapatkan status transaksi dapat menggunakan perintah berikut `\Midtrans\Transaction::status(order_id)`.

2.13 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah serangkaian proses yang dirancang untuk mendapati kesalahan atau error yang muncul dari system perangkat lunak tersebut. Pentingnya dilakukan pengujian dalam pengembangan perangkat lunak agar dipastikan sistem yang dibuat dapat bekerja dengan baik seperti yang diharapkan untuk memastikan kualitas dari sebuah software (Rouf, 2012).

Pengujian perangkat lunak dapat dibedakan menjadi dua yaitu :

1. Black Box : Uji untuk menentukan apakah semua fungsi perangkat lunak beroperasi dengan benar sesuai dengan persyaratan fungsional yang ditentukan
2. White Box : Tes untuk menunjukkan secara detail bagaimana produk bekerja sesuai dengan spesifikasinya

2.12.1 Black Box

Black box testing atau biasa disebut dengan collision testing adalah metode pengujian perangkat lunak yang digunakan untuk menguji perangkat lunak tanpa mengetahui struktur internal dari kode atau program. Metode kotak hitam memungkinkan perekayasa perangkat lunak untuk memperoleh serangkaian kondisi input yang memanfaatkan sepenuhnya semua persyaratan fungsional program (Rouf, 2012).

Kelebihan Black Box Testing yaitu:

1. Efisien untuk segmen kode besar
2. Akses kode tidak diperlukan

3. Pemisahan antara perspektif pengguna dan pengembang

Kelemahan Black Box Testing yaitu:

Cakupan terbatas, karena hanya sebagian kecil dari skenario pengujian yang dijalankan dan penguji tidak beruntung karena pengetahuan perangkat lunak internal, sehingga efisiensi pengujian rendah.

Dilakukan evaluasi keluaran yang diharapkan dari sistem yang diuji untuk menentukan apakah dapat bekerja seperti yang diharapkan. Pengujian black box menggunakan parameter sebagai berikut, yaitu skenario pengujian sistem, keluaran sistem yang diharapkan, dan hasil aktual pengujian sistem. Melalui pengujian black box diharapkan ditemukan kesalahan pada fungsi yang tidak dijalankan dengan benar, kesalahan antarmuka, kesalahan struktur data, dan kesalahan kinerja, inisialisasi dan terminasi.

2.13.2 Compatibility Testing

Uji kompatibilitas perangkat lunak berguna untuk memeriksa apakah perangkat lunak berinteraksi dan berbagi informasi dengan benar dengan perangkat lunak lain. Interaksi ini dapat terjadi secara bersamaan pada komputer yang sama atau antara dua program pada komputer yang berbeda yang terhubung melalui Internet atau melalui kabel jaringan. Untuk pengujian kompatibilitas sangat cocok diterapkan pada halaman web, karena halaman web tersebut pasti berhubungan dengan internet, dan data yang diterima dan dikirim oleh bukanlah data dari komputer. Juga dapat menggunakan uji kompatibilitas untuk menguji perbedaan antara HTML dan driver database yang digunakan (Tjandra & Pickerling, 2015). Dapat dikatakan bahwa Pengujian kompatibilitas berguna dalam menentukan satu set lingkungan yang diharapkan untuk menjalankan perangkat lunak pengembangan. Semakin beragam lingkungan yang dapat dijalankan pada perangkat lunak, semakin baik kompatibilitasnya.

2.13.3 User Acceptance Testing

UAT adalah proses pengujian pengguna, yang dirancang untuk menghasilkan dokumen untuk digunakan sebagai bukti bahwa perangkat lunak yang dikembangkan telah diterima oleh pengguna. Uji Penerimaan Pengguna atau dikenal sebagai pengujian beta ini dilakukan oleh pengguna sistem untuk memastikan bahwa fungsi dalam sistem bekerja dengan baik dan memenuhi kebutuhan pengguna, akan dilakukan pemeriksaan ada atau tidaknya kesesuaian item-item yang ada dalam dokumen requirement dengan sistem. Sehingga didapati semua item yang telah ada telah dapat memenuhi kebutuhan penggunaannya (Nathan & Scobell, 2012).

UAT adalah langkah terakhir dari pengembangan aplikasi, diikuti oleh perbaikan peningkatan aplikasi, dengan literatur dan dokumentasi pengembangan. Peran UAT dalam siklus pengembangan aplikasi adalah sangat penting. Dalam implementasinya, pengguna sebagai objek uji UAT dihadapkan

pada serangkaian masalah sesuai dengan format dokumen uji UAT . Isi dokumen uji UAT ini, tentang fungsi dan menu dari aplikasi yang diberikan, serta urutan langkah-langkahnya, hasil yang diharapkan dari , apakah pengujian pengguna telah dilakukan, , dan komentar yang diperlukan . Semakin banyak pengguna, semakin didapati hasil yang valid (Pratama, 2018).

Berikut ini, Proses dalam melakukan pengujian *user acceptance testing* (UAT) yang dilakukan penguji, antara lain (Supriatna, 2019):

1. Menyusun Rencana UAT

Pertama-tama akan diuraikan strategi yang digunakan untuk memverifikasi dan memastikan sistem telah memenuhi persyaratan yang sebelumnya sudah dirancang

2. Desain UAT

Dilakukan identifikasi skenario pengujian dan kasus uji, sehubungan dengan proses-peoses yang dilakukan, dan persiapan data uji

3. Eksekusi Uji UAT

Akan dijalankan kasus uji yang akan kemudian akan dubuat keputusan penerimaan

4. Konfirmasi Tujuan Bisnis yang dipenuhi

Penguji mengirimkan tanda surat setelah dilakukan pengujian UAT, dengan pertimbangan tidak ada cacat , dan proses bisnis dapat bekerja dengan baik

5. Sign off

Menerima hasil pengujian , pengguna dapat menerima sistem dan menunjukkan bahwa sistem yang dibuat telah memenuhi persyaratan pengguna

Untuk melakukan perhitungan pada pengujian ini, digunakan skala likert. Dengan memberikan pertanyaan berupa kuesioner kepadabeberapa responden mengenai sistem yang dikembangkan. Dari setiap jawaban kuesioner tersebut memiliki bobot nilai, menggunakan skala likert yang dijelaskan pada Tabel 2.1

Jawaban	Bobot
SS = Sangat Setuju	5
S = Setuju	4
N = Netral	3
TS = Tidak Setuju	2
STS = Sangat Tidak Setuju	1

Tabel 2. 1 Bobot Penilaian Skala Likert

Sumber : (Hanggara & Ratnawati, 2020)

Langkah selanjutnya, dilakukan perhitungan nilai total dari jawaban responden, yang dapat diperoleh dengan mengalikan jumlah nilai dari responden (n) dengan bobot nilai dengan persamaan seperti berikut :

$$\text{Total Nilai} = (nx5) + (nx4) + (nx3) + (nx2) + (nx1) \quad (2.1)$$

Selanjutnya, akan dilakukan perhitungan nilai Y dengan cara jumlah skala penilaian (N1) dikali dengan jumlah responden (n) dan jumlah pertanyaan (U), dengan persamaan sebagai berikut :

$$Y = N1 \times n \times U \quad (2.2)$$

Kemudian setelah didapat nilai Y, akan ditentukan presentase nilai akhir dengan menggunakan rumus index :

$$\text{Rumus Index} = (\text{Total Nilai} / Y) \times 100\% \quad (2.3)$$

Persentase nilai akhir akan menunjukkan tingkat persentase sistem yang diterima oleh pengguna. Persentase nilai akhir akan dikategorikan berdasarkan interval tertentu untuk memahami hasil skala penilaian. Interval yang digunakan yaitu 20 yang didapatkan dari pembagian nilai 100 dengan jumlah skor skala likert dengan 5 skala. Kualifikasi interpretasi skor yang diperoleh pada pengujian *User Acceptance Test (UAT)* berdasarkan interval seperti yang dijelaskan Tabel 2.2.

Tabel 2. 2 Kualifikasi Interpretasi Skor Berdasarkan Interval

Interval Presentase Pencapaian	Kualifikasi
0% - 19.99%	Sangat Tidak Setuju
20% - 39.99%	Tidak Setuju
40% - 59.99%	Netral
60% - 79.99%	Setuju
80% - 100%	Sangat Setuju

Sumber : (Hanggara & Ratnawati, 2020)

BAB 3 METODOLOGI

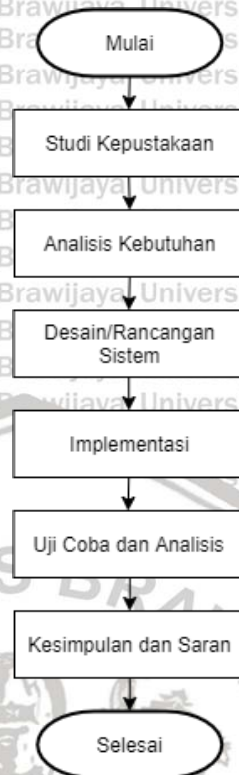
Pada bab metodologi ini berisi uraian tentang langkah-langkah yang dilakukan secara sistematis dalam pengerjaan sistem yang akan diimplementasikan untuk memecahkan permasalahan yang dibahas pada skripsi ini.

3.1 Tipe Penelitian

Jenis atau tipe penelitian yang digunakan dalam penelitian ini adalah implementatif pengembangan. Tipe penelitian implementatif pengembangan berfokus pada penyusunan desain atau pembangunan solusi teknologi yang berupa sebuah produk/artefak utama utuh perangkat lunak atau perangkat keras pengelola informasi yang tepat untuk diterapkan pada sebuah organisasi. dari sesuatu yang belum ada menjadi ada dengan menerapkan prinsip-prinsip rekayasa secara utuh, yang meliputi analisis, perancangan, konstruksi dan pengujian. Hal ini akan menghasilkan suatu produk sebagai solusi dari permasalahan yang diangkat dalam penelitian.

3.2 Strategi Penelitian

Penelitian ini menggunakan metode *waterfall*, yang terdiri dari 6 tahapan seperti pada gambar dibawah. Tahap pertama dilakukan studi kepustakaan untuk mengidentifikasi teori dan penelitian yang sudah ada sebelumnya yang berkaitan dengan topik. Selanjutnya, dilakukan analisis kebutuhan yang akan mengidentifikasi kebutuhan fungsional dan non fungsional sistem. Tahap ketiga yaitu membuat perancangan sistem dengan menggambarkan alur sistem, diagram UML dan juga desain antar muka sistem. Keempat, setelah dilakukan perancangan selanjutnya seluruh rancangan akan diimplementasikan ke dalam kode program, pada penelitian ini akan menggunakan *framework* Codeigniter. Kemudian setelah tahap implementasi selesai, dilakukan uji coba sistem menggunakan metode *blackbox* dan *compability testing* yang akan dilakukan analisis hingga sistem siap untuk digunakan. Dan pada tahap terakhir akan dituliskan hasil kesimpulan dari penelitian dan juga diberikan saran untuk pengembangan selanjutnya.



Gambar 3.1 Diagram alir metodologi penelitian

Sumber : (Penulis, 2021)

Alasan Penulis memilih menggunakan metode waterfall dalam pembangunan sistem dikarenakan, proses pengembangannya menggunakan fase *one by one* atau yang dilakukan berdasarkan urutan yang diharapkan dapat meminimalisir kesalahan yang mungkin terjadi, juga pengembangan sistem akan lebih tertata, karena setiap tahapan harus diselesaikan sebelum dapat melanjutkan ke tahapan berikutnya, dan dengan menggunakan metode waterfall sistem dapat dikembangkan sesuai apa yang diinginkan pemilik bisnis.

Dalam pengembangan sistem e-commerce ini akan terintegrasi dengan *payment gateway* midtrans, yaitu dengan pertimbangan, midtrans memiliki 20 metode pembayaran mulai dari bank hingga mini market. Keamanan data perusahaan yang terjamin aman, dan midtrans dapat mendeteksi apabila ada transaksi yang mencurigakan maka transaksi tersebut ditahan dan dilakukan penelusuran melalui telepon oleh pihak midtrans.

3.2.1 Studi Kepustakaan

Tahap studi kepustakaan merupakan tahap mengumpulkan seluruh informasi yang relevan dengan topik bahasan baik dari jurnal, buku, karya ilmiah, internet, dan sumber-sumber lain untuk dipelajari dan dipahami. Teori-teori tersebut nantinya akan dijadikan sebagai acuan dasar penelitian

serta dapat mempermudah untuk menyelesaikan permasalahan yang akan dibahas dalam skripsi ini.

3.2.2 Analisis Kebutuhan

Dalam tahap Analisis Kebutuhan ini dilakukan identifikasi sehingga dapat diketahui apa saja yang menjadi kebutuhan sistem, identifikasi pengguna dan kebutuhan pengguna sistem. Termasuk pada bab ini dijabarkan penjelasan dari seluruh kebutuhan fungsional, kebutuhan non-fungsional Sehingga dapat memahami software yang dibutuhkan oleh pengguna dan batasan-batasan sistem, dan alur aktivitas pada setiap fungsi. Informasi ini dapat diperoleh dari wawancara langsung maupun survei kepada pemilik bisnis, dan pengguna sistem.

3.2.3 Perancangan Sistem

Pada tahap perancangan sistem ini dilakukan desain yang diterjemahkan dari langkah analisis kebutuhan, seluruh kebutuhan yang sudah diidentifikasi akan dirancang sebelum kemudian dilakukan implementasi. Proses ini berfokus pada struktur data, arsitektur perangkat lunak, representasi antarmuka, dan detail algoritma program. Perancangan berorientasi objek meliputi beberapa tahapan yaitu, sequence diagram, class diagram, user interface, dan perancangan model data fisik.

3.2.4 Implementasi

Pada tahap implementasi ini dilakukan penerjemahan seluruh rancangan desain sistem yang sudah didefinisikan sebelumnya, ke dalam bentuk yang dapat dimengerti oleh mesin, dengan menggunakan kode kode bahasa pemrograman oleh developer. Implementasi pada pengembangan website *E-commerce* Butik Kebaya Iffah menggunakan Payment Gateway Midtrans, sebagai berikut :

1. Implementasi user interface, dilakukan implementasi tampilan berdasarkan perancangan user interface menggunakan HTML, CSS dan Javascript dengan framework Codeigniter.
2. Implementasi database, dilakukan implementasi rancangan database menggunakan MySQL sebagai tools database management system dan menggunakan bahasa SQL.
3. Implementasi kode program, dilakukan implementasi dari perancangan sistem menggunakan bahasa PHP dengan framework Codeigniter.

3.2.4.1 Implementasi Payment Gateway

Pada bagian implementasi pada penelitian ini dilakukan pengintegrasian program dengan memanfaatkan *payment gateway* menggunakan API *sandbox* salah satu layanan *payment gateway* midtrans yang nantinya diharapkan dapat memudahkan pada proses pembayaran pada *Website E-commerce* Butik Kebaya

liffah lebih cepat, kemudahan pilihan pembayaran, dan aman untuk pelanggan butik.

3.2.5 Pengujian

Setelah melakukan implementasi, tahap selanjutnya yang dilakukan adalah tahap Pengujian. Pada tahap ini dilakukan uji coba terhadap sistem yang telah selesai diimplementasi agar dapat digunakan tanpa menemukan kendala-kendala apapun. Adapun ujicoba yang akan dilakukan dengan menggunakan metode black box testing untuk menguji fungsionalitas sistem, Pengujian kompatibilitas menggunakan tools penunjang menggunakan SortSite untuk memastikan sistem dapat berjalan pada lingkungan yang berbeda, dan juga dilakukan *User Acceptance Test* (UAT) untuk memastikan kebutuhan user berjalan sesuai. Dan dalam pengujian ini akan dilakukan perbaikan apabila ditemukan kesalahan pada sistem.

3.2.6 Penutup

Setelah semua tahapan pada metodologi penelitian telah selesai maka diambil Kesimpulan. Kesimpulan tersebut diambil dari hasil analisis dari sistem yang telah selesai dibuat dan merupakan jawaban dari rumusan masalah yang sudah dirumuskan sebelumnya. Selanjutnya juga diberikan saran sebagai bahan masukan agar dapat memperbaiki kekurangan dari pengembangan sistem maupun cara penulisan bagi penelitian selanjutnya.

BAB 4 ANALISIS KEBUTUHAN DAN PERANCANGAN SISTEM

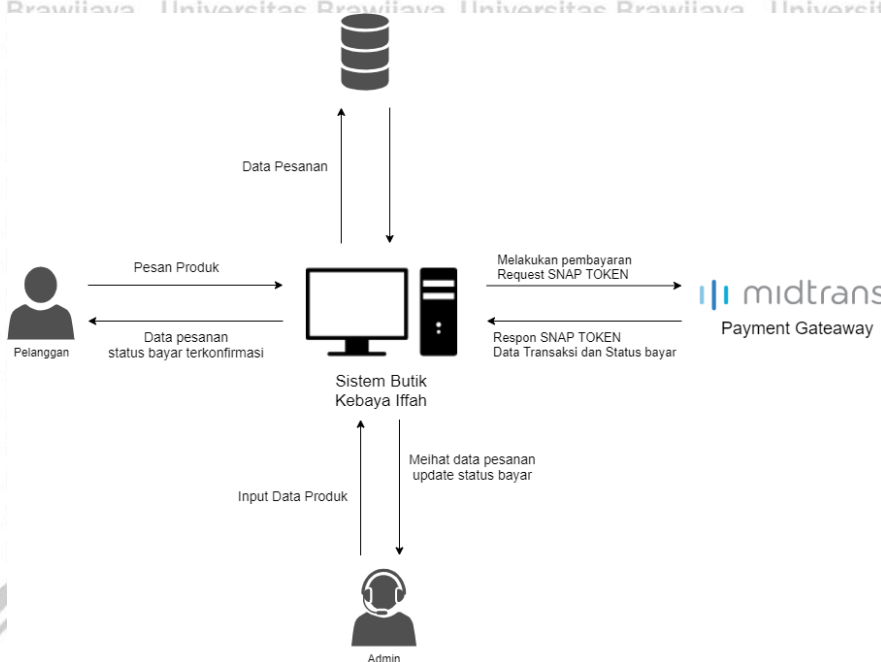
Dalam tahap analisis kebutuhan dan perancangan sistem ini akan dilakukan pembahasan mengenai deskripsi umum sistem, arsitektur sistem, pemodelan proses bisnis dan identifikasi kebutuhan sistem yaitu dengan cara mengidentifikasi kebutuhan informasi. Dari hasil identifikasi tersebut akan diambil hasil analisis untuk mengetahui perilaku sistem dan aktivitas apa saja yang ada pada sistem. Serta pada tahap ini akan dilakukan identifikasi aktor sesuai kebutuhan sistem, perencanaan kebutuhan fungsional dan kebutuhan non-fungsional kemudian digambarkan ke dalam *diagram UML* untuk merancang sistem website e-commerce butik kebaya iffah.

4.1 Deskripsi Umum Sistem

Butik kebaya iffah E-commerce adalah situs E-commerce yang berfungsi untuk menjadi platform website jual beli secara online produk produk yang dijual oleh toko butik kebaya iffah, selain itu pada E-commerce ini seorang pelanggan juga dapat melakukan request custom produk kepada butik. Dan pada sistem ini seorang pelanggan dapat membeli produk dengan melakukan pembayaran melalui payment gateway sehingga memberikan kemudahan bertransaksi.

User dalam Sistem E-commerce butik kebaya iffah memiliki kewenangan berbeda, yaitu Administrator (ADM), Pelanggan (PLNG) dan Midtrans (MID). Administrator (ADM) adalah satu orang yang berwenang untuk memanajemen keseluruhan dari website, mulai dari mengelola produk, mengelola kategori, mengelola konfigurasi web, manajemen pengguna, melihat daftar pesana dan custom order serta melakukan updtae status pembayaran, dan mencetak laporan. Sedangkan pelanggan adalah orang yang dapat melihat produk produk yang dijual pada website dan akan melakukan transaksi pada website. Seorang pelanggan diharuskan untuk melakukan login terlebih dahulu sebelum melakukan pemesanan produk, yang kemudian akan melakukan pembayaran secara online. Dan satu aktor sistem yaitu midtrans yang menjadi payment gateway untuk melakukan pembayaran dan melakukan update status pembayaran.

4.2 Arsitektur Website E-commerce Butik Kebaya Iffah



Gambar 4. 1 Arsitektur Website E-commerce Butik Kebaya Iffah

Sumber : (Penulis,2021)

Pada gambar 4.1 dijelaskan gambaran arsitektur Sistem website *e-commerce* butik kebaya iffah. Terdapat langkah pertama yang dimulai saat pelanggan ingin melakukan pemesanan melalui *e-commerce* butik kebaya iffah, yang selanjutnya saat melakukan checkout data pembelian akan tersimpan pada database setelah itu sistem akan merequest SNAP token pada payment gateway midtrans, setelah midtrans mengirimkan respon SNAP token akan ditampilkan halaman pembayaran, yang kemudian user melakukan pembayaran. Setelah terkonfirmasi midtrans akan mengirimkan data transaksi dan status bayar yang akan diupdate pada database sistem butik kebaya iffah. Dan admin dapat melihat detail transaksi, menginputkan produk juga status pembayaran yang kemudian dilanjutkan untuk mengkonfirmasi pesanan dan mengirimkannya. Setelah data terupdate pelanggan juga dapat melihat data pesanan, status bayar yang sudah terkonfirmasi dan menunggu barang dikirimkan.

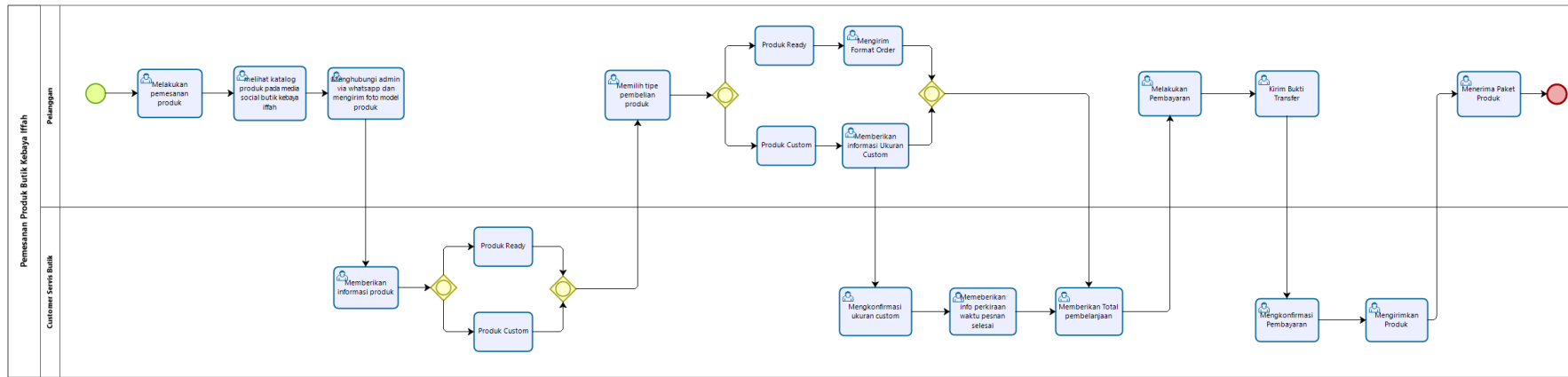
4.3 Pemodelan Proses Bisnis

Pada bagian ini akan dilakukan identifikasi dan melakukan pemodelan proses bisnis. Sehingga didapati dua pemodelan proses bisnis, yaitu tentang proses bisnis yang sedang berlangsung (model As-Is), dan yang kedua sehingga dapat menentukan area untuk perbaikan yang dapat dilakukan di masa depan (model To-Be) dan pada akhirnya meningkatkan kinerja bisnis proses saat adanya sistem website *e-commerce* butik kebaya iffah.

4.1.1 Pemodelan Proses Bisnis As-Is

Berikut adalah pemodelan proses bisnis as-is yang dapat digambarkan menggunakan Business Process Modelling Notation (BPMN). Proses didapatkan setelah melakukan wawancara dengan pemilik bisnis yaitu Ibu Iffah selaku owner. Saat ini Terdapat dua metode proses pemesanan produk butik kebaya iffah yaitu pertama dapat *online* melalui whatsapp yang akan dijelaskan pada gambar 4.2, dimana pelanggan melihat katalogg produk pada *media social instagram* lalu mengirimkan ke whatsapp, kemudian mengirimkan format order dan kemudian admin akan meberikan total pemesanan dan memberikan nomor rekening untuk dilakukan pembayaran, setelah melakukan pembayaran pelanggan mengirimkan bukti transfer dan setelah admin mengkonfirmasi pembayaran maka produk baru akan dikirimkan. dan yang kedua yaitu dapat dilakukan dengan *offline* datang langsung ke lokasi butik yang dijelaskan pada gamabar 4.3. dimana pelanggan datang langsung ke butik dan melakukan pembelian dan pembayaran langsung pada butik, serta melakukan pengukuran secara *offline*.

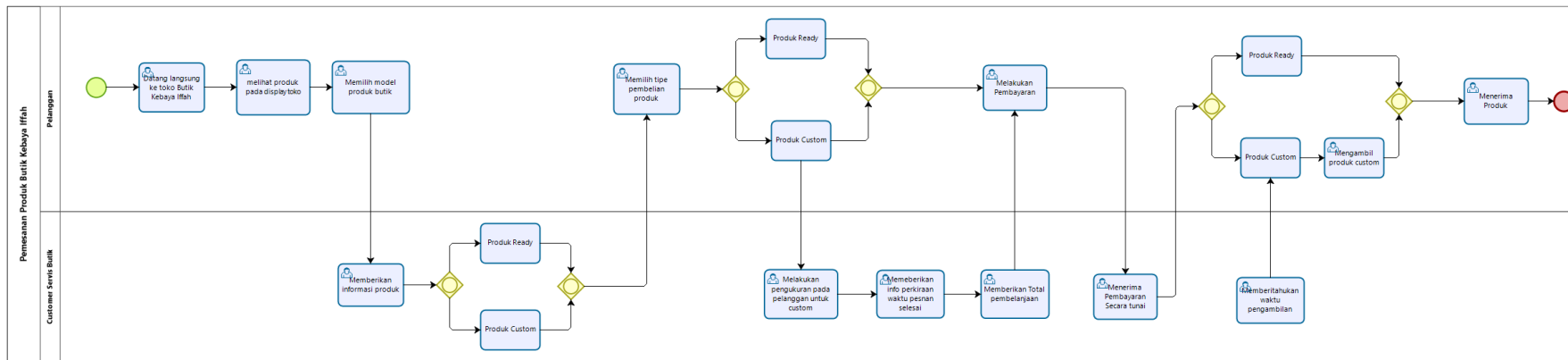




Powered by bizagi Model

Gambar 4. 2 Proses Bisnis Saat Ini Pada Butik Kebaya Iffah Pemesanan Melalui Whatsapp

Sumber : (Penulis, 2021)



Powered by bizagi Model

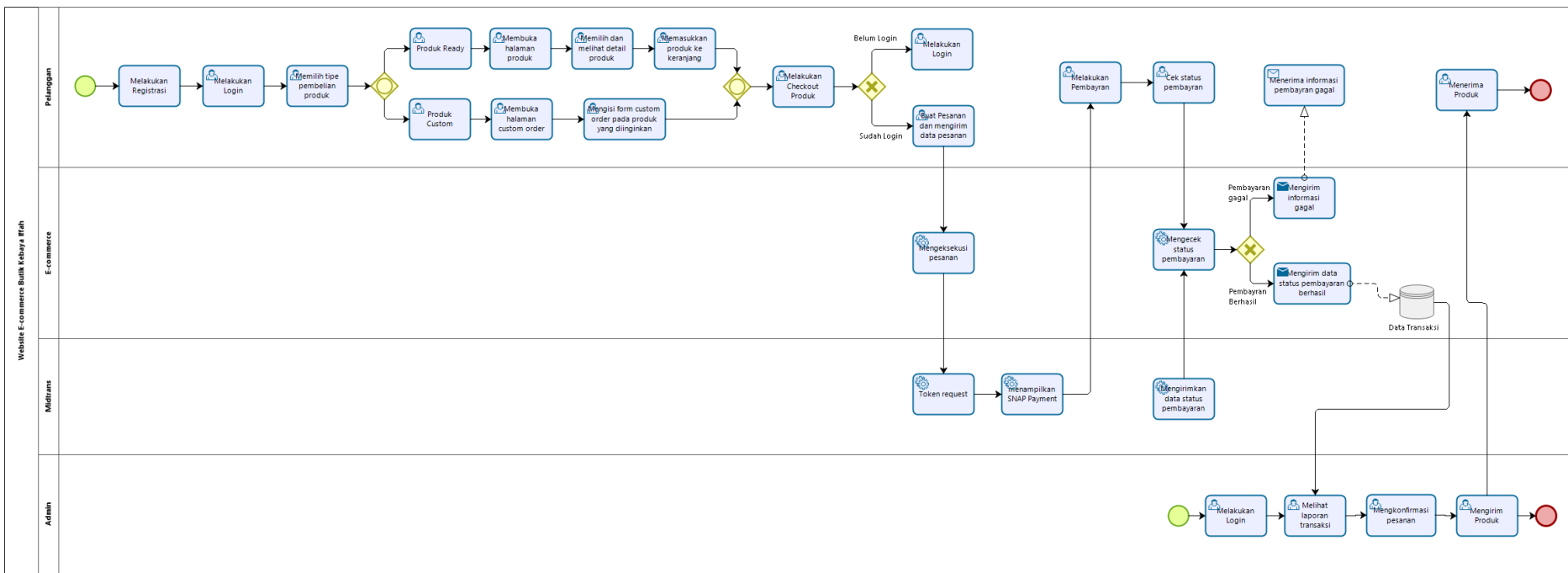
Gambar 4. 3 Proses Bisnis Saat Ini Pada Butik Kebaya Iffah Datang Langsung ke Toko

Sumber : (Penulis, 2021)

4.1.2 Pemodelan Proses Bisnis *To-Be*

Berikut adalah pemodelan proses bisnis to-be yang merupakan proses bisnis yang diusulkan oleh penulis dapat digambarkan menggunakan Business Process Modelling Notation (BPMN). Proses bisnis ini merupakan usulan dari penulis yang dibuat dengan tujuan untuk mengembangkan proses bisnis yang sudah ada sebelumnya. Beberapa aktivitas pada proses bisnis sebelumnya yang masih dilakukan secara manual dan terbatas penggunaannya dirasa penulis bisa dikembangkan dengan memperbaiki aktivitas manual menjadi sistem yang diharapkan akan sangat membantu pemasaran produk, dan mempermudah pelanggan.





Gambar 4. 4 Proses Bisnis Usulan Pembelian Produk Pada Butik Kebaya Iffah

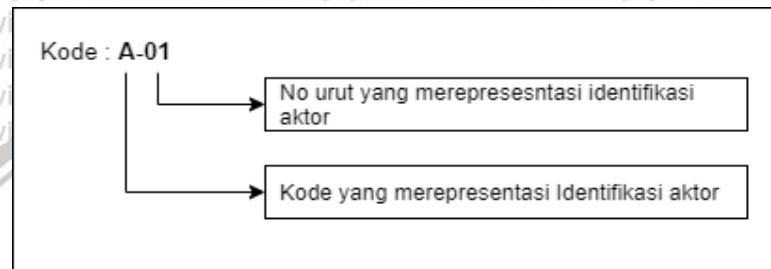
Sumber : (Penulis, 2021)

4.4 Analisis Kebutuhan

Pada bagian ini akan dijelaskan mengenai apa saja kebutuhan yang dibutuhkan oleh sistem, meliputi identifikasi aktor, kebutuhan fungsional, dan kebutuhan non-fungsional.

4.4.1 Identifikasi Aktor

Pada bagian ini akan dijelaskan nama nama aktor yang ada pada sistem. Dimana setiap aktor memiliki penomoran masing masing yang dijelaskan pada Gambar 4.5. dan gambaran karakteristik tiap aktor yang berbeda beda, akan dipaparkan pada Tabel 4.1.



Gambar 4. 5 Penomoran Identifikasi Aktor

Sumber : (Penulis,2021)

Tabel 4. 1 Identifikasi Aktor

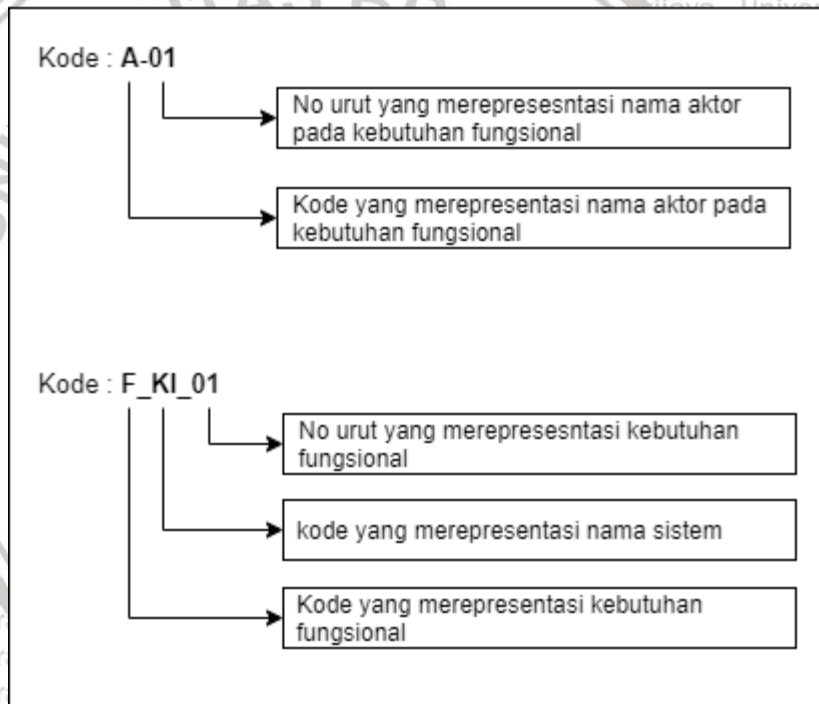
No.	Kode Aktor	Identifikasi Aktor	Karakteristik
1.	A-01	Pelanggan (PLNG)	Seseorang yang menggunakan aplikasi ecommerce Butik Kebaya Iffah untuk melakukan transaksi pembelian produk butik.
2.	A-02	Administrator (ADM)	Seorang yang memiliki hak akses sebagai admin. Yang Memiliki kewenangan untuk memanajemen data pada aplikasi ecommerce Butik Kebaya Iffah yaitu mengelola data produk, melihat data pesanan dan custom order, melakukan update status pembayaran, manajemen pengguna, serta dapat mencetak detail pesanan.

3.	A-03	Midtrans	Sistem yang akan digunakan pada fungsional pembayaran dan update status, dimana midtrans memiliki peranan dalam keberhasilan operasi dari sistem butik kebaya iffah.
----	------	----------	--

Sumber : (Penulis, 2021)

4.4.2 Analisis Kebutuhan Fungsional

Analisis Kebutuhan fungsional bertujuan untuk menganalisis fungsi-fungsi apa saja yang harus dilakukan oleh sistem yang dikembangkan. Pada sistem butik kebaya iffah, kebutuhan fungsional akan dibagi berdasarkan aktor yang ada dan penomoran kebutuhan fungsional akan dijelaskan pada gambar 4.6. Untuk kebutuhan fungsional pada sistem ini pada aktor Pelanggan dijelaskan pada Tabel 4.2. dan untuk kebutuhan fungsional pada aktor admin dijelaskan pada Tabel 4.3.



Gambar 4. 6 Penomoran Kebutuhan Fungsional

Sumber : (Penulis,2021)

Tabel 4. 2 Kebutuhan Fungsional Pelanggan

No.	[A-01] Pelanggan		
	Kode Fungsi	Nama Fungsi	Deskripsi
1.	F_KI_01	Melakukan Registrasi	Pelanggan diharuskan untuk melakukan registrasi data diri pelanggan untuk dapat melakukan transaksi pada sistem.
2.	F_KI_02	Login	Pelanggan dapat masuk ke sistem menggunakan <i>email</i> dan <i>password</i> yang telah terdaftar pada sistem.
3.	F_KI_03	Logout	Pelanggan dapat keluar dari sistem.
4.	F_KI_04	Melihat Daftar Produk	Pelanggan dapat melihat daftar produk yang dijual pada sistem, dan dapat terkategori sesuai kategori yang dipilih.
5.	F_KI_05	Mengelola Keranjang Belanja	Pelanggan dapat menambahkan produk ke dalam keranjang belanja dengan menekan tombol 'Add to cart'. Selanjutnya, pada halaman keranjang belanja, Pelanggan dapat melihat daftar produk yang sudah dimasukkan ke dalam keranjang, mengupdate jumlah produk, maupun menghapus produk dari keranjang sebelum melakukan transaksi.
6.	F_KI_06	Melakukan Custom Order	Pelanggan dapat melakukan custom detail pada 'produk custom' yang dipilih, dengan menginputkan atribut-atribut custom kepada butik.
7.	F_KI_07	Melakukan	Pelanggan melakukan transaksi

		<i>Checkout</i>	pemesanan produk yang telah dipilih dari daftar keranjang belanja.
8.	F_KI_08	Melakukan Pembayaran	Pelanggan melakukan pembayaran dengan melewati payment gateway midtrans.
9.	F_KI_09	Melihat Riwayat Transaksi	Pelanggan dapat melihat riwayat transaksi beserta detail pembelian produk, dan dapat melihat status pembayaran produk.

Sumber : (Penulis,2021)

Tabel 4. 3 Kebutuhan Fungsional Admin

No.	[A-02] Admin		
	Kode Fungsi	Nama Fungsi	Deskripsi
1.	F_KI_01	<i>Login</i>	Seorang Admin dapat masuk ke sistem menggunakan <i>username</i> dan <i>password</i> yang telah terdaftar pada sistem.
2.	F_KI_02	<i>Logout</i>	Admin dapat keluar dari sistem.
3.	F_KI_03	Melihat <i>Dashboard</i>	Admin dapat melihat dashboard pada awal halaman admin yang berisi rangkuman dari sistem <i>e-commerce</i> dan juga total pendapatan penjualan.
4.	F_KI_04	Mengelola Produk	Admin dapat mengelola data produk, meliputi melakukan input data produk baru, melihat tabel daftar produk yang telah ditambahkan, melakukan perubahan data produk, dan menghapus data produk yang ada dalam tabel daftar produk.

5.	F_KI_05	Melihat Data Pesanan	Admin dapat melihat daftar pesanan dari pelanggan pada sub menu laporan transaksi, dan dapat melihat detail pesanan.
6.	F_KI_06	Melihat Data Custom Order	Admin dapat melihat daftar custom order dari pelanggan pada sub menu laporan custom, dan dapat melihat detail pesanan.
7.	F_KI_07	Update Status Bayar	Admin dapat melakukan update status pembayaran pesanan.
8.	F_KI_08	Mengelola Admin	Pengelolaan data Admin sistem.
9.	F_KI_09	Mencetak Detail Pesanan	Admin dapat melihat detail pesanan dan dapat mengunduh laporan untuk dicetak.

Sumber : (Penulis,2021)

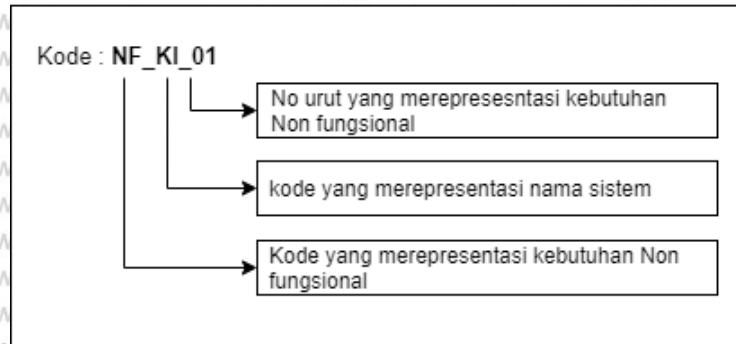
Tabel 4. 4 Kebutuhan Fungsional Midtrans

No.	[A-03] Midtrans		
	Kode Fungsi	Nama Fungsi	Deskripsi
1.	F_KI_01	Melakukan Pembayaran	Midtrans akan merespon token API dan menampilkan SNAP payment.
2.	F_KI_02	Update Status	Midtrans melakukan update status bayar.

Sumber : (Penulis,2021)

4.4.3 Analisis Kebutuhan Non Fungsional

Kebutuhan Non Fungsional adalah kebutuhan yang menitik beratkan pada properti perilaku yang dimiliki oleh sistem. Kebutuhan non fungsional pada sistem ini dijelaskan pada Tabel 4.5. Dan diberikan penomoran mengenai kebutuhan Non-fungsional pada gambar 4.7.



Gambar 4. 7 Penomoran Kebutuhan Non-Fungsional

Sumber : (Penulis,2021)

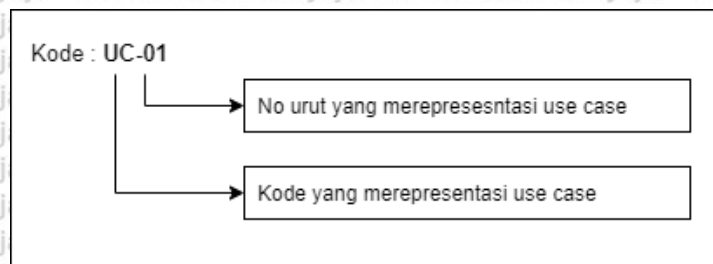
Tabel 4. 5 Kebutuhan Non Fungsional

No.	Kode Fungsi	Nama Fungsi	Deskripsi
1.	NF_KI_01	<i>Compatibility Browser</i>	Sistem dapat berjalan pada aplikasi <i>web browser</i> yang berbeda

Sumber : (Penulis,2021)

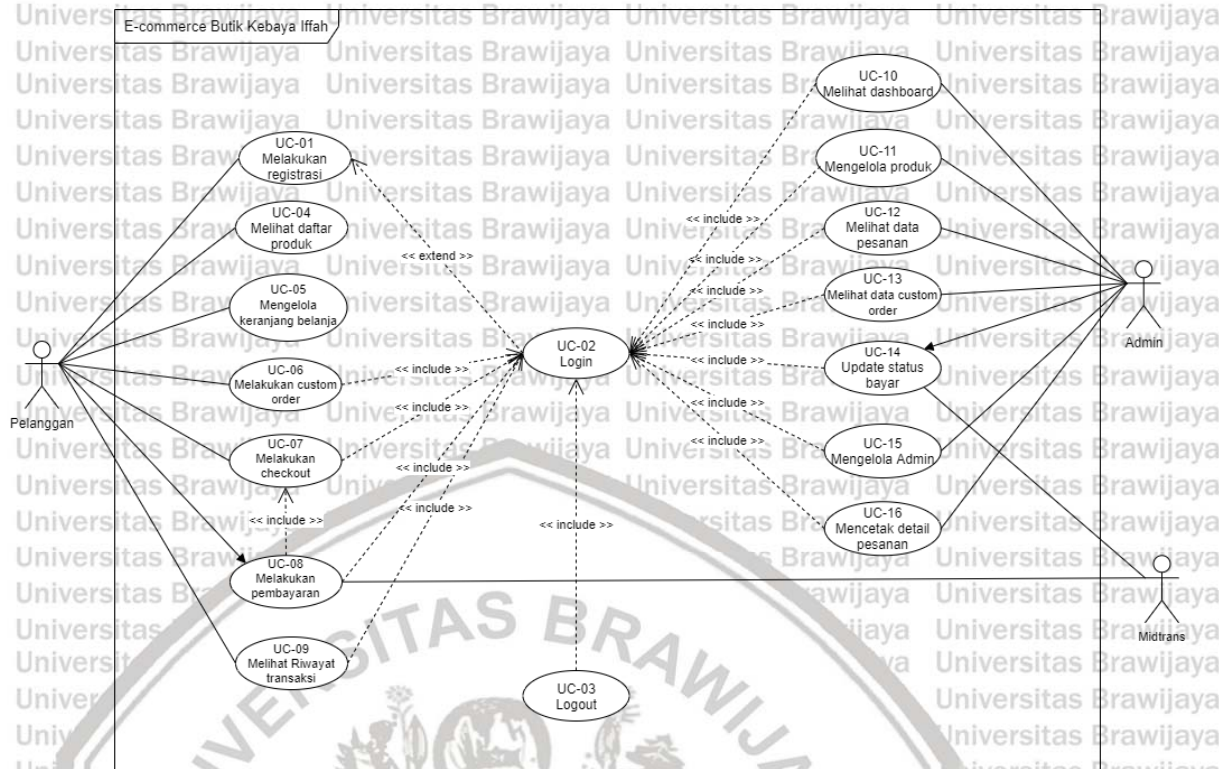
4.4.4 Use Case Diagram

Use case diagram merupakan gambaran scenario dari interaksi antara pengguna dengan sistem. Didalam *use case* ini akan diketahui fungsi-fungsi apa saja yang berada pada sistem yang dibuat (Maulana & Rachmawati, 2017). Tiap-tiap use case akan diberikan penomoran seperti yang dijelaskan pada gambar 4.8. Diagram Use Case dari E-commerce butik kebaya iffah digambarkan seperti pada Gambar 4.9



Gambar 4. 8 Penomoran Diagram Use Case

Sumber : (Penulis,2021)



Gambar 4. 9 Use Case Diagram E-commerce Butik Kebaya Iffah

Sumber : (Penulis,2021)

4.5 Skenario Use Case

Pada bagian ini terdapat tabel-tabel yang menjelaskan skenario dari setiap kebutuhan yang telah dipaparkan pada analisis kebutuhan fungsional dan non fungsional.

Tabel 4. 6 Skenario Use Case Melakukan Registrasi

Nama Use Case	Melakukan Registrasi
Kode Use Case	UC-01
Aktor	Pelanggan
Tujuan	Aktor dapat melakukan registrasi agar dapat masuk ke sistem menggunakan <i>email</i> dan <i>password</i> yang didaftarkan.
Precondition	Aktor telah berada di halaman utama sistem <i>e-commerce</i> butik kebaya iffah

Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih sub menu Registrasi pada icon user 2. Sistem menampilkan halaman form registrasi pelanggan 3. Aktor memasukkan data diri pada form registrasi pelanggan 4. Aktor menekan tombol 'Registrasi' 5. Sistem memverifikasi email 6. Sistem mengarahkan pada halaman sukses registrasi, dan menampilkan pesan registrasi berhasil
Postcondition	Aktor berhasil melakukan registrasi pelanggan
Alternative Flow	<ol style="list-style-type: none"> 1. Jika aktor tidak melakukan input pada salah satu atau beberapa data pada form, maka sistem akan memberikan peringatan untuk mengisi form yang kosong 2. Jika aktor menginputkan email yang sama dengan yang sudah ada sebelumnya pada sistem, maka sistem memberi peringatan untuk mengganti dengan email yang lain

Sumber: (Penulis,2021)

Tabel 4. 7 Skenario Use Case Login Pelanggan

Nama Use Case	<i>Login</i>
Kode Use Case	UC-02
Aktor	Pelanggan
Tujuan	Aktor masuk ke sistem menggunakan <i>email</i> dan <i>password</i> yang telah terdaftar pada sistem.
Precondition	Aktor telah berada di halaman utama sistem <i>e-commerce</i> butik kebaya iffah

Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih sub menu <i>Login</i> pada icon user 2. Sistem menampilkan halaman form <i>login</i> 3. Aktor menginputkan <i>email</i> dan <i>password</i> 4. Aktor menekan tombol 'Login' 5. Sistem memverifikasi <i>email</i> dan <i>password</i> yang diinputkan oleh aktor 6. Aktor berhasil masuk ke sistem pada halaman <i>dashboard</i> pelanggan, dan menampilkan pesan selamat datang.
Postcondition	Aktor berhasil masuk ke dalam sistem dan tampil halaman <i>dashboard</i> pelanggan
Alternative Flow	<ol style="list-style-type: none"> 1. Jika aktor memasukkan <i>email</i> atau <i>password</i> yang salah atau tidak terdaftar dalam sistem, maka sistem akan mengarahkan aktor pada halaman <i>login</i> kembali 2. Jika aktor hanya mengisi salah satu form diantara <i>username</i> dan <i>password</i>, maka sistem akan memberikan peringatan pada form yang kosong

Sumber : (Penulis,2021)

Tabel 4. 8 Skenario Use Case Logout

Nama Use Case	Logout
Kode Use Case	UC-03
Aktor	pelanggan
Tujuan	Aktor dapat logout dari sistem
Precondition	Aktor telah masuk di halaman sistem e-commerce
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol 'Logout' pada topbar sistem, untuk keluar dari sistem 2. Sistem menghapus authentication pelanggan. 3. Sistem menampilkan halaman login.
Postcondition	Aktor berhasil Logout dari sistem
Alternative Flow	

Tabel 4. 9 Skenario Use Case melihat daftar produk

Nama Use Case	Melihat daftar produk
Kode Use Case	UC_04
Aktor	Pelanggan
Tujuan	Aktor dapat melihat daftar produk butik kebaya iffah
Precondition	Aktor telah berada di halaman utama sistem <i>e-commerce</i> butik kebaya iffah
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu 'Shop' 2. Sistem menampilkan halaman sistem <i>e-commerce</i> pada menu shop 3. Aktor melihat seluruh daftar produk butik kebaya iffah dari semua kategori 4. Aktor dapat memilih kategori yang ingin ditampilkan daftar produknya, dengan menekan tombol kategori yang diinginkan 5. Sistem akan menampilkan daftar produk sesuai kategori yang dipilih
Postcondition	Aktor berhasil melihat produk yang terdaftar
Alternative Flow	-

Sumber : (Penulis,2021)

Tabel 4. 10 Skenario Use Case Mengelola keranjang belanja

Nama Use Case	Mengelola keranjang belanja
Kode Kebutuhan	UC_05
Aktor	pelanggan

Tujuan	Aktor dapat mengelola keranjang belanja
Precondition	Aktor telah berada di halaman utama sistem <i>e-commerce</i> butik kebaya iffah
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu 'Shop' 2. Sistem menampilkan halaman sistem <i>e-commerce</i> pada menu shop 3. Aktor melihat detail produk yang diinginkan 4. Aktor menekan tombol 'Add to cart' untuk memasukkan produk ke dalam keranjang belanja 5. Sistem menampilkan Pemberitahuan bahwa produk berhasil ditambahkan ke keranjang 6. Aktor menekan icon keranjang, dan klik tombol 'view cart' untuk melihat detail keranjang belanja 7. Sistem menampilkan keranjang belanja 8. Saat melakukan perubahan quantity produk, Selanjutnya aktor dapat menekan tombol 'update cart' 9. Sistem menampilkan update detail keranjang belanja 10. Jika ingin menghapus, Aktor dapat menekan tombol 'Delete' untuk menghapus produk dari keranjang belanja 11. Sistem menampilkan Pemberitahuan bahwa produk yang dipilih berhasil terhapus dari keranjang
Postcondition	Aktor berhasil mengelola keranjang belanja
Alternative Flow	-

Sumber : (Penulis,2021)

Tabel 4. 11 Skenario Use Case Melakukan Custom Order

Nama Use Case	Melakukan <i>custom order</i>
Kode Use Case	UC_06
Aktor	Pelanggan
Tujuan	Aktor dapat melakukan custom order produk
Precondition	Aktor telah berada di halaman utama sistem <i>e-commerce</i> butik kebaya iffah
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu 'Custom' 2. Sistem menampilkan halaman sistem <i>e-commerce</i> pada menu custom 3. Aktor memilih produk yang diinginkan untuk dilakukan custom 4. Sistem menampilkan gambar produk dan form custom order 5. Aktor menginputkan data custom order yang diperlukan pada form yang tersedia, seperti lingkaran dada, pilihan busui, dll. 6. Aktor menekan tombol 'Send Request' 7. Sistem meverifikasi form custom 8. Sistem menampilkan halaman checkout, dengan data keranjang produk custom beserta total harga custom.
Postcondition	Aktor berhasil melakukan custom produk
Alternative Flow	<ol style="list-style-type: none"> 1. Jika aktor tidak melakukan input atau input data tidak sesuai pada salah satu atau beberapa data pada form custom produk, maka sistem akan memberikan peringatan untuk mengisi form yang kosong 2. Jika aktor belum melakukan login, maka tidak dapat melakukan custom order, dan sistem menampilkan halaman login

Sumber : (Penulis,2021)

Tabel 4. 12 Skenario *Use Case* Melakukan *checkout*

Nama Use Case	Melakukan <i>checkout</i>
Kode Use Case	UC_07
Aktor	pelanggan
Tujuan	Aktor dapat Melakukan <i>checkout</i> pesanan
Precondition	Aktor telah berada di halaman utama sistem <i>e-commerce</i> butik kebaya iffah
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu 'Checkout' pada topbar 2. Sistem menampilkan halaman <i>checkout</i> pesanan, berisi detail pesanan dan form pengiriman 3. Aktor mengecek atau menginputkan data pada form pengiriman 4. Aktor menekan tombol 'Buat Pesanan' untuk melakukan transaksi 5. Sistem melakukan cek session pelanggan 6. Sistem akan melanjutkan pada proses pembayaran menggunakan midtrans
Postcondition	Aktor berhasil Melakukan <i>checkout</i>
Alternative Flow	<ol style="list-style-type: none"> 1. Saat menekan tombol 'Buat Pesanan' pada langkah 4, jika aktor belum melakukan login, maka sistem akan mengharuskan login dengan mengarahkan pada halaman login pelanggan.

Sumber : (Penulis,2021)

Tabel 4. 13 Skenario *Use Case* Melakukan Pembayaran

Nama Use Case	Melakukan pembayaran
Kode Use Case	UC_08

Aktor	Pelanggan, Midtrans
Tujuan	Aktor dapat Melakukan pembayaran menggunakan payment gateway midtrans
Precondition	Aktor telah berada di halaman sistem e-commerce pada menu checkout
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol 'Buat Pesanan' untuk melakukan transaksi belanja 2. Sistem akan Request SNAP token api kepada midtrans 3. API Midtrans mengirimkan respon token dan menampilkan Snap paymen halaman pembayaran 4. Aktor menekan tombol 'Lanjut' 5. Sitem menampilkan halaman pilihan pembayaran 6. Aktor memilih metode pembayaran yang tersedia 7. Sistem midtrans menampilkan cara pembayaran dari metode yang dipilih 8. Aktor menekan tombol 'Lihat nomor rekening' 9. Sistem menampilkan informasi rekening 10. Aktor melanjutkan pembayaran 11. Sistem akan cek pembayaran 12. Aktor dapat melihat status pembayaran 13. Sistem menampilkan pembayaran berhasil
Postcondition	Aktor berhasil Melakukan pembayaran menggunakan payment gateway midtrans
Alternative Flow	<ol style="list-style-type: none"> 1. Status pembayaran dapat hangus saat sudah melewati batas waktu, dan akan gagal

Sumber : (Penulis,2021)

Tabel 4. 14 Skenario Use Case Melihat Riwayat transaksi

Nama Use Case	Melihat Riwayat transaksi
Kode Use Case	UC_09
Aktor	pelanggan

Tujuan	Aktor dapat Melihat Riwayat transaksi
Precondition	Aktor telah berada di halaman utama sistem <i>e-commerce</i> butik kebaya iffah
Main Flow	<ol style="list-style-type: none"> 1. Aktor melakukan proses login 2. Sistem menampilkan halaman <i>dashboard</i> pelanggan 3. Aktor memilih submenu 'Riwayat Belanja' pada <i>dashboard</i> 4. Sistem menampilkan daftar riwayat transaksi belanja pelanggan 5. Aktor dapat menekan tombol 'detail' pada transaksi yang dipilih, untuk informasi transaksi lebih lanjut 6. Sistem menampilkan detail transaksi belanja terpilih
Postcondition	Aktor berhasil Melihat Riwayat transaksi
Alternative Flow	-

Sumber : (Penulis,2021)

Tabel 4. 15 Skenario Use Case Login Admin

Nama Use Case	<i>Login Admin</i>
Kode Use Case	UC_02
Aktor	Admin
Tujuan	Aktor masuk ke sistem menggunakan <i>username</i> dan <i>password</i> yang telah tec ccrdaftar pada sistem.
Precondition	Aktor telah berada di halaman <i>login</i> admin

Main Flow	<ol style="list-style-type: none"> 1. Aktor membuka halaman login admin 2. Sistem menampilkan halaman <i>login</i> admin 3. Aktor memasukkan <i>username</i> dan <i>password</i> 4. Aktor menekan tombol 'Login' 5. Sistem memverifikasi <i>username</i> dan <i>password</i> yang diinputkan oleh aktor 6. Sistem menampilkan halaman <i>dashboard</i> admin
Postcondition	Aktor berhasil login pada halaman admin
Alternative Flow	<ol style="list-style-type: none"> 1. Jika aktor memasukkan <i>username</i> atau <i>password</i> yang salah atau tidak terdaftar dalam sistem, maka sistem akan mengarahkan aktor pada halaman <i>login</i> kembali 2. Jika aktor hanya mengisi salah satu form diantara <i>username</i> dan <i>password</i>, maka sistem akan memberikan peringatan pada form yang kosong

Sumber : (Penulis,2021)

Tabel 4. 16 Skenario Use Case Logout Admin

Nama Use Case	Logout Admin
Kode Use Case	UC_03
Aktor	Admin
Tujuan	Aktor dapat logout dari sistem
Precondition	Aktor telah berada pada halaman admin
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol 'Logout' untuk keluar dari sistem 2. Sistem menghapus authentication admin. 3. Sistem menampilkan halaman login.
Postcondition	Aktor berhasil Logout dari sistem
Alternative Flow	

Sumber : (Penulis,2021)

Tabel 4. 17 Skenario Use Case Melihat Dashboard

Nama Use Case	Melihat <i>dashboard</i>
Kode Use Case	UC_10
Aktor	Admin
Tujuan	Aktor dapat melihat halaman <i>dashboard</i> admin
Precondition	Aktor telah berada di halaman admin
Main Flow	<ol style="list-style-type: none"> 1. aktor memilih menu <i>dashboard</i> pada halaman admin 2. sistem menampilkan halaman dashboard admin berisi ringkasan sistem dan total pendapatan.
Postcondition	Aktor berhasil melihat halaman dashboard admin
Alternative Flow	-

Sumber : (Penulis,2021)

Tabel 4. 18 Skenario Use Case Mengelola produk

Nama Use Case	Mengelola Produk
Kode Use Case	UC_11
Aktor	Admin
Tujuan	Aktor dapat melakukan kelola data produk dengan <i>create, read,update dan delete</i>

Precondition	Aktor telah berada pada halaman admin
Main Flow	<ol style="list-style-type: none"> 1. Sistem menampilkan halaman dashboard admin 2. Sistem menampilkan menu sidebar 3. Aktor memilih menu 'Toko' Submenu 'Produk' 4. Sistem menampilkan daftar produk yang telah terdaftar 5. Untuk melakukan tambah produk, Aktor dapat menekan tombol 'Tambah' 6. Sistem menampilkan form tambah produk 7. Aktor memasukkan data produk pada form yang tersedia 8. Aktor menekan tombol 'Simpan' 9. Sistem menampilkan pesan berhasil dan halaman daftar tabel produk 10. Untuk melakukan edit data produk, Aktor dapat memilih tombol 'Edit' pada produk yang ingin diubah datanya 11. Sistem menampilkan form Edit data produk 12. Aktor melakukan edit data produk pada form yang ingin diubah 13. Aktor menekan tombol 'Simpan' 14. Sistem menampilkan daftar produk dengan data produk yang baru saja diedit. 15. Untuk dapat menghapus data produk, Aktor dapat menekan tombol 'Delete' 16. Sistem menampilkan pesan pop up validasi 17. Aktor menekan tombol 'Ya,Hapus' 18. Sistem akan memproses hapus data produk
Postcondition	Aktor berhasil melakukan <i>create,read,update dan delete</i> data produk
Alternative Flow	<ol style="list-style-type: none"> 1. Jika aktor tidak melakukan input pada salah satu atau beberapa data pada form tambah produk, maka sistem akan memberikan peringatan untuk mengisi form yang kosong 2. Jika aktor mengunggah gambar produk dengan tipe file selain tipe (jpg,jpeg,png) maka sistem akan menampilkan peringatan bahwa tipe file yang diunggah tidak sesuai 3. Jika data pada form kode produk telah tersedia di dalam database, maka sistem akan menolak input dari aktor dan memberi peringatan bahwa kode produk tidak bisa dipakai 4. Untuk proses edit,jika aktor tidak melakukan input pada salah satu atau beberapa data pada form edit produk,

	<p>maka sistem akan memberikan peringatan untuk mengisi form yang kosong</p> <p>5. Untuk proses delete, Jika aktor menekan tombol 'Batal' pada <i>pop up</i> saat melakukan penghapusan, maka data tidak jadi terhapus dari daftar produk</p>
--	---

Sumber : (Penulis,2021)

Tabel 4. 19 Skenario Use Case Melihat Data Pesanan

Nama Use Case	Melihat Data Pesanan
Kode Use Case	UC_12
Aktor	admin
Tujuan	Aktor dapat melihat data pesanan
Precondition	Aktor telah berada di halaman admin
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih submenu 'Laporan Transaksi' 2. Sistem menampilkan daftar transaksi produk yang sudah terdaftar pada database 3. Aktor menekan tombol 'Detail' untuk melihat detail pesanan dari pelanggan
Postcondition	Aktor berhasil melihat daftar pesanan
Alternative Flow	-

Sumber : (Penulis,2021)

Tabel 4. 20 Skenario Use Case Melihat Data Custom Order

Nama Use Case	Melihat data <i>custom order</i>
Kode Use case	UC_13
Aktor	admin

Tujuan	Aktor dapat melihat data <i>custom order</i>
Precondition	Aktor telah berada di halaman
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih submenu 'Laporan custom' 2. Sistem menampilkan daftar custom order yang sudah terdaftar pada database 3. Aktor menekan tombol 'Detail' untuk melihat detail atribut atribut custom order yang digunakan untuk custom, dan detail transaksi
Postcondition	Aktor berhasil melihat data custom order
Alternative Flow	-

Sumber : (Penulis,2021)

Tabel 4. 21 Skenario Use Case Update Status Bayar

Nama Use Case	<i>Update Status bayar</i>
Kode Use Case	UC_14
Aktor	Admin, Midtrans
Tujuan	Aktor dapat mengupdate status pembayaran transaksi
Precondition	Aktor telah berada di halaman admin
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih submenu laporan transaksi 2. Sistem menampilkan daftar transaksi 3. Aktor menginputkan kode transaksi 4. Aktor menekan tombol 'Cari' 5. Sistem akan get status bayar dari json midtrans sesuai dengan kode transaksi yang dicari 6. Status pembayaran akan terganti sesuai dengan status pada midtrans dan ditampilkan dengan warna sesuai status.
Postcondition	Aktor berhasil melakukan update status pembayaran
Alternative Flow	<ol style="list-style-type: none"> 1. Jika kode transaksi yang diinputkan tidak ada maka, sistem akan menampilkan halaman kesalahan kode, dan meminta agar aktor menginputkan lagi.

Sumber : (Penulis,2021)

Tabel 4. 22 Skenario Use Case Mengelola Data admin

Nama Use Case	Mengelola data admin
Kode Use Case	UC_15
Aktor	admin
Tujuan	Aktor dapat mengelola data admin
Precondition	Aktor telah berada di halaman admin
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih sub menu admin 2. Sistem menampilkan data pengguna sistem 3. Aktor menekan tombol 'Tambah' untuk menginput pengguna baru 4. Aktor memasukkan data pengguna pada form yang tersedia, dan memilih peran pengguna 5. Aktor menekan tombol 'Simpan' 6. Sistem menampilkan Pemberitahuan bahwa penambahan pengguna berhasil ditambahkan 7. Sistem menampilkan daftar pengguna sistem 8. Jika ingin melakukan perubahan, Aktor dapat menekan tombol 'Edit' 9. Aktor melakukan perubahan pada form 10. Aktor menekan tombol 'Simpan' 11. Sistem menampilkan Pemberitahuan bahwa perubahan data pengguna berhasil 12. Jika ingin menghapus, Aktor dapat menekan tombol 'Delete' untuk menghapus pengguna yang dipilih 13. Sistem akan menampilkan konfirmasi kepada aktor berupa <i>pop up</i> 14. Aktor menekan tombol 'Oke' pada <i>pop up</i> 15. Sistem menampilkan Pemberitahuan bahwa pengguna yang dipilih berhasil terhapus
Postcondition	Aktor berhasil mengelola data admin

Alternative Flow	<ol style="list-style-type: none"> 1. Jika aktor tidak melakukan input pada salah satu atau beberapa data pada form tambah/edit pengguna, maka sistem akan memberikan peringatan untuk mengisi form yang kosong 2. Jika aktor menekan tombol 'Batal' pada <i>pop up</i> saat melakukan penghapusan, maka data tidak jadi terhapus dari daftar pengguna
-------------------------	--

Sumber : (Penulis,2021)

Tabel 4. 23 Skenario Use Case Mencetak Detail Pesanan

Nama Use Case	Mencetak detail pesanan
Kode UseCase	UC_16
Aktor	admin
Tujuan	Aktor dapat mencetak detail pesanan
Precondition	Aktor telah berada di halaman admin
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih submenu 'Laporan Transaksi' 2. Sistem menampilkan halaman laporan transaksi 3. Aktor menekan tombol 'detail' 4. Sitem menampilkan detail data transaksi 5. Aktor menekan tombol 'Cetak laporan' 6. Sistem akan menampilkan halaman siap cetak dan unduh detail pesanan tersebut
Postcondition	Aktor berhasil mencetak detail pesanan
Alternative Flow	<ol style="list-style-type: none"> 1. Jika aktor menekan tombol 'Batal' pada <i>pop up validasi</i> saat melakukan pencetakan, maka laporan tidak jadi terunduh

Sumber : (Penulis,2021)

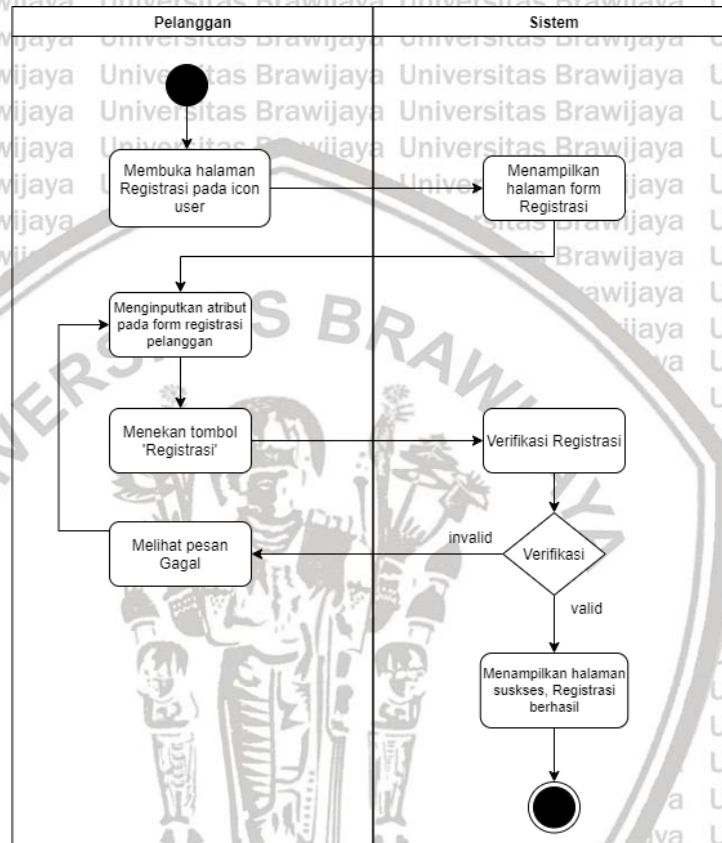
4.6 Activity Diagram

Activity Diagram merupakan diagram yang menjelaskan tentang alir kegiatan dalam program yang sedang dirancang, bagaimana proses alir berawal, keputusan yang mungkin terjadi, dan bagaimana sistem akan berakhir. Berikut adalah *activity diagram* dari E-commerce butik kebaya iffah.

4.6.1 Activity Diagram Register

Berikut adalah Activity Diagram Melakukan registrasi yang akan dijelaskan pada gambar 4.10. Sebelum dapat melakukan transaksi pada sistem, diperlukan

langkah registrasi, user dapat membuka halaman registrasi pada icon user, kemudian sistem akan menampilkan halaman regiter dan user dapat menginputkan seluruh atribut yang diminta pada form registrasi lalu user menekan tombol registrasi, yang kemudian sistem akan melakukan verifikasi email dan jika email yang didaftarkan belum ada pada sistem, maka registrasi berhasil, kemudian sistem mengarahkan user ke halaman sukses berisi informasi proses registrasi telah berhasil dan user dapat melanjutkan transaksi. Jika email sudah terdaftar pada sistem, maka proses registrasi gagal.



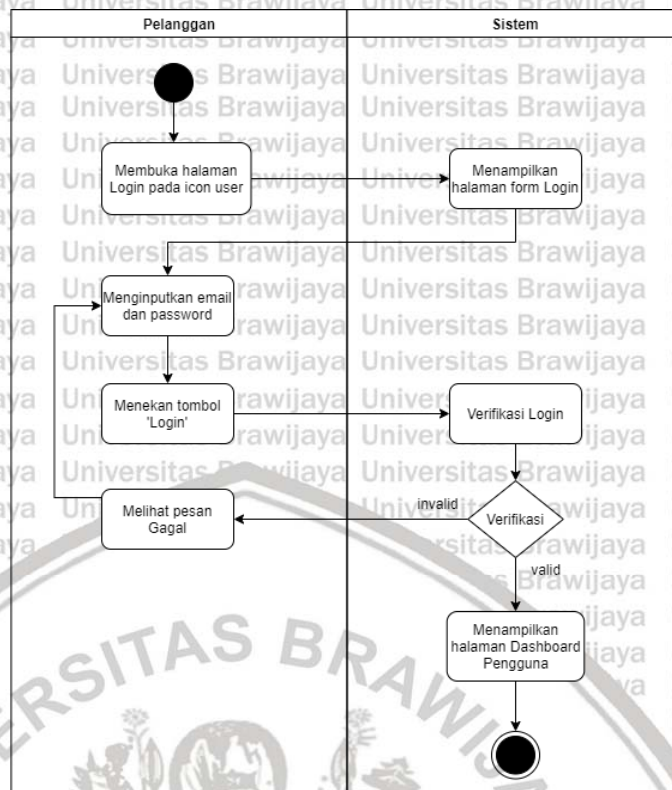
Gambar 4. 10 Activity Diagram Register

Sumber : (Penulis,2021)

4.6.2 Activity Diagram Login

Berikut adalah Activity Diagram Login pelanggan yang akan dijelaskan pada gambar 4.11. Dengan membuka halaman login pada icon user, kemudian sistem akan menampilkan halaman login pelanggan. User melakukan login dengan menggunakan email dan password yang telah terdaftar pada sistem. Jika email dan password sesuai, user akan diarahkan pada halaman dashboard pelanggan. Jika tidak sesuai, user akan kembali menemui halaman login dengan pesan gagal.

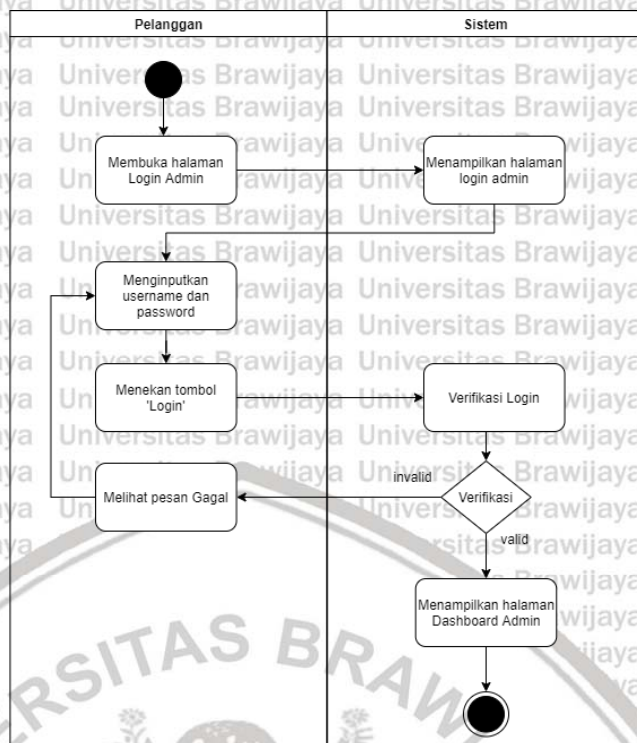




Gambar 4. 11 Activity Diagram Login Pelanggan

Sumber : (Penulis,2021)

Berikut adalah Activity Diagram Login admin yang akan dijelaskan pada gambar 4.12. Dengan membuka halaman login admin, kemudian sistem akan menampilkan halaman login admin. User melakukan login dengan menggunakan username dan password yang telah terdaftar pada sistem. Jika username dan password sesuai, user akan diarahkan pada halaman dashboard pelanggan. Jika tidak sesuai, user akan kembali menemui halaman login dengan pesan gagal.

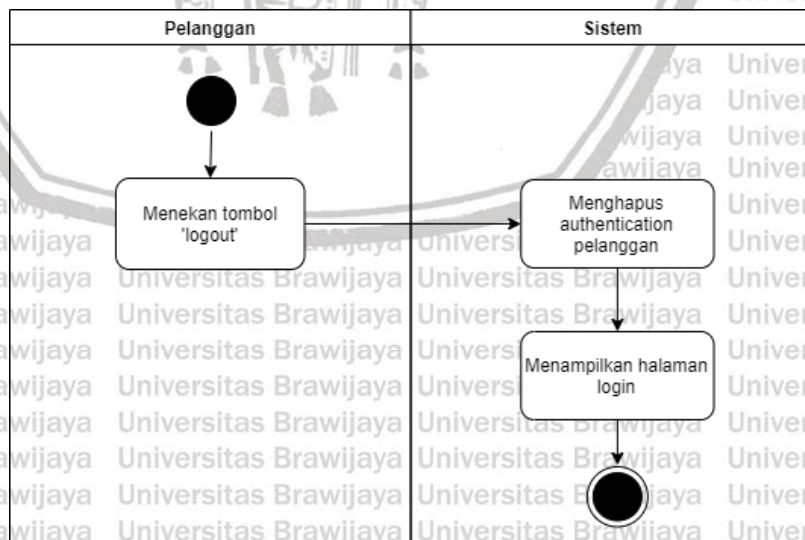


Gambar 4. 12 Activity Diagram Login Admin

Sumber : (Penulis,2021)

4.6.3 Activity Diagram Logout

Berikut adalah Activity Diagram logout yang dilakukan oleh pelanggan dan admin akan dijelaskan pada gambar 4.13. Jika users ingin meninggalkan sistem setelah masuk pada sistem, atau proses logout dapat dimulai dengan users menekan tombol 'logout' dan kemudian sistem akan menghapus session dari user dan jika berhasil maka sistem akan menampilkan halaman login.

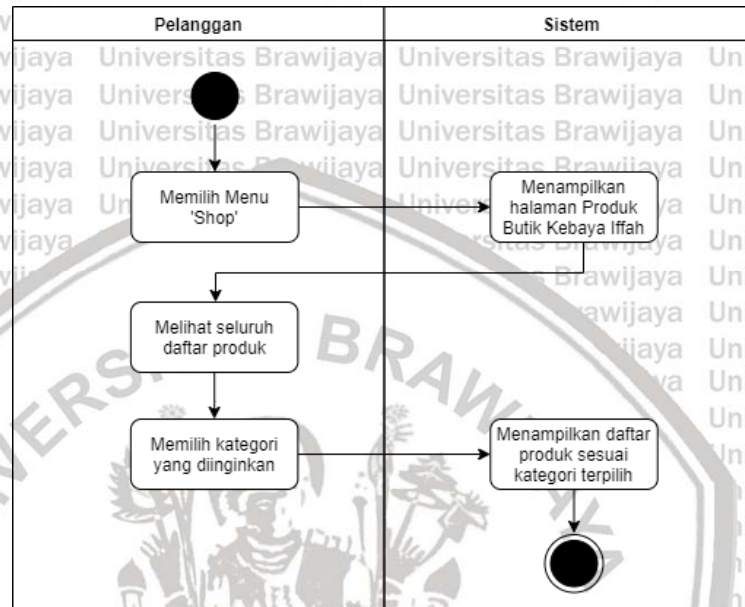


Gambar 4. 13 Activity Diagram Logout pada pelanggan dan admin

Sumber : (Penulis, 2021)

4.6.4 Activity Diagram Melihat Daftar Produk

Berikut adalah Activity Diagram melihat daftar produk yang akan dijelaskan pada gambar 4.14. user dapat melihat daftar produk butik kebaya iffah dengan memilih menu 'shop'. Kemudian sistem akan menampilkan seluruh daftar produk yang terdaftar, user juga dapat memilih kategori yang ingin ditampilkan, maka sistem akan menampilkan daftar produk sesuai dengan kategori yang dipilih.

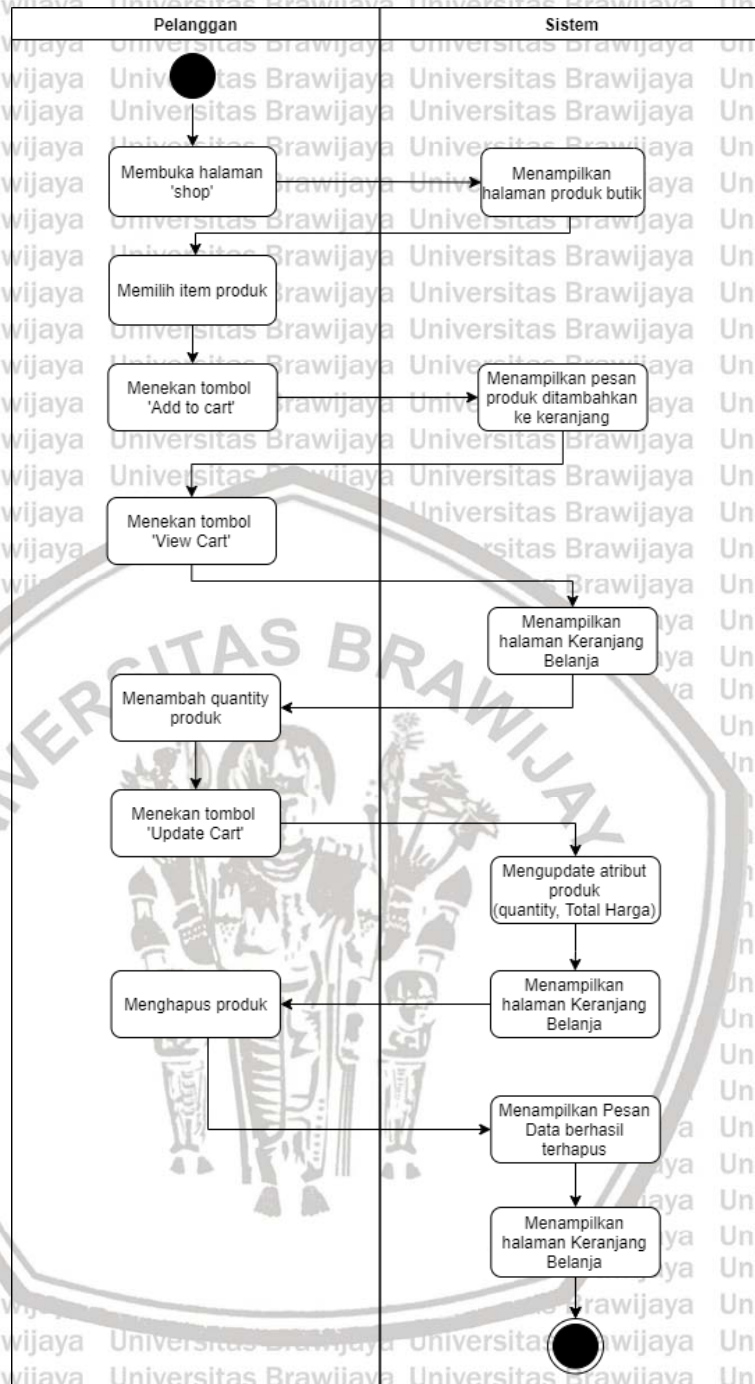


Gambar 4. 14 Activity Diagram Melihat Daftar Produk

Sumber : (Penulis, 2021)

4.6.5 Activity Diagram Mengelola Keranjang Belanja

Berikut adalah Activity Diagram mengelola keranjang belanja yang akan dijelaskan pada gambar 4.15. Terdapat 3 aktivitas yaitu pertama User dapat menambahkan produk yang diinginkan dengan memilih tombol 'Add to cart', sistem akan menampilkan bahwa produk berhasil dimasukkan ke keranjang belanja. Aktivitas kedua yaitu user dapat melakukan perubahan, dengan menambah jumlah produk, setelah dilakukan perubahan dan klik update cart, maka data keranjang belanja berhasil diperbarui. Aktivitas ketiga yaitu user dapat menghapus produk dari keranjang belanja dengan memilih tombol 'delete' pada produk yang diinginkan, kemudian sistem akan menampilkan pop up validasi, jika 'Ya' maka data akan terhapus, dan jika 'Tidak' maka sistem akan kembali menampilkan halaman daftar keranjang belanja.



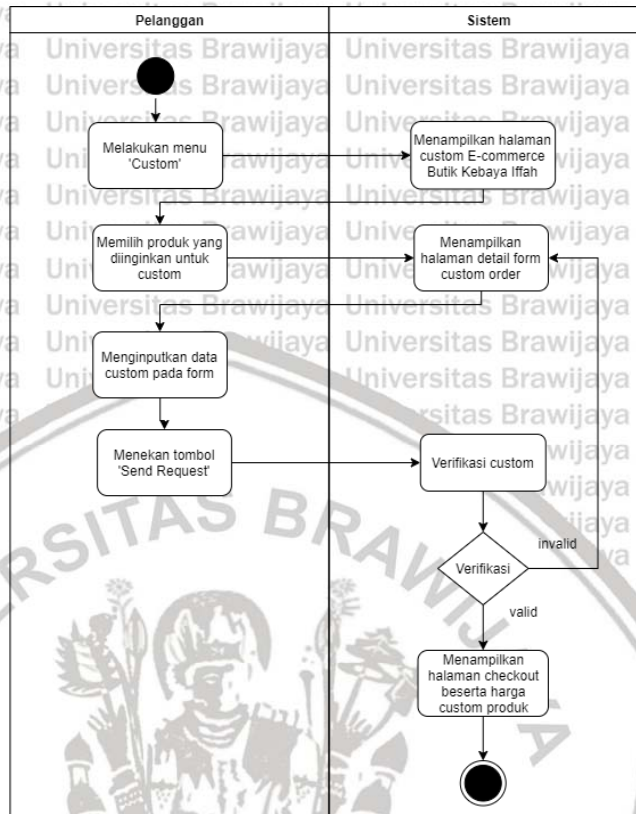
Gambar 4. 15 Activity Diagram Mengelola Keranjang Belanja

Sumber : (Penulis,2021)

4.6.6 Activity Diagram Melakukan Custom Order

Berikut adalah Activity Diagram melakukan custom order yang akan dijelaskan pada gambar 4.16. User terlebih dahulu melakukan login, user dapat melakukan custom order pada halaman custom, user dapat memilih produk dan kemudian akan ditampilkan form untuk keperluan custom order, lalu dengan menekan tombol send request, user akan diarahkan pada halaman checkout

dengan total harga custom produk. Jika validasi form salah maka akan diberikan untuk mengisi kembali, namun jika user belum login maka akan ditampilkan halaman login untuk melakukan login terlebih dahulu.

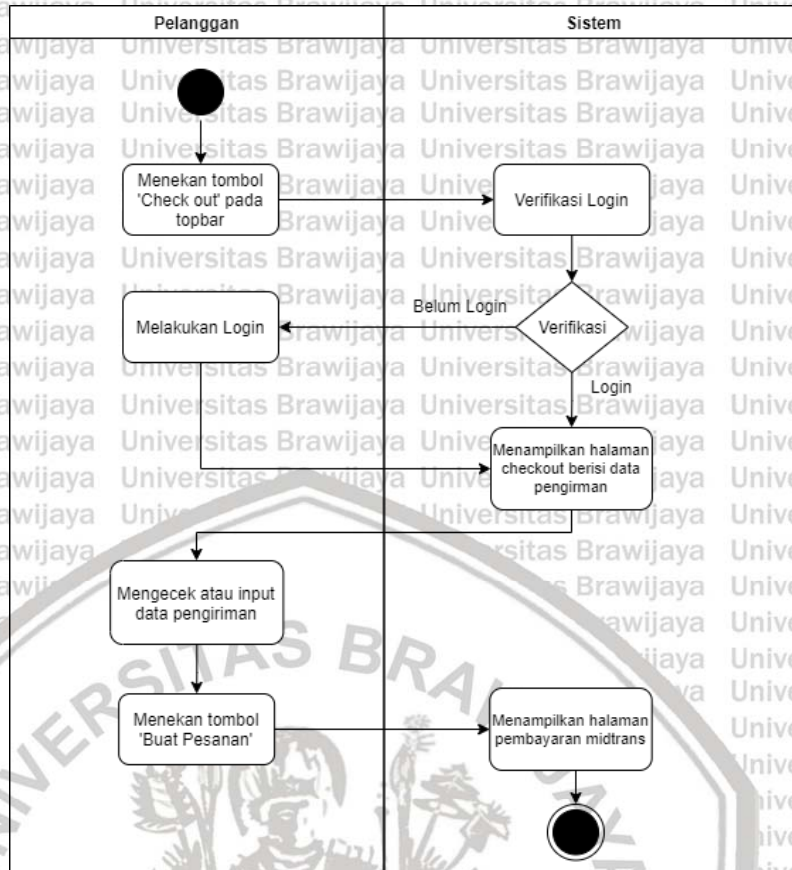


Gambar 4. 16 Activity Diagram Melakukan Custom Order

Sumber : (Penulis,2021)

4.6.7 Activity Diagram Melakukan Checkout

Berikut adalah Activity Diagram Melakukan Checkout yang akan dijelaskan pada gambar 4.17. User terlebih dahulu melakukan *login*, user memilih submenu keranjang belanja, lalu memilih 'check out' untuk melakukan pemesanan, kemudian sistem akan menampilkan data yang akan digunakan untuk proses pengiriman, jika data dirasa sudah benar maka dilanjutkan dengan memilih 'Buat Pesanan', sehingga data pesanan sudah terdaftar, dan user akan diarahkan untuk proses pembayaran.

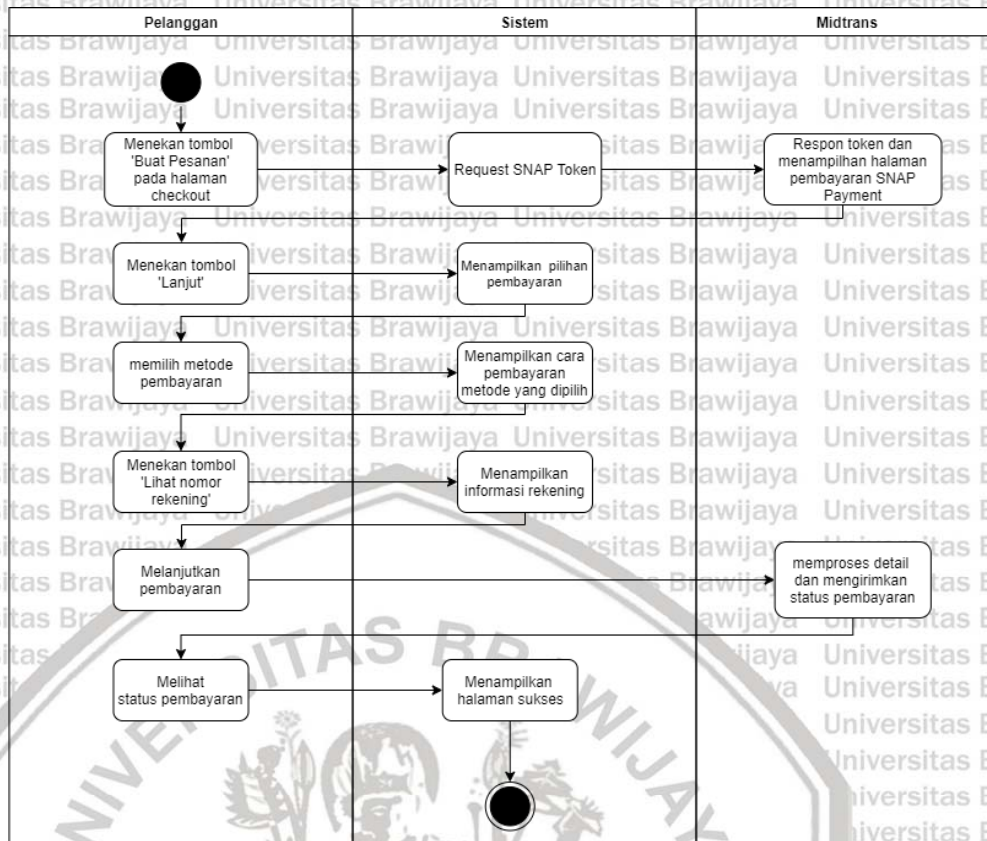


Gambar 4. 17 Activity Diagram Melakukan Checkout

Sumber : (Penulis,2021)

4.6.8 Activity Diagram Melakukan Pembayaran

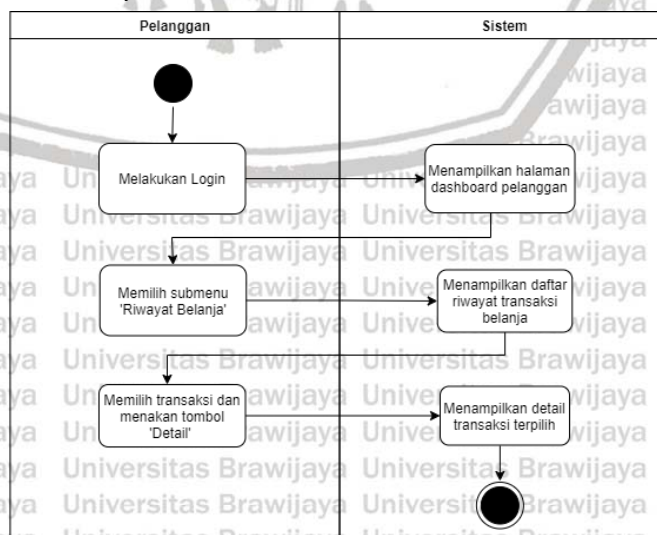
Berikut adalah Activity Diagram Melakukan pembayaran yang akan dijelaskan pada gambar 4.18. User terlebih dahulu melakukan proses checkout, setelah pada halaman checkout user membuat pesanan, maka sistem akan melakukan request SNAP token pada midtrans, dan kemudian api midtrans merespon dengan menampilkan SNAP Payment tau halaman pembayaran dan nominal harga sesuai tipe pembelian yaitu ready atau dari halaman custom order. Selanjutnya user memilih metode pembayaran, lalu sistem menampilkan cara pembayaran, user memilih lihat nomor rekening, dan sistem menampilkan informasi rekening, selanjutnya user akan melanjutkan pembayaran, dan setelah terbayar sistem akan mengecek pembayaran, dan user dapat melihat status pembayaran dan pembayaran berhasil tersimpan.



Gambar 4. 18 Activity Diagram Melakukan Pembayaran
 Sumber : (Penulis,2021)

4.6.9 Activity Diagram Melihat Riwayat Transaksi

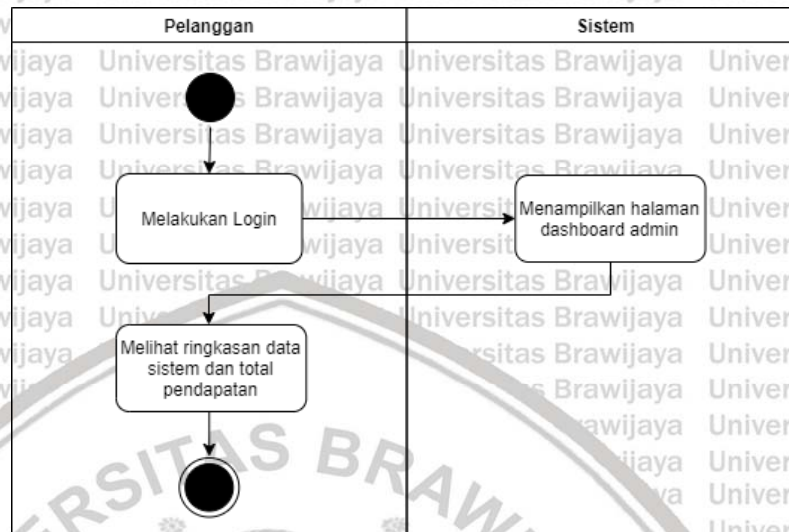
Berikut adalah Activity Diagram melihat riwayat transaksi pada gambar 4.19. user dapat melihat riwayat pada dashboard pelanggan sub menu 'riwayat belanja'.Sistem akan menampilkan daftar riwayat transaksi belanja pelanggan,dan detailnya.



Gambar 4. 19 Activity Diagram Melihat Riwayat Transaksi
 Sumber : (Penulis, 2021)

4.6.10 Activity Diagram Dashboard

Berikut adalah *Activity Diagram* melihat dashboard admin yang akan dijelaskan pada gambar 4.20. *User* terlebih dahulu melakukan login, dan akan ditampilkan dashboard admin yang berisi ringkasan sistem.

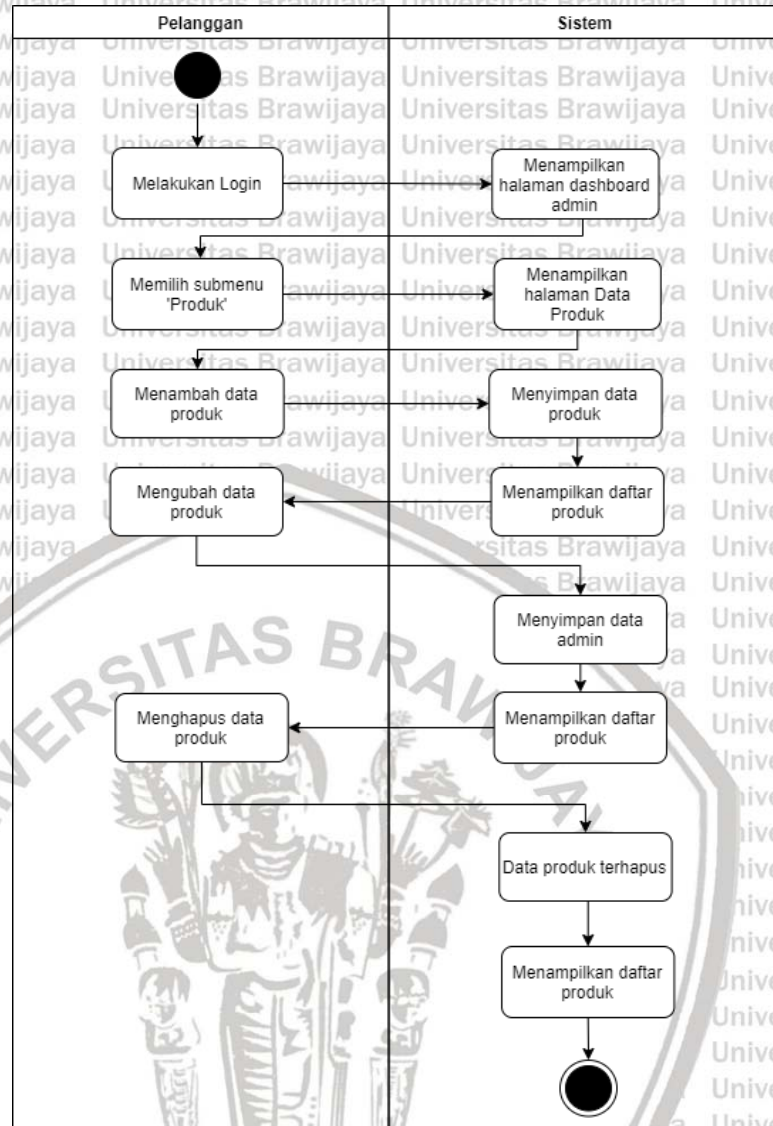


Gambar 4. 20 Activity Diagram Dashboard Admin

Sumber : (Penulis,2021)

4.6.11 Activity Diagram Mengelola Produk

Berikut adalah *Activity Diagram* mengelola produk yang akan dijelaskan pada gambar 4.21. *user* terlebih dahulu melakukan *login*, dan tertuju pada halaman *dashboard* admin, kemudian sistem akan menampilkan halaman untuk menginputkan data produk. Selanjutnya *user* akan menginputkan data produk dengan menekan tombol 'Tambah'. Setelah ditampilkan form untuk input data produk, *user* harus mengisi seluruh form, jika salah satu kolom tidak diisi maka akan muncul pesan gagal simpan. Jika semua sudah diisi maka akan muncul pesan berhasil simpan. Selanjutnya admin juga dapat melakukan edit pada produk yang diinginkan dan menghapus produk yang dipiih.

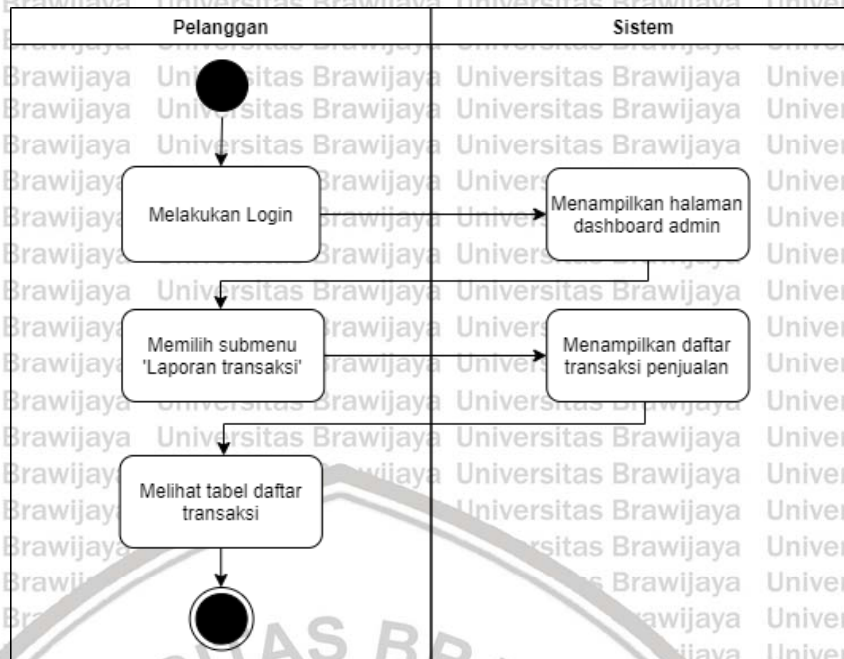


Gambar 4. 21 Activity Diagram Mengelola Produk

Sumber : (Penulis,2021)

4.6.12 Activity Diagram Melihat Data Pesanan

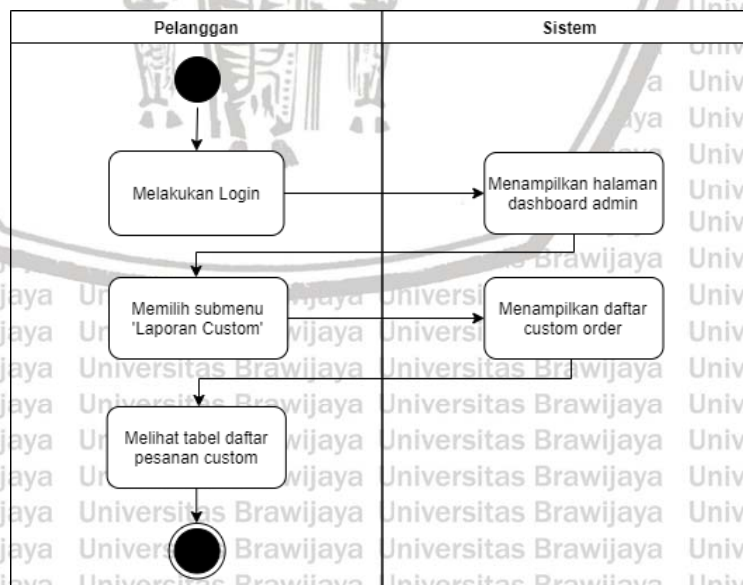
Berikut adalah Activity Diagram melihat data pesanan yang akan dijelaskan pada gambar 4.22. User dapat melihat laporan daftar transaksi penjualan yang telah terjadi dan untuk melihat lebih lanjut user dapat menekan tombol detail.



Gambar 4. 22 Activity Diagram Melihat Data Pesanan
 Sumber : (Penulis,2021)

4.6.13 Activity Diagram Melihat Data Custom Order

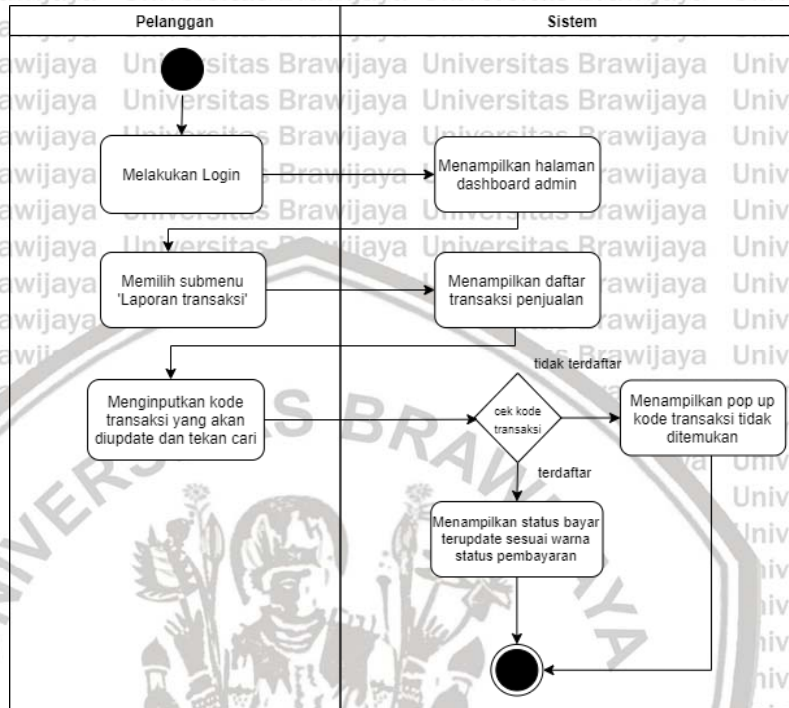
Berikut adalah *Activity Diagram* daftar melihat custom order pada halaman admin admin yang akan dijelaskan pada gambar 4.23. *User* dapat melihat laporan daftar custom order penjualan yang telah terjadi, dan untuk melihat lebih lanjut mengenai atribut atribut cutom user dapat menekan tombol detail.



Gambar 4. 23 Activity Diagram Melihat Data Custom Order
 Sumber : (Penulis,2021)

4.6.14 Activity Diagram Update Status Bayar

Berikut adalah Activity Diagram daftar melakukan update staus bayar yang akan dijelaskan pada gambar 4.24. User dapat melakukan update status pembayaran yang telah dilakukan pelanggan pada transaksinya dengan menginputkan kode transaksi dan kemudian data akan terupdate.

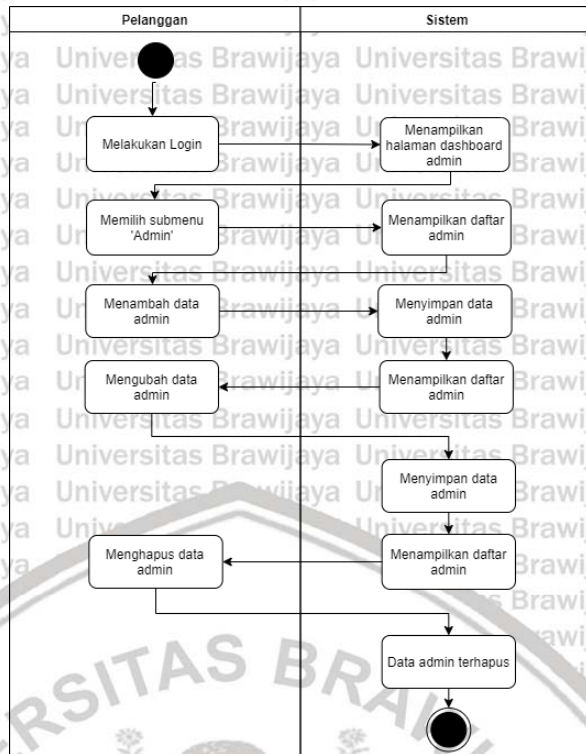


Gambar 4. 24 Activity Diagram Update Status Bayar

Sumber : (Penulis,2021)

4.6.15 Activity Diagram Mengelola Admin

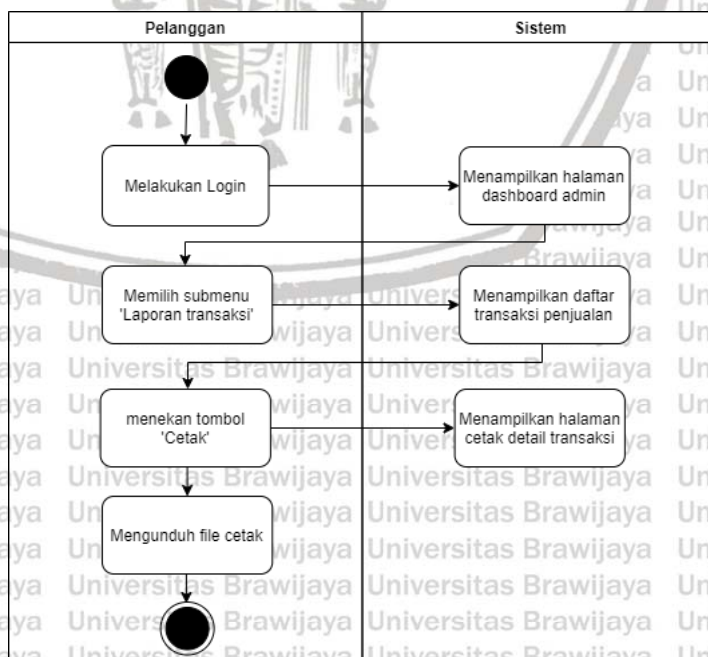
Berikut adalah Activity Diagram manajemen pengguna yang akan dijelaskan pada gambar 4.25. Aktor dapat melakukan aktivitas create,read,update dan delete pada admin.



Gambar 4. 25 Activity Diagram Mengelola Admin
 Sumber : (Penulis,2021)

4.6.16 Activity Diagram Mencetak Detail Pesanan

Berikut adalah Activity Diagram mencetak detail pesanan yang akan dijelaskan pada gambar 4.26. User dapat mencetak detail pesanan dengan membuka transaksi pada sub menu laporan, dan nantinya akan ditampilkan halaman berupa preview file dan juga pilihan untuk mengunduh detail pesanan.



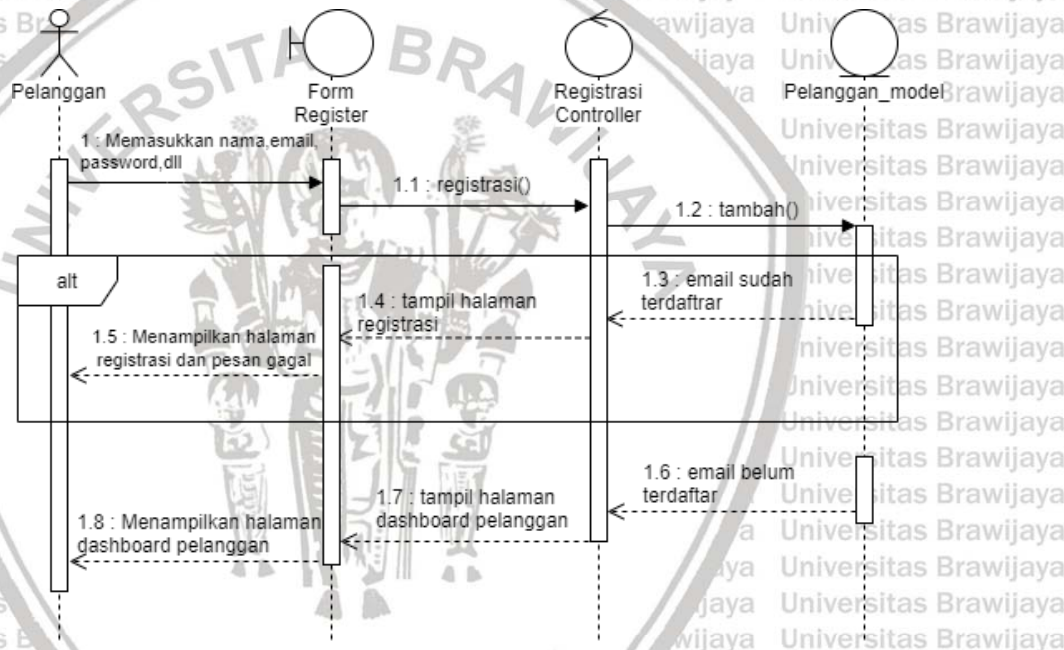
Gambar 4. 26 Activity Diagram Mencetak Detail Pesanan
 Sumber : (Penulis, 2021)

4.7 Sequence Diagram

Pemodelan *Sequence Diagram* ditujukan untuk memodelkan interaksi dan skenario yang dijalankan dalam sistem. Website E-commerce Butik Kebaya Iffah.

4.7.1 Sequence Diagram Register

Berikut adalah *Sequence Register* yang dijelaskan pada gambar 4.27. Pertama-tama pelanggan membuka halaman registrasi dan menginputkan nama, email, password dan lainnya untuk keperluan register. Setelah itu controller register akan menjalankan fungsi registrasi untuk melakukan cek apakah email yang diinputkan sudah terdaftar atau belum, jika belum, akan dijanakna fungsi tambah pada pelanggan model, dan data tersimpan pada database dan menampilkan halaman sukses registrasi, namun jika email sudah terdaftar maka user akan diminta untuk menggunakan email yang lain.

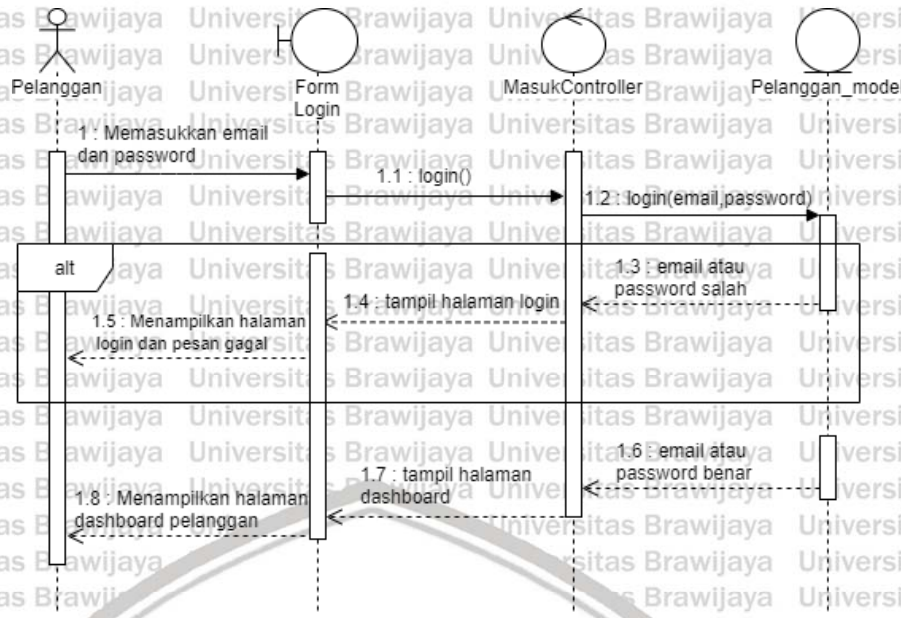


Gambar 4. 27 Sequence Diagram Register

Sumber : (Penulis, 2021)

4.7.2 Sequence Diagram Login

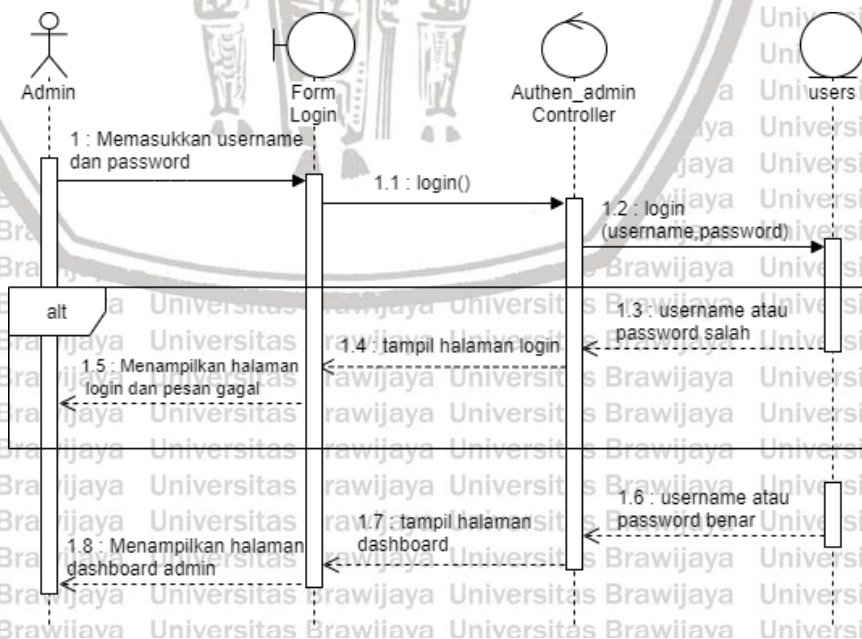
Berikut adalah *Sequence Diagram Login* oleh pelanggan yang dijelaskan pada gambar 4.28. Pertama-tama pelanggan memasuki halaman *Login*, kemudian memasukkan email dan password kemudian controller login akan menjalankan fungsi *login*, Sistem akan mengecek data pelanggan pada *database*, dengan fungsi login pada pelanggan model. dan jika *email* dan *password* sesuai maka akan ditampilkan halaman dashboard pelanggan, namun jika salah maka akan ditampilkan halaman login dengan pesan gagal.



Gambar 4. 28 Sequence Diagram Login Pelanggan

Sumber : (Penulis, 2021)

Berikut adalah Sequence Diagram Login oleh pelanggan yang dijelaskan pada gambar 4.29. Pertama-tama admin membuka halaman login admin, kemudian memasukkan username dan password kemudian controller authen admin akan menjalankan fungsi *login*, Sistem akan mengecek data admin pada *database*, dengan fungsi login pada user model. dan jika *username* dan *password* sesuai maka akan ditampilkan halaman dashboard admin, namun jika salah maka akan ditampilkan halaman login dengan pesan gagal.

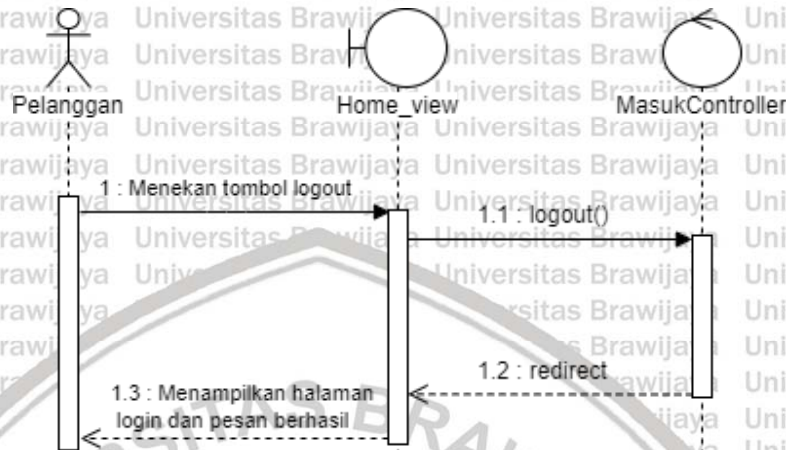


Gambar 4. 29 Sequence Diagram Login Admin

Sumber : (Penulis, 2021)

4.7.3 Sequence Diagram logout

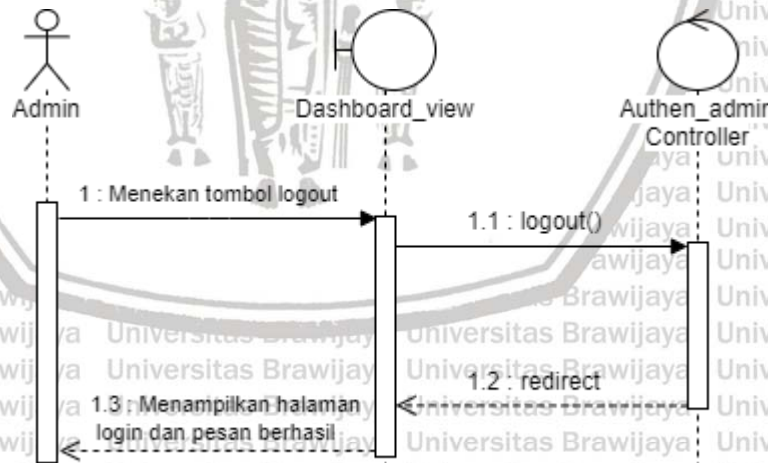
Berikut adalah *Sequence Diagram logout* yang dijelaskan pada gambar 4.30. untuk melakukan logout pelanggan dapat menekan logout yang kemudian masuk_controller akan menjalankan fungsi logout menghapus session pelanggan dan mengarahkan kembali pada halaman login.



Gambar 4. 30 Sequence Diagram Logout Pelanggan dan Admin

Sumber : (Penulis, 2021)

Berikut adalah *Sequence Diagram logout admin* yang dijelaskan pada gambar 4.31. untuk melakukan logout admin dapat menekan logout yang kemudian authen admin controller akan menjalankan fungsi logout menghapus session pelanggan dan mengarahkan kembali pada halaman login

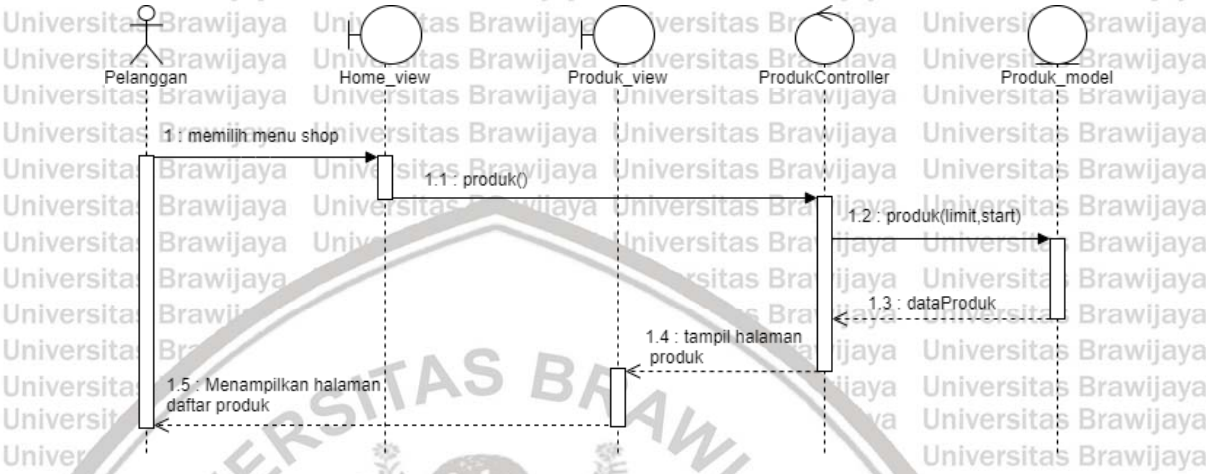


Gambar 4. 31 Sequence Diagram Logout Admin

Sumber : (Penulis, 2021)

4.7.4 Sequence Diagram Melihat Daftar Produk

Berikut adalah *Sequence Diagram* melihat daftar produk yang dijelaskan pada gambar 4.32. diawali dengan pelanggan memilih menu shop dan akan dijalankan fungsi produk pada controller produk, dan mengambil data dari database produk. Setelahnya sistem akan mengembalikan dengan tampil halaman daftar produk.

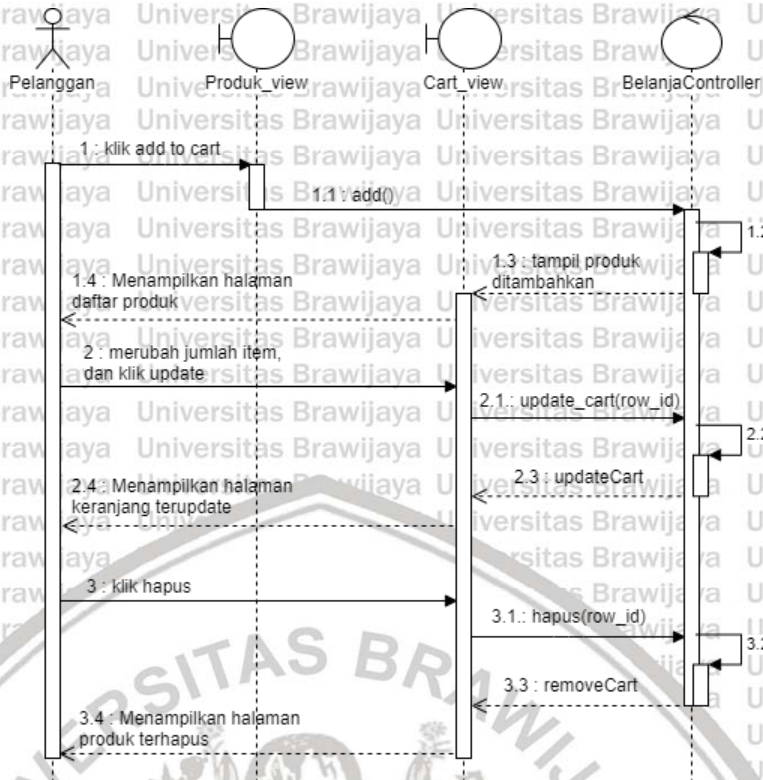


Gambar 4. 32 *Sequence Diagram* melihat daftar produk

Sumber : (Penulis, 2021)

4.7.5 Sequence Diagram Mengelola Keranjang Belanja

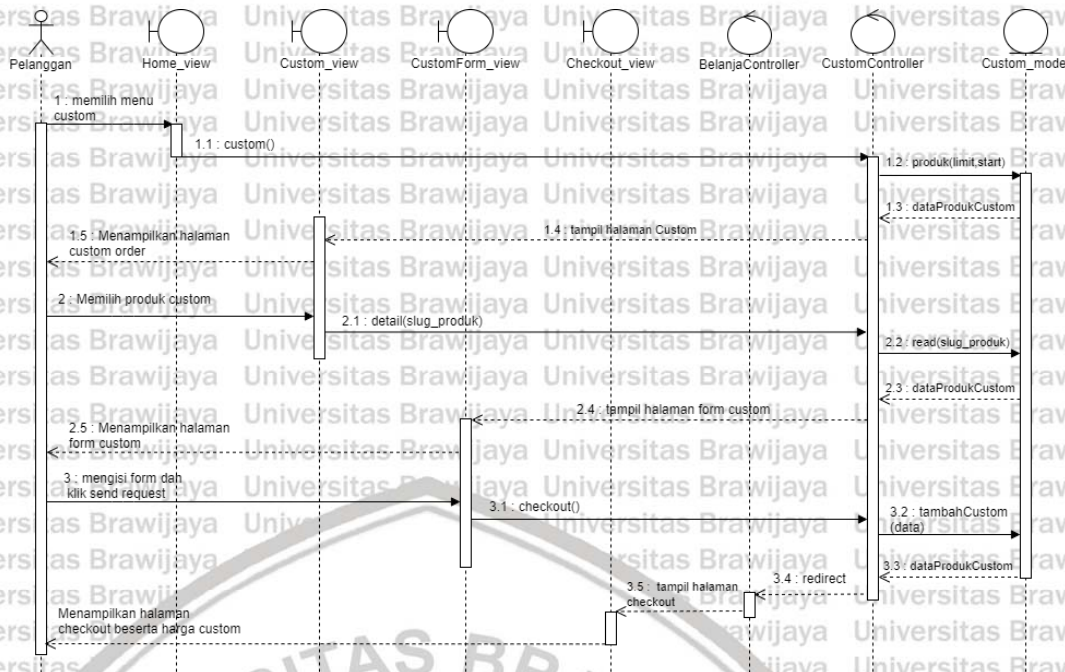
Berikut adalah *Sequence Diagram* Mengelola keranjang yang dijelaskan pada gambar 4.33. untuk dapat mengelola keranjang, terdapat tiga aktifitas dimana user dapat memasukkan produk ke keranjang dengan menekan tombol add to cart akan dieksekusi oleh fungsi add. Untuk proses edit dengan menambah quantity produk akan dijalankan fungsi updatecart, dan untuk hapus akan dijalankan fungsi hapus dengan parameter rowid, dan fungsi destroy untuk hapus semua. Fungsi fungsi keranjang pada controller belanja ini menggunakan library CI_cart.



Gambar 4. 33 Sequence Diagram Mengelola Keranjang Belanja
 Sumber : (Penulis, 2021)

4.7.6 Sequence Diagram Melakukan Custom Order

Berikut adalah Sequence Diagram custom order yang dijelaskan pada gambar 4.34. Dimulai dengan memilih menu custom akan dijalankan fungsi custom dan data dari produk pada custom model, ditampilkan kembali pada pelanggan halaman custom order berisikan form input, setelah diisi custom controller akan menjalankan fungsi checkout dan menambah data dengan fungsi tambahCustom pada custom model dan tampil halaman checkout custom dengan harga customnya.

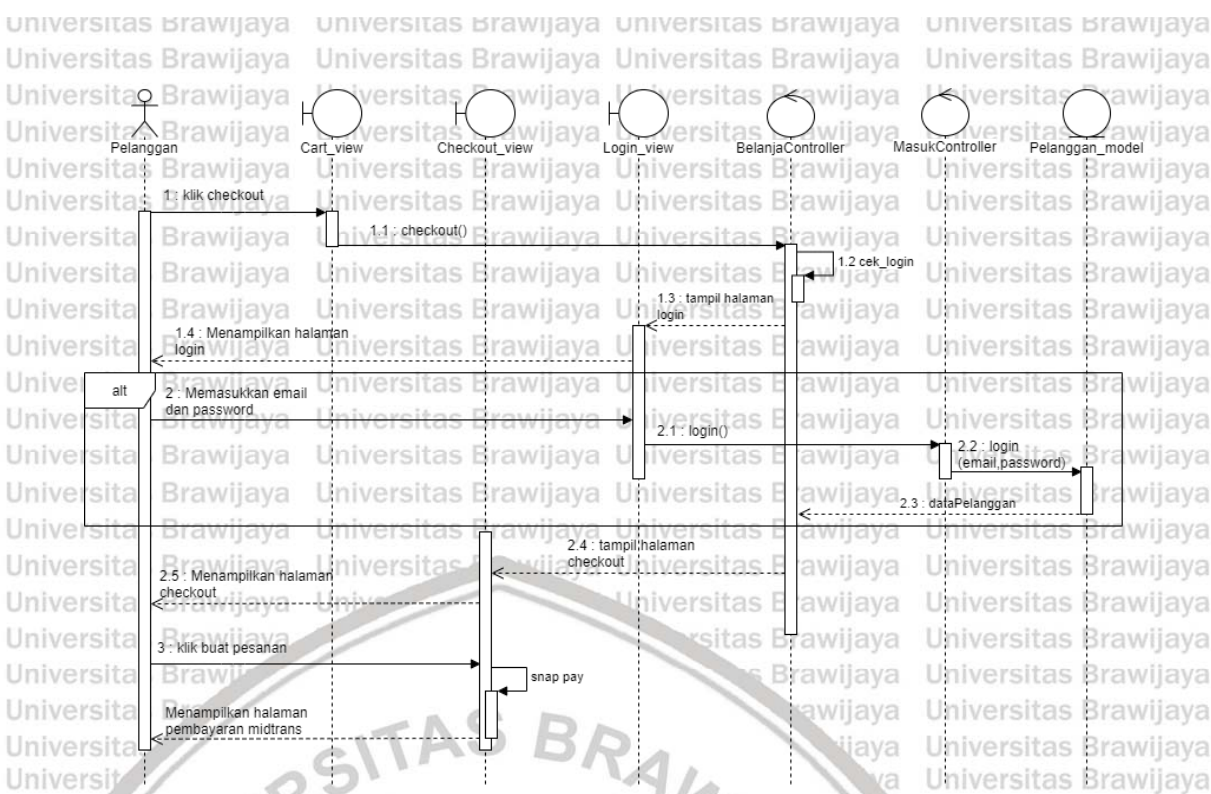


Gambar 4. 34 Sequence Diagram Melakukan Custom Order

Sumber : (Penulis, 2021)

4.7.7 Sequence Diagram Melakukan Checkout

Berikut adalah *Sequence* melakukan *checkout* yang dijelaskan pada gambar 4.35. dimulai dari pelanggan membuka halaman keranjang dan menekan klik akan dijalankan fungsi *checkout* pada *belanja controller*, dan akan dilakukan fungsi *cek_login* jika belum maka akan dikembalikan tampil halaman login. Jika aktor sudah login maka akan ditampilkan halaman *checkout* dan proses dilanjutkan dengan *snap pay* pada proses pembayaran dengan *midtrans*.

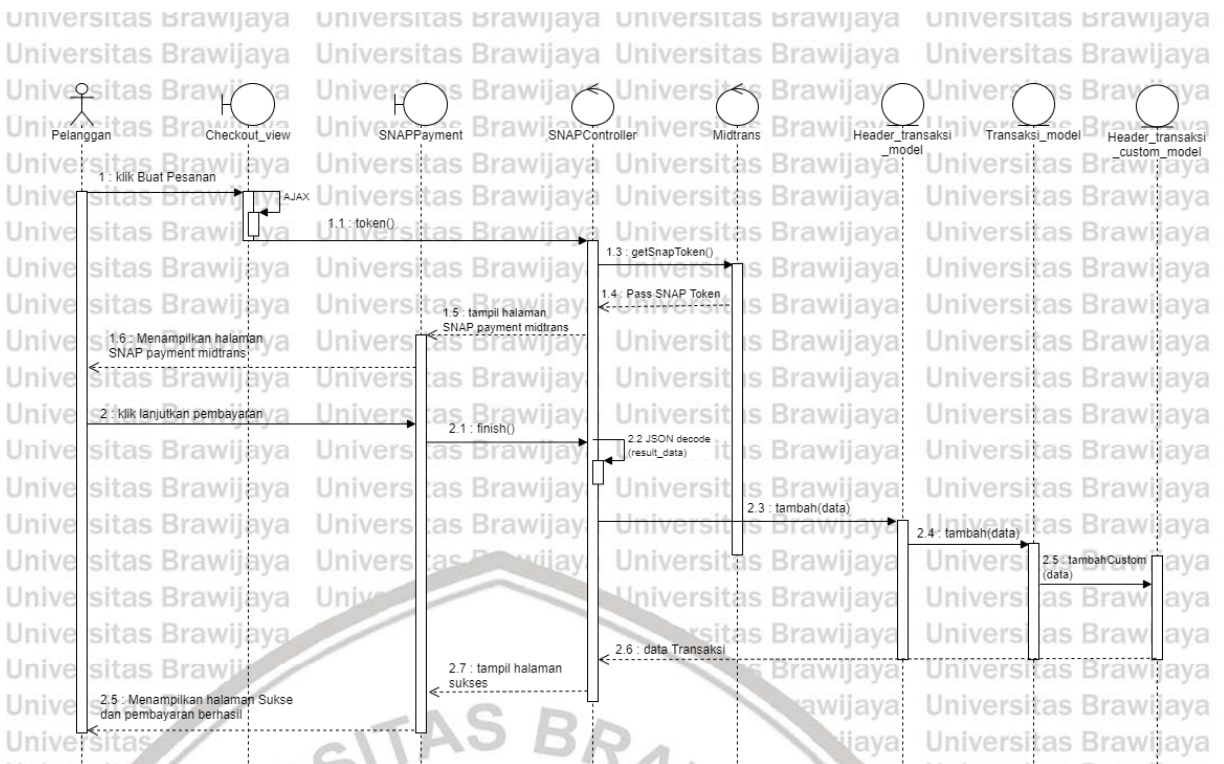


Gambar 4. 35 Sequence Diagram Melakukan Checkout

Sumber : (Penulis, 2021)

4.7.8 Sequence Diagram Melakukan Pembayaran

Berikut adalah *Sequence Diagram* melakukan pembayaran yang dijelaskan pada gambar 4.36. Dimulai dari halaman checkout produk ready maupun produk custom, saat pelanggan menekan tombol buat pesanan pada view data tersebut dipanggil ajax yang akan menjalankan fungsi token pada controller snap. Kemudian controller merequest SNAP token, dan midtrans merespon snap token dan akan tampil halaman SNAP payment. Dilanjutkan dengan user melanjutkan pembayaran, dan akan dijalankan fungsi finish pada controller snap, midtrans akan mengembalikan data berupa json dan transaksi akan ditambahkan pada transaksi dan header transaksi, sedangkan untuk custom akan ditambahkan pada header transaksi custom, dan tampil halaman transaksi berhasil.

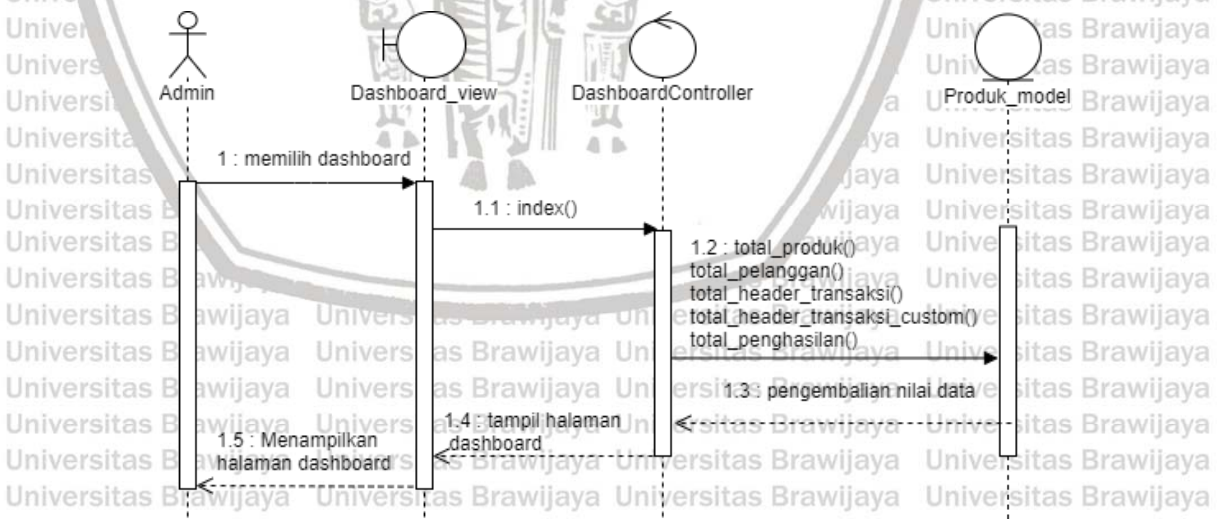


Gambar 4. 36 Sequence Diagram Pembayaran

Sumber : (Penulis, 2021)

4.7.9 Sequence Diagram Melihat Dashboard

Berikut adalah Sequence Diagram melihat dashboard yang dijelaskan pada gambar 4.37. dimana saat memilih halaman dashboard, akan dijalankan fungsi index pada controller dashboard admin. Dan akan mengambil data dari dashboard model, kan dikembalikan berupa halaman dashboard beserta data.

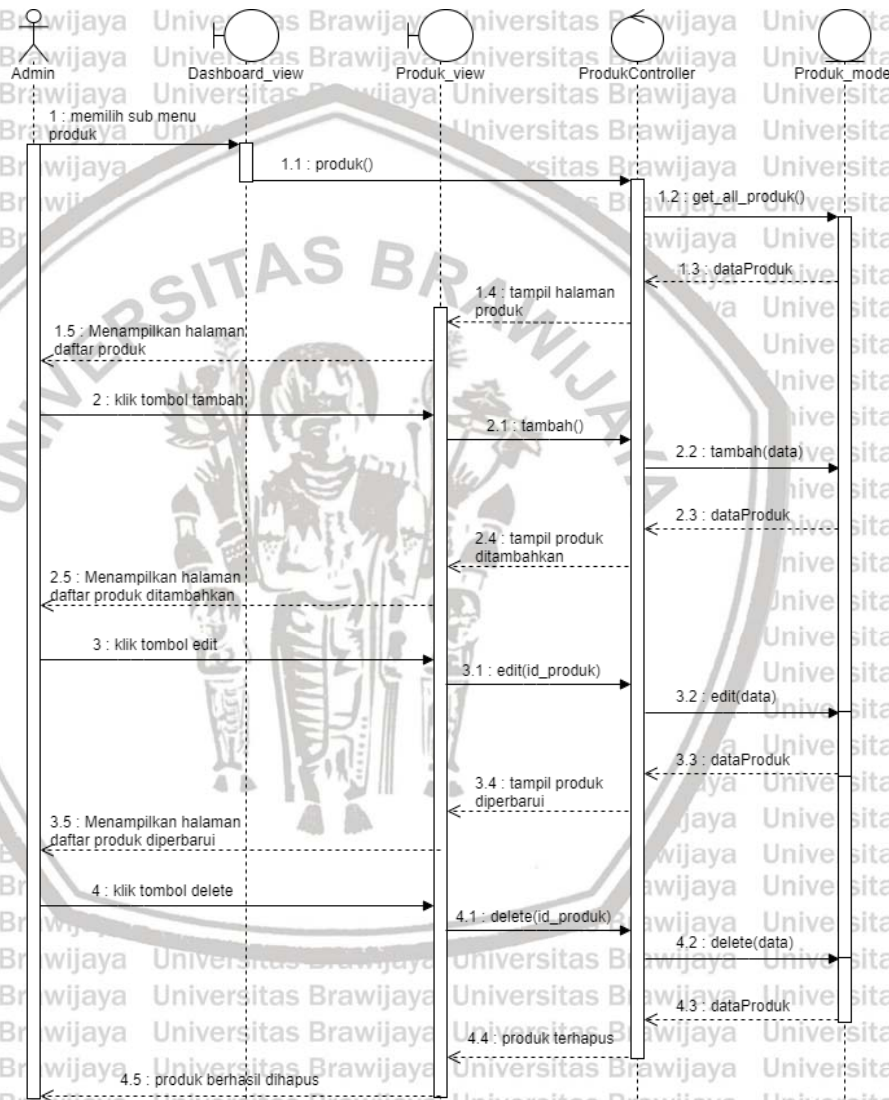


Gambar 4. 37 Sequence Diagram Melihat Dashboard

Sumber : (Penulis, 2021)

4.7.10 Sequence Diagram Mengelola Produk

Berikut adalah *Sequence Diagram* mengelola produk yang dijelaskan pada gambar 4.38. Pertama-tama admin memilih sub menu produk dan dijalankan fungsi produkcontroller dengan mengambil data produk fungsi `get_all_produk` pada produk model. Saat user menekan tombol tambah akan dijalankan fungsi tambah pada controller dan data ditambahkan dengan fungsi tambah pada model. Sama juga seperti proses tambah, proses edit hanya berbeda nama fungsi, yaitu fungsi edit dengan parameter `id_produk`. Untuk menghapus produk dijalankan fungsi delete dengan parameter `id_produk` pada controller.

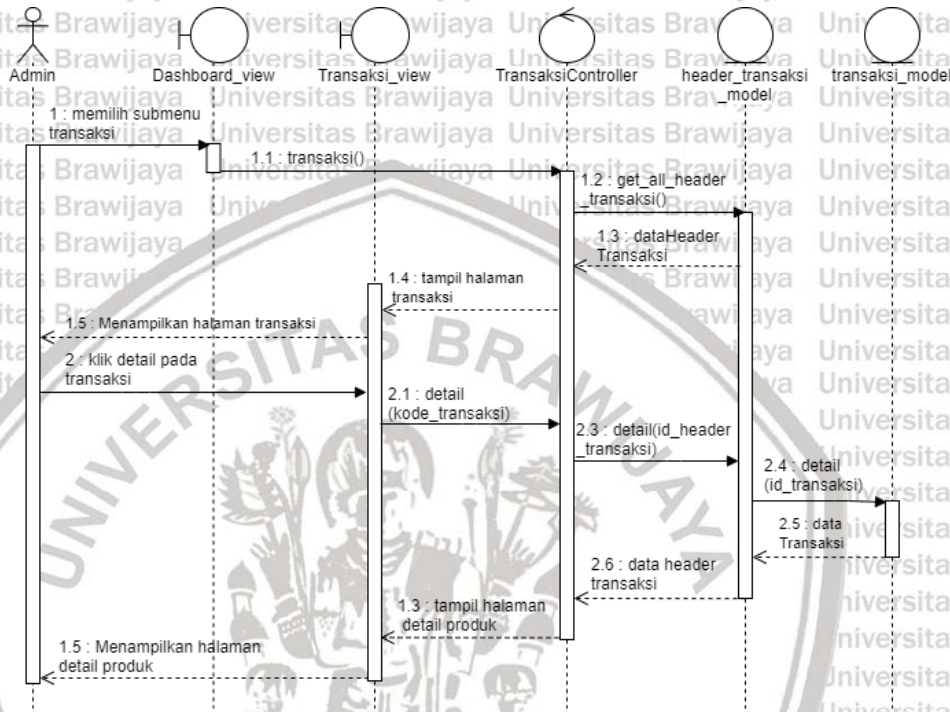


Gambar 4. 38 Sequence Diagram Mengelola Produk

Sumber : (Penulis, 2021)

4.7.11 Sequence Diagram Melihat Data Pesanan

Berikut adalah *Sequence Diagram* melihat data pesanan yang dijelaskan pada gambar 4.39. Pertama-tama admin dapat memilih submenu transaksi dan dijalankan pengambilan data dari `get_all_header_transaksi` pada model header transaksi, kemudian tampil daftar transaksi. Admin juga dapat melihat detail yang akan dijalankan fungsi detail dengan parameter kode transaksi pada controller, dan diambil data pada fungsi detail dari header dan juga transaksi.

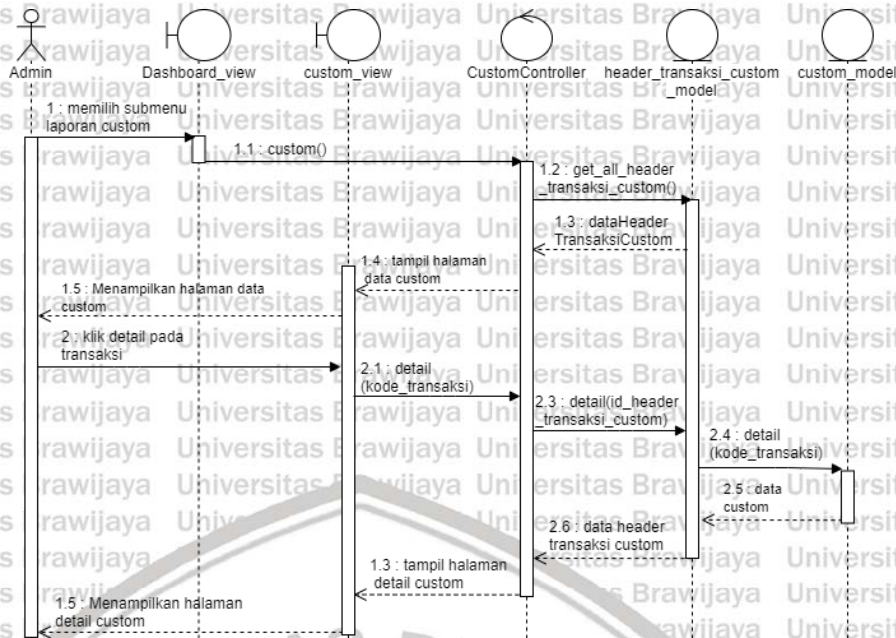


Gambar 4. 39 Sequence Diagram Melihat Data Pesanan

Sumber : (Penulis, 2021)

4.7.12 Sequence Diagram Melihat Data Custom Order

Berikut adalah *Sequence Diagram* melihat data *custom order* yang dijelaskan pada gambar 4.40. Pertama-tama admin dapat memilih submenu laporan custom dan dijalankan pengambilan data dari `get_all_header_transaksi_custom` pada model header transaksi custom, kemudian tampil daftar custom order. Admin juga dapat melihat detail yang akan dijalankan fungsi detail dengan parameter kode transaksi pada controller, dan diambil data pada fungsi detail dari header dan juga transaksi custom.

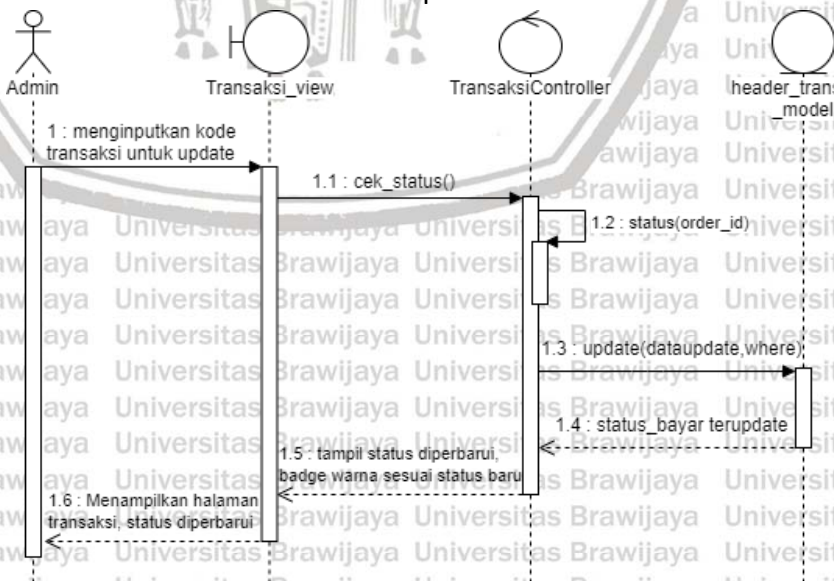


Gambar 4. 40 Sequence Diagram Melihat Data Custom Order

Sumber : (Penulis, 2021)

4.7.13 Sequence Diagram Update Status Bayar

Berikut adalah *Sequence Diagram update* status bayar yang dijelaskan pada gambar 4.41. Pertama-tama admin membuka halaman transaksi, kemudian menginputkan kode transaksi yang akan diupdate, lalu pada controller transaksi dijalankan fungsi `cek_status` yang menjalankan fungsi private status dengan parameter `order_id`, setelah data didapat dilakukan update pada database dengan dijalankan fungsi `update`, dan mengembalikan nilai status bayar kemudian status berganti sesuai warna keterangan. Proses ini juga dapat dilakukan pada halaman laporan custom order untuk melakukan update status.

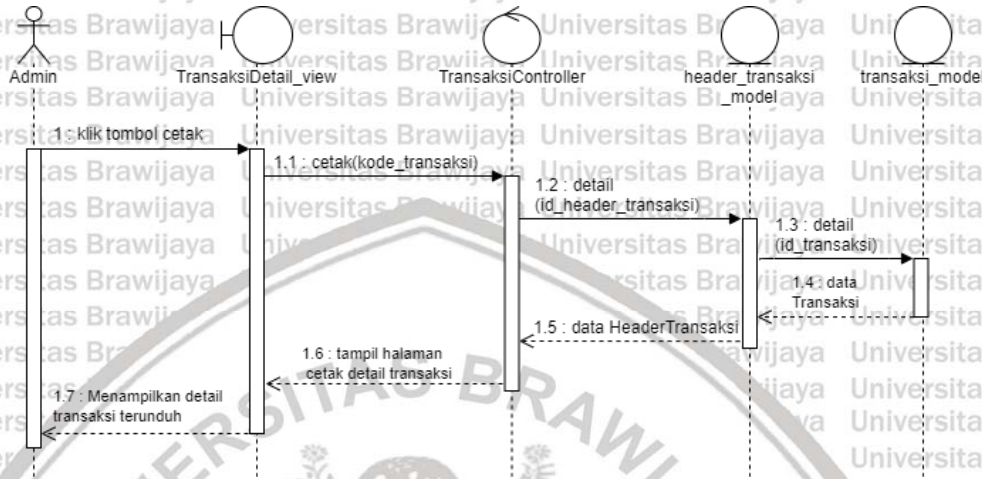


Gambar 4. 41 Sequence Diagram Update Status Bayar

Sumber : (Penulis, 2021)

4.7.14 Sequence Diagram Mencetak Detail Pesanan

Berikut adalah *Sequence Diagram* mencetak detail pesanan yang dijelaskan pada gambar 4.42. Pertama-tama admin membuka halaman transaksi dan memilih detail transaksi yang akan dicetak. Setelah klik dijalankan fungsi cetak dengan data detail dari heder transaksi model dan transaksi model dan kemudian tampil preview dan unduh file.

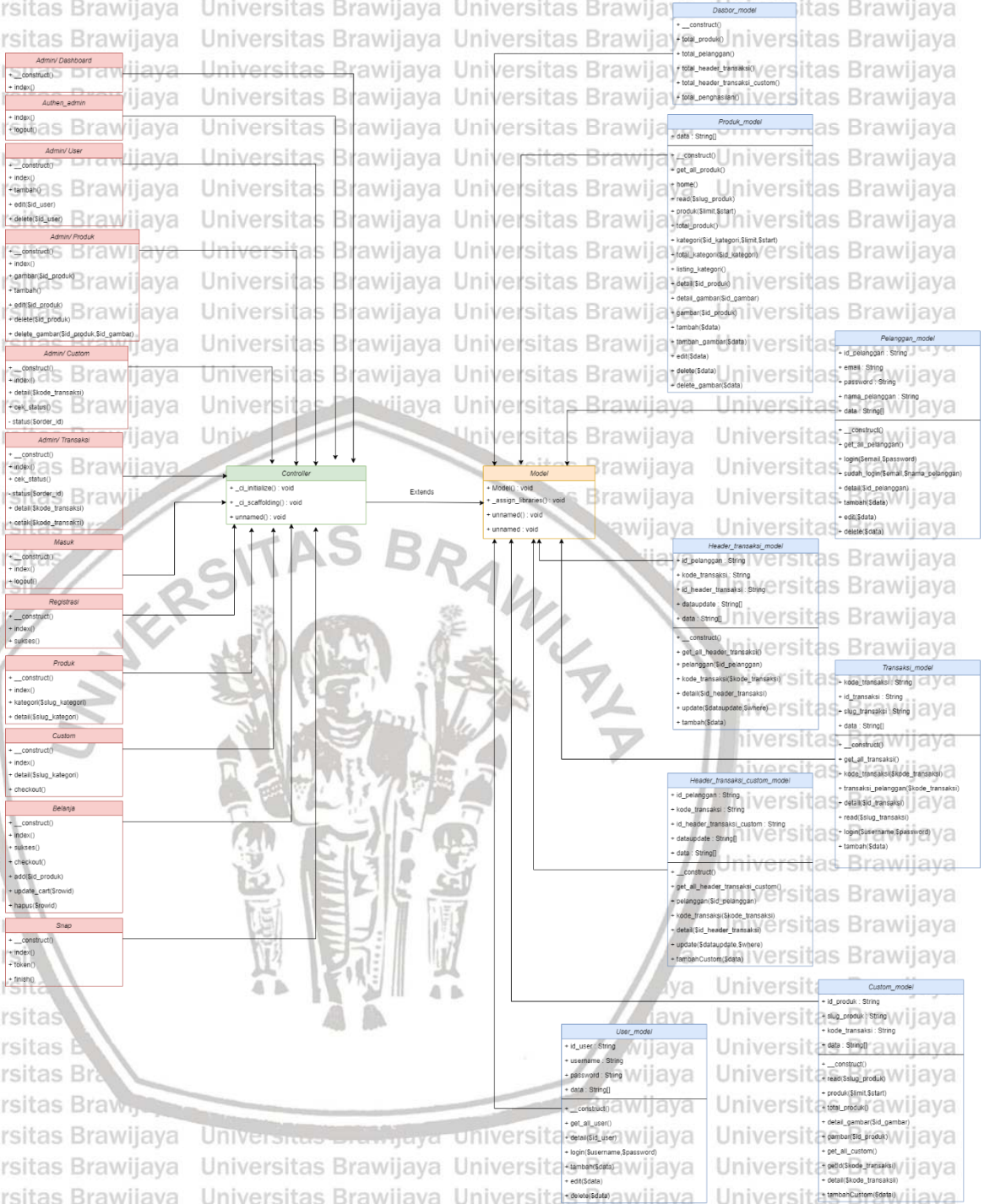


Gambar 4. 42 Sequence Diagram Mencetak Detail Pesanan

Sumber : (Penulis, 2021)

4.8 Class Diagram

Class diagram merupakan struktur sistem yang digambarkan melalui kelaskelas yang mempunyai atribut, *method*, dan relasi antar kelas. Kelas-kelas yang dirancang nantinya akan diimplementasikan ke *source code* pada *Framework CodeIgniter*. *Class Diagram* yang dibuat nantinya memperlihatkan relasi antara *Controller* dan *Model* pada *Framework CodeIgniter*. *Class Diagram* dirancang menjadi dua jenis berdasarkan kelas abstrak yang di-*extend* yaitu kelas *Controller* dan kelas *Model* yang kelas turunannya saling berelasi. Pada gambar 4.43, terdapat 12 kelas turunan dari kelas *Controller*, yaitu Dashboard, Authen_admin, User, Produk(Admin), Custom(Admin), Transaksi(Admin), Masuk, Registrasi, Belanja, Produk, Custom, dan Snap. Ada 8 kelas yang merupakan turunan dari kelas *Model*, yaitu Dasbor_model, Pelanggan_model, Users_model, Produk_model, Header_transaksi_model, Header_transaksi_custom_model, transaksi_model, custom_model,. Masing-masing memiliki fungsionalitas yang berbeda beda yang merupakan gambaran keseluruhan sistem.



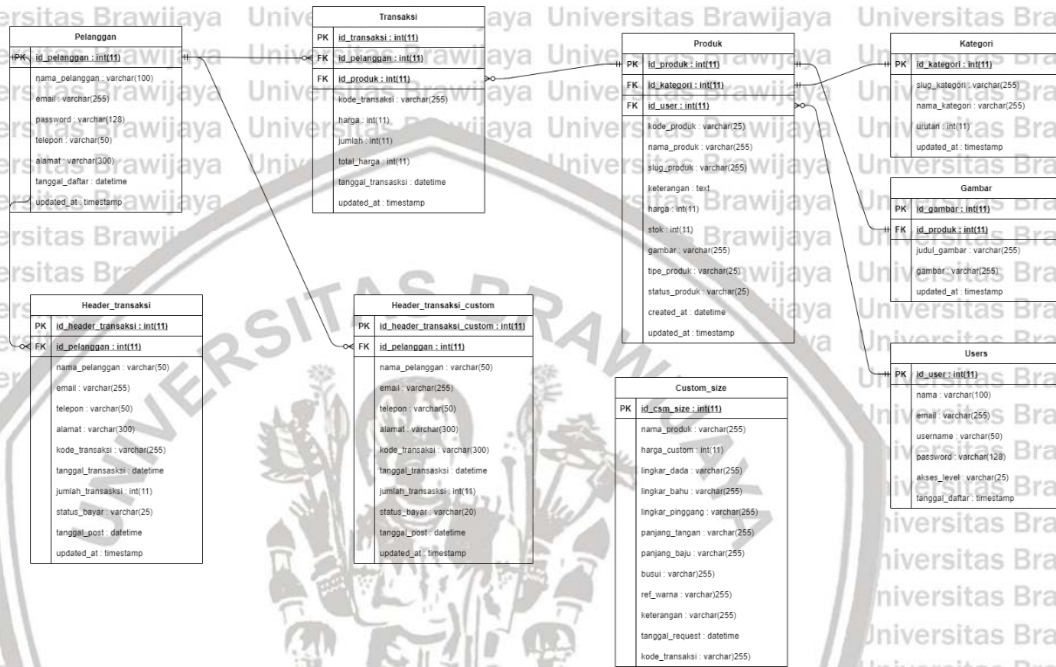
Gambar 4. 43 Class Diagram Website E-commerce Butik Kebaya Iffah

Sumber : (Penulis, 2021)



4.9 Pemodelan Physical Data Model (PDM)

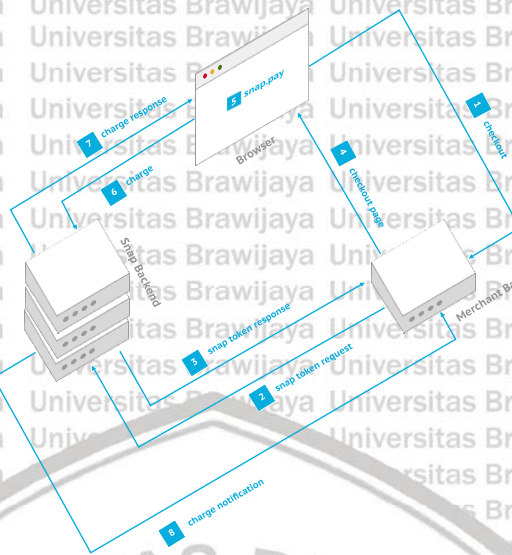
PDM atau *Physical Data Model* adalah diagram yang memodelkan hubungan atau relasi antar objek data sebagai rancangan dalam membangun sebuah *database*. Berikut adalah Physical Data Model Website E-commerce butik kebaya iffah yang dijelaskan pada gambar 4.44. yang terdiri dari 9 tabel, yakni tabel pelanggan, tabel users, tabel kategori, tabel produk, tabel gambar, tabel header_transaksi, tabel transaksi, tabel header_transaksi_custom, dan tabel custom_size.



Gambar 4. 44 Physical Data Model Website E-commerce Butik Kebaya Iffah

Sumber : (Penulis, 2021)

4.10 Perancangan Integrasi *Payment Gateway* Midtrans



Gambar 4. 45 Perancangan Integrasi *Payment Gateway* Midtrans

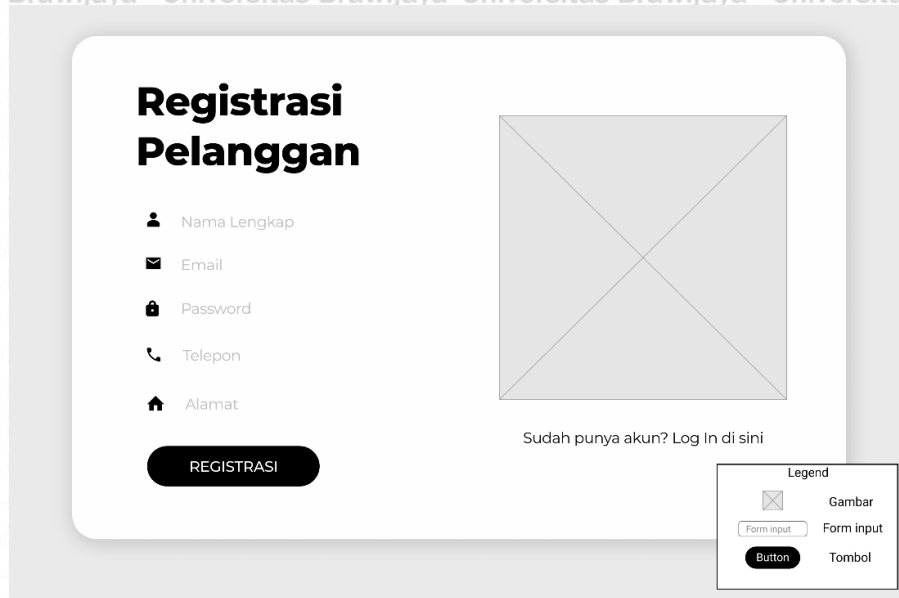
Sumber : (Midtrans,2021)

Pada bagian ini dijelaskan mengenai perancangan integrasi *payment gateway* yang akan digunakan pada website e-commerce butik kebaya iffah. Pada penelitian ini terdapat Batasan sistem hanya akan diintegrasikan pada lingkup API sandbox. gambar 4.45 diatas merupakan gambaran perancangan indegrasi API *Payment gateway* midtrans sistem, dimana ketika user melakukan checkout, sistem akan meminta token kepada midtrans, yang kemudian akan di response dan ditampilkan snap payment. setelah itu user dapat memverifikasi detail transaksi dan melakukan pembayaran. dan dilanjutkan dengan backend midtrans akan memproses dan mengirimkan json berisi *transaction detail* beserta status pembayaran.

4.11 Perancangan *User Interface*

User Interface merupakan tampilan yang berhubungan langsung dengan pengguna untuk melakukan interaksi dengan sistem. Perancangan User Interface digunakan sebagai acuan untuk membuat tampilan dari Website E-commerce Butik Kebaya Iffah.

4.11.1 User Interface Register

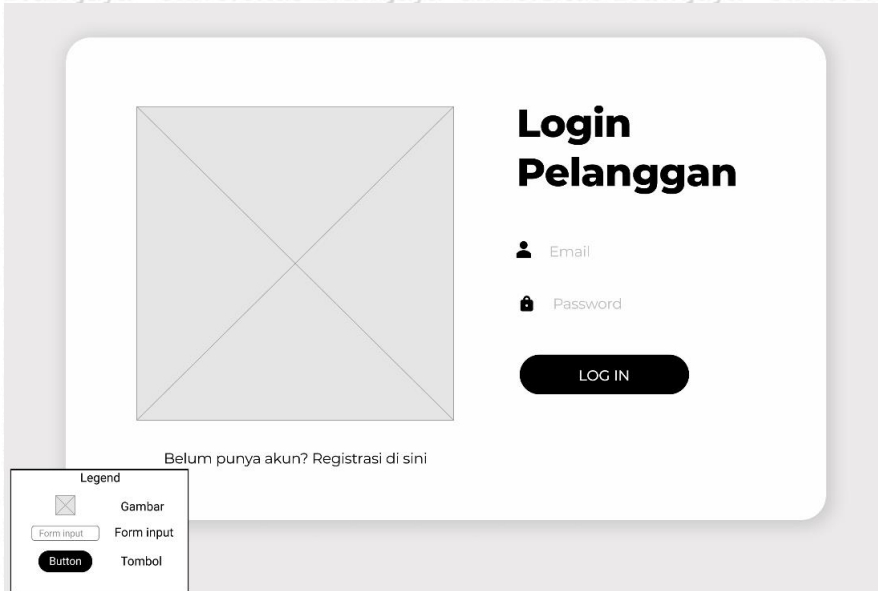


Gambar 4. 46 Perancangan User Interface Registrasi Pelanggan

Sumber : (Penulis, 2021)

Perancangan dari *user interface* halaman register yang ditunjukkan pada Gambar 4.46 ini akan menjadi acuan dalam mengimplementasikan user interface halaman register. Pada bagian atas form terdapat text berjudul 'Registrasi Pelanggan'. Terdapat komponen form registrasi berupa form input untuk data diri pengguna agar dapat terdaftar pada sistem. Juga terdapat tombol 'Registrasi' dibawah form input, yang apabila ditekan dan terverifikasi oleh sistem, maka registrasi pelanggan berhasil. Dan pada sebelah kiri form login terdapat gambar dari logo butik kebaya iffah, dan bagian bawah gambar logo terdapat link menuju halaman login apabila pelanggansudah terdaftar dan ingin masuk pada sistem.

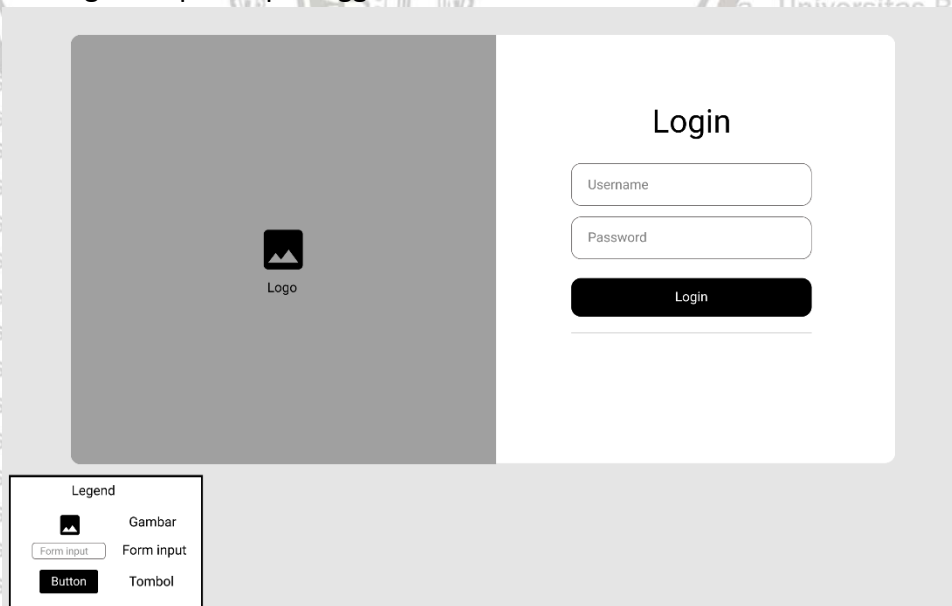
4.11.2 User Interface Login



Gambar 4. 47 Perancangan User Interface Login Pelanggan

Sumber : (Penulis, 2021)

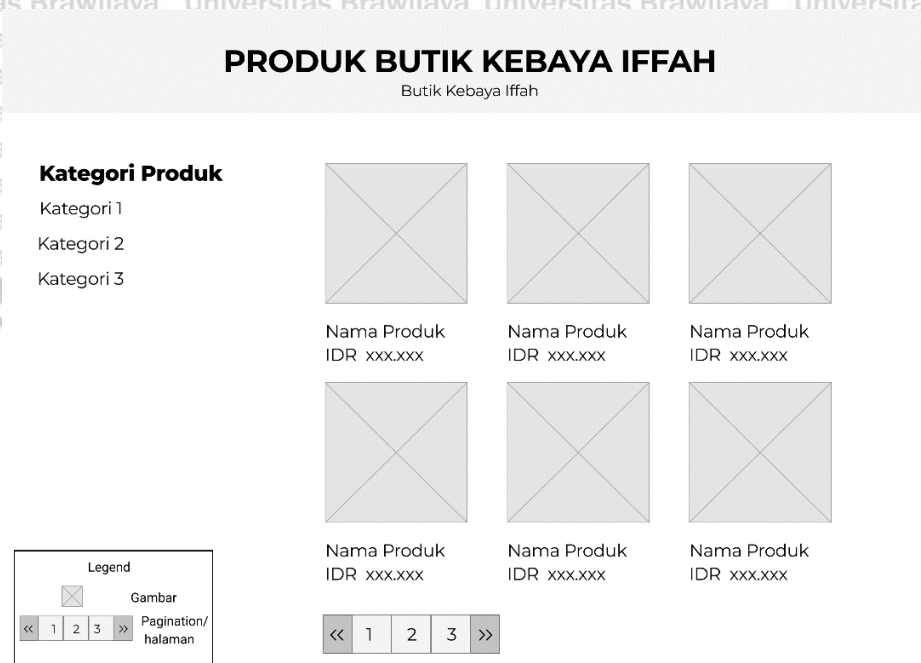
Perancangan dari user interface halaman login pelanggan yang ditunjukkan pada Gambar 4.47. ini akan menjadi acuan dalam mengimplementasikan user interface halaman Login pelanggan. Pada bagian atas form terdapat text berjudul 'Login Pelanggan'. Terdapat komponen form login berupa form input untuk menginputkan *email* dan *password*. Juga terdapat tombol 'Login' dibawah form input, yang apabila ditekan saat *email* dan *password* pelanggan terverifikasi oleh sistem, maka akan ditampilkan halaman dashboard pelanggan. Dan pada sebelah kiri form login terdapat gambar dari logo butik kebaya iffah, dan bagian bawah gambar logo terdapat link menuju halaman register apabila pelanggan belum mendaftar.



Gambar 4. 48 Perancangan User Interface Login Admin

Perancangan dari user interface halaman login admin yang ditunjukkan pada Gambar 4.48. ini akan menjadi acuan dalam mengimplementasikan user interface halaman Login admin. Pada bagian atas form terdapat text berjudul 'Login'. Terdapat komponen form login berupa form input untuk menginputkan *username* dan *password*. Juga terdapat tombol 'Login' dibawah form input, yang apabila ditekan saat *username* dan *password admin* terverifikasi oleh sistem, maka akan ditampilkan halaman dashboard admin. Dan pada sebelah kiri form login terdapat card berupa gambar dari logo butik kebaya iffah.

4.11.3 User Interface Melihat Daftar Produk

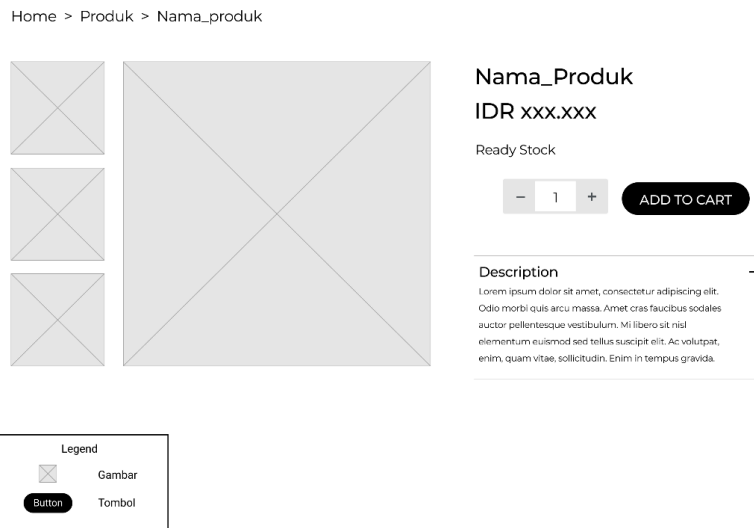


Gambar 4. 49 Perancangan User Interface Melihat Daftar Produk

Sumber : (Penulis, 2021)

Perancangan dari user interface halaman melihat daftar produk yang ditunjukkan pada Gambar 4.49. Pada bagian atas terdapat text berjudul 'Produk Butik Kebaya Iffah'. Terdapat komponen pola yang berulang mengenai produk yaitu terdiri dari gambar produk dan terdapat nama produk serta harga produk tersebut. Dan pada bagian sebelah kiri terdapat macam macam kategori yang tersedia, dan apabila pelanggan menekan salah satunya maka sistem menampilkan daftar produk kategori terpilih. Pada bagian bawah halaman daftar produk terdapat pagination/ halaman dari produk yang ditampilkan agar pelanggan dapat melihat dan memilih hingga halaman yang tersedia.

4.11.4 User Interface Melihat Detail Produk



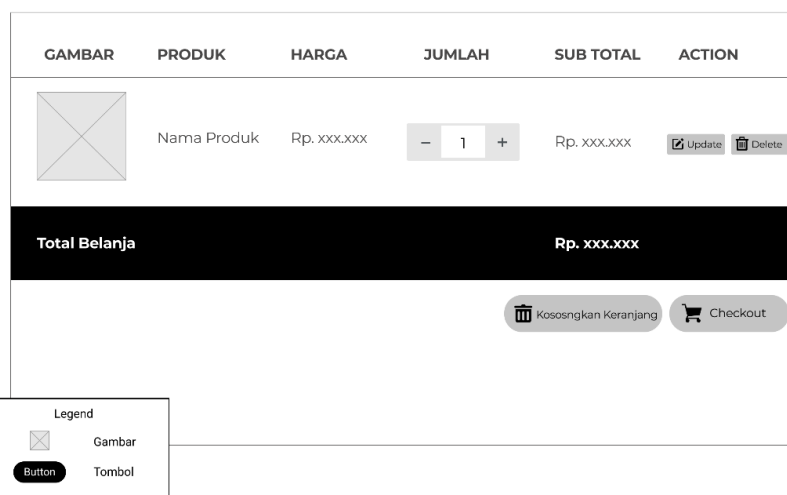
Gambar 4. 50 Perancangan User Interface Melihat Detail Produk

Sumber : (Penulis, 2021)

Perancangan dari user interface halaman melihat detail produk yang ditunjukkan pada Gambar 4.50. Pada bagian atas terdapat breadcrumb yang menunjukkan alamat detail produk. Terdapat komponen Gambar produk yang terpilih dan deret menurun beberapa gambar lain produk tersebut pada sebelah kiri. Terdapat text bold mengenai produk yaitu nama produk serta harga produk tersebut beserta informasi ketersediaan barang. Dan pada bagian bawahnya terdapat tombol untuk menambah atau mengurangi jumlah item yang akan dibeli, dan tombol 'add to cart' untuk menambahkan produk ke keranjang, serta bagian paling bawah dijelaskan mengenai deskripsi produk.

4.11.5 User Interface Mengelola Keranjang Belanja

Keranjang Belanja

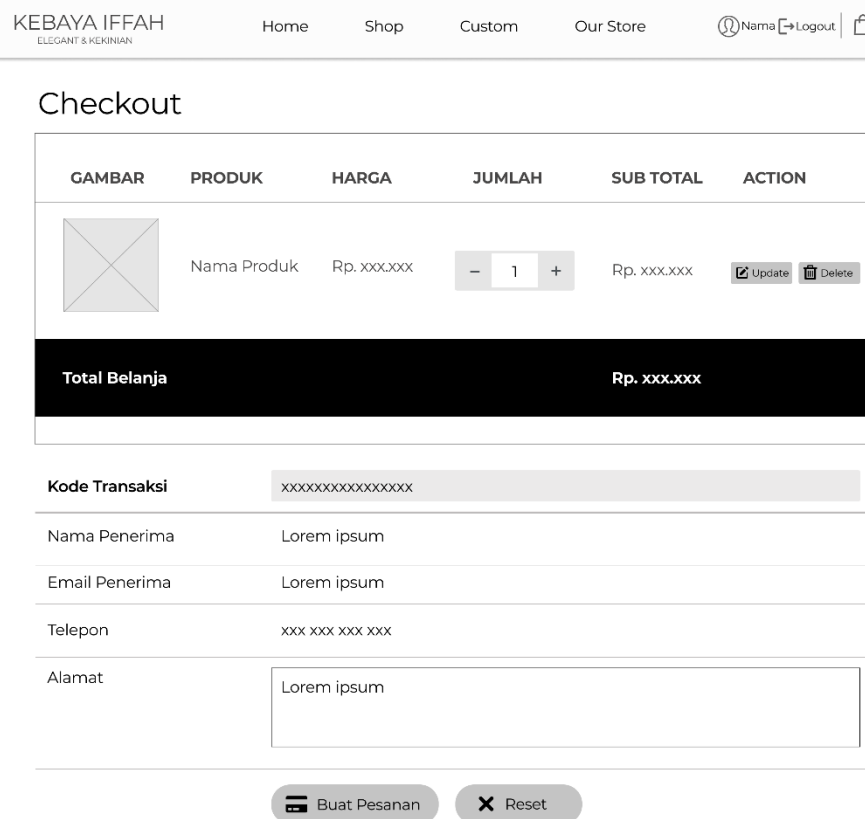


Gambar 4. 51 Perancangan User Interface Keranjang Belanja

Sumber : (Penulis, 2021)

Perancangan dari *user interface* halaman mengelola keranjang belanja yang ditunjukkan pada Gambar 4.51. Pada bagian atas terdapat text 'Keranjang belanja'. Terdapat komponen tabel dengan kolom berisi, gambar, nama produk, harga, tombol untuk menambah atau mengurangi jumlah item yang akan dibeli, subtotal yaitu jumlah dari harga produk dikali jumlah item, dan terdapat dua tombol action yaitu update dan delete. 'update' untuk memperbarui subtotal produk dari update item, dan 'delete' untuk menghapus produk terpilih dari baris tabel keranjang. Pada bagian bawah terdapat subtotal keseluruhan dari seluruh produk yang ada pada tabel keranjang. serta bagian paling bawah terdapat dua tombol, 'kosongkan keranjang' digunakan untuk menghapus seluruh produk pada keranjang, dan 'checkout' digunakan untuk melanjutkan proses transaksi ke halaman selanjutnya checkout untuk mengisi pengiriman.

4.11.6 User Interface Melakukan Checkout



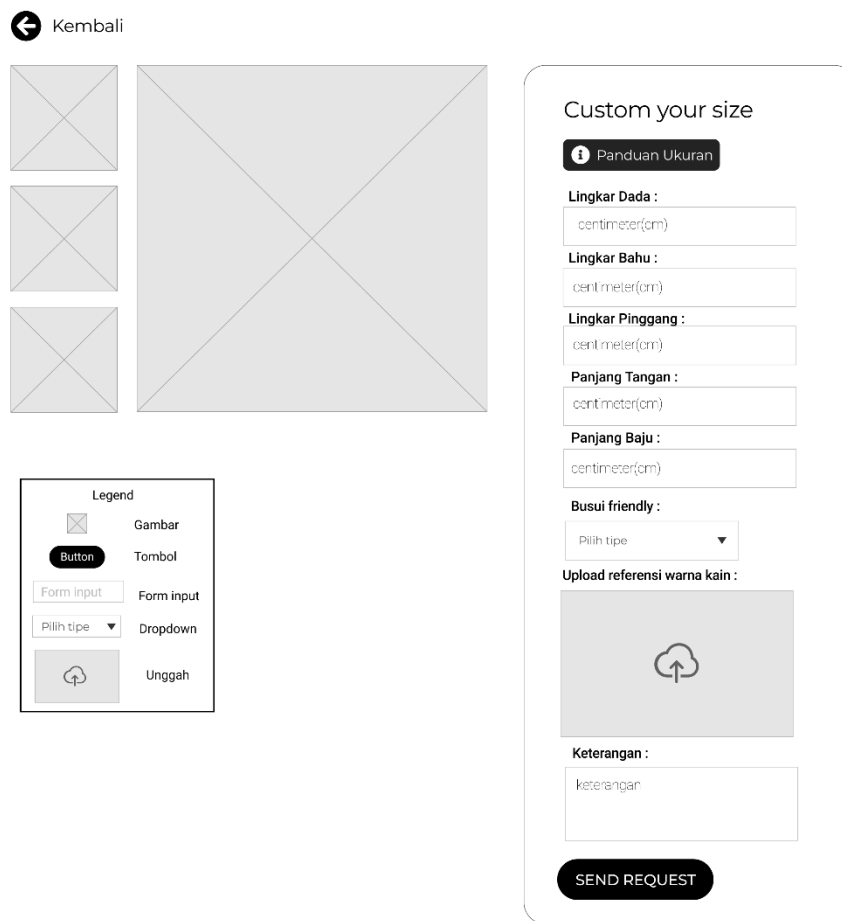
Gambar 4. 52 Perancangan User Interface Melakukan Checkout

Sumber : (Penulis, 2021)

Perancangan dari *user interface* halaman membuat pesanan yang ditunjukkan pada Gambar 4.52. Pada bagian atas terdapat text 'checkout'. Terdapat komponen tabel keranjang belanja seperti pada *user interface* keranjang belanja. Namun, pada halaman checkout ini menampilkan form yang terdiri dari text bold kode transaksi, yang otomatis tercatat dan tidak dapat diganti,

setelahnya dilanjutkan form input untuk proses pengiriman barang yang dipesan, sesuai biodata pelanggan dan alamat kirim sebagai tujuan. Pada bagian bawah terdapat dua tombol yaitu 'Reset' untuk menghapus field yang sudah terisi apabila user ingin mengganti seluruhnya, dan ada tombol 'Buat Pesanan' yang jika pelanggan menekan tombol tersebut maka akan dilanjutkan pada halaman integrasi midtrans untuk dilakukan proses pembayaran dan nantinya pesanan sudah tercatat pada admin.

4.11.7 User Interface Melakukan Custom Order



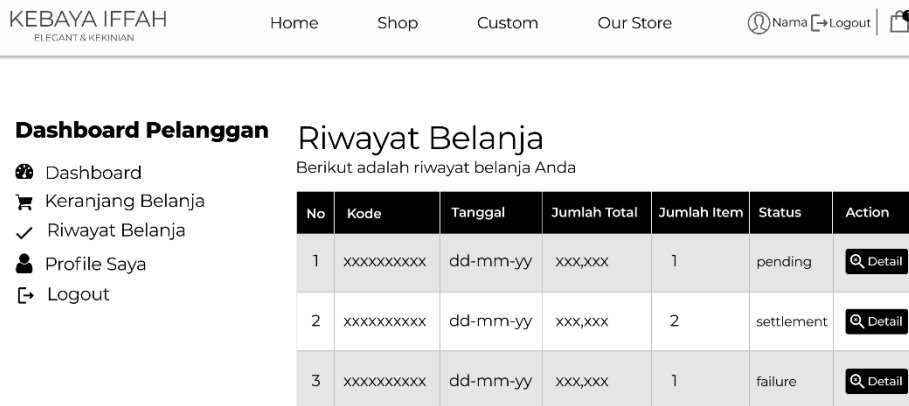
Gambar 4. 53 Perancangan User Interface Melakukan Custom Order

Sumber : (Penulis, 2021)

Perancangan dari user interface halaman membuat pesanan yang ditunjukkan pada Gambar 4.53. Pada bagian atas terdapat text 'checkout'. Terdapat komponen tabel keranjang belanja seperti pada user interface keranjang belanja. Namun, pada halaman checkout ini menampilkan form yang terdiri dari text bold kode transaksi, yang otomatis tercatat dan tidak dapat diganti, setelahnya dilanjutkan form input untuk proses pengiriman barang yang dipesan, sesuai biodata pelanggan dan alamat kirim sebagai tujuan. Pada bagian bawah terdapat dua tombol yaitu 'Reset' untuk menghapus field yang sudah terisi apabila user ingin mengganti seluruhnya, dan ada tombol 'Buat Pesanan'

yang jika pelanggan menekan tombol tersebut maka akan dilanjutkan pada halaman integrasi midtrans untuk dilakukan proses pembayaran dan nantinya pesanan sudah tercatat pada admin.

4.11.8 User Interface Melihat Riwayat Belanja

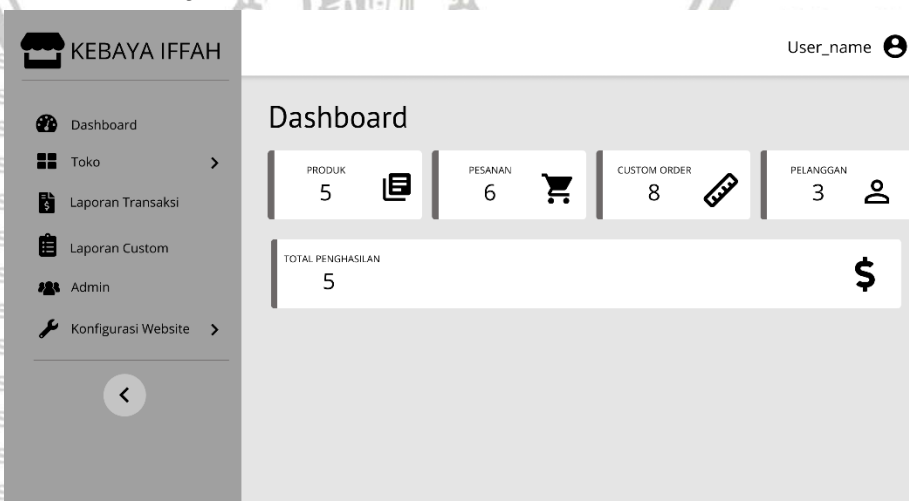


Gambar 4. 54 Perancangan *User Interface* Melihat Riwayat Belanja

Sumber : (Penulis, 2021)

Perancangan dari user interface halaman melihat riwayat belanja yang ditunjukkan pada Gambar 4.54. Riwayat belanja dapat diakses pada halaman dashboard pelanggan, sub menu riwayat transaksi. Dan akan ditampilkan tabel riwayat transaksi yang telah pelanggan lakukan sebelumnya, terdapat kolom kode transaksi, tanggal transaksi dilakukan, jumlah total, jumlah item, dan status pembayaran produk, serta terdapat tombol action 'Detail' jika tombol ditekan maka akan melanjutkan pada halaman detail transaksi terpilih untuk ditampilkan informasi lebih rinci dari transaksi tersebut.

4.11.9 User Interface Melihat Halaman Dashboard

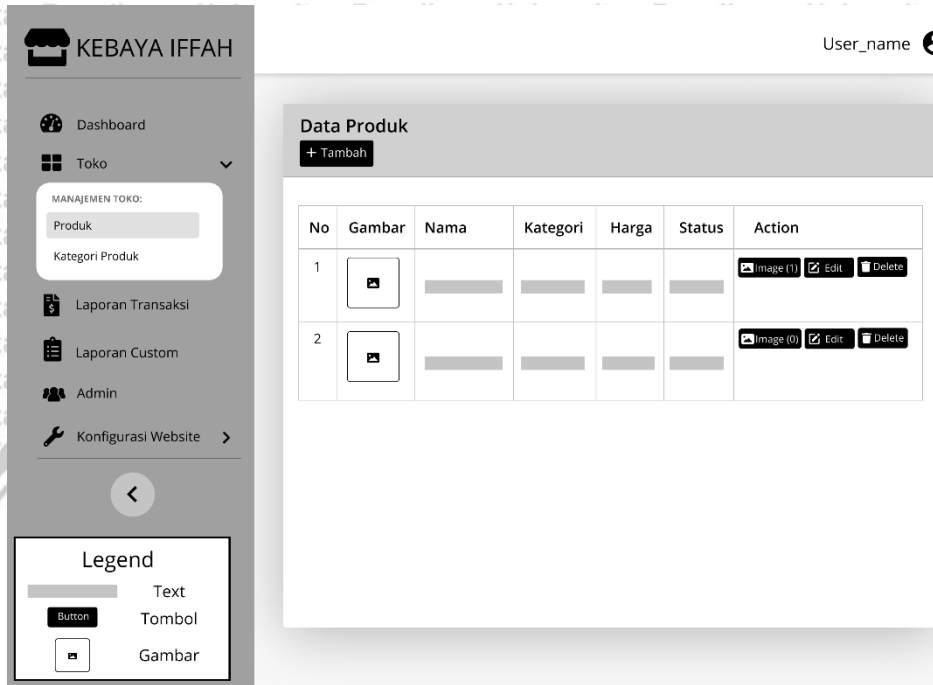


Gambar 4. 55 Perancangan *User Interface* Melihat Dashboard

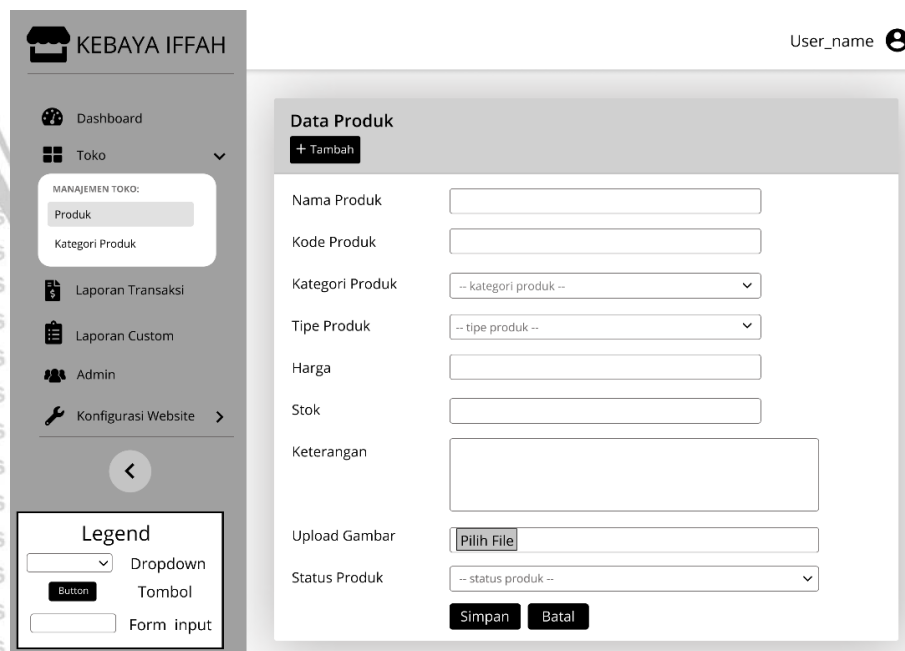
Sumber : (Penulis, 2021)

Perancangan dari *user interface* halaman melihat dashboard admin yang ditunjukkan pada Gambar 4.55. Dashboard admin dapat diakses pada halaman admin saat sudah melakukan login, halaman ini berisi informasi tentang sistem yang ditampilkan pada card card.

4.11.10 User Interface Mengelola Produk



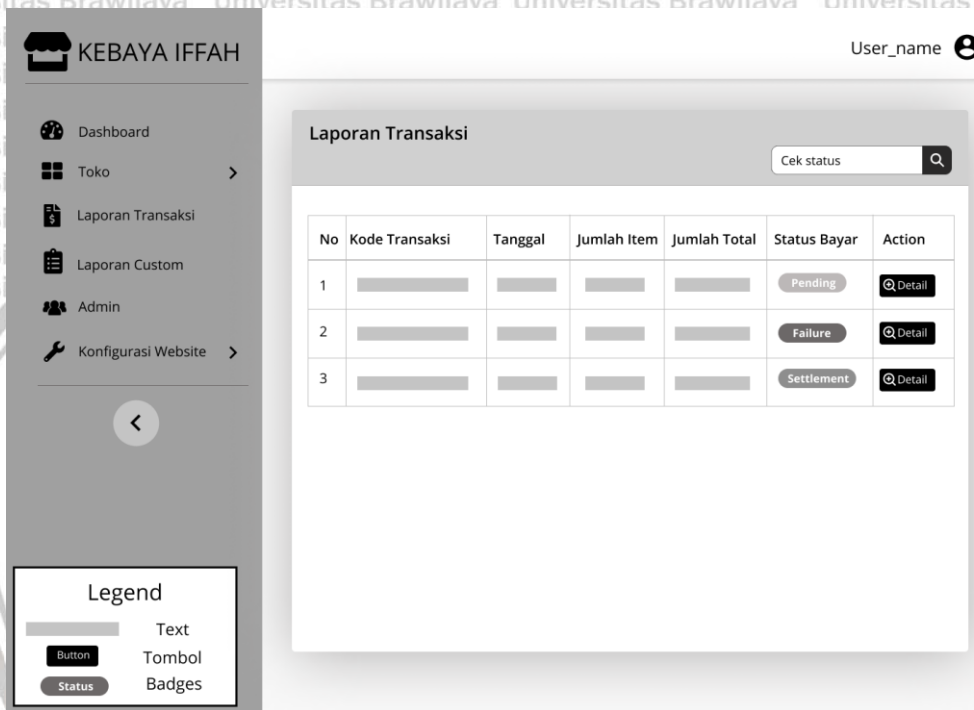
Gambar 4. 56 Perancangan *User Interface* Mengelola Produk



Gambar 4. 57 Perancangan *User Interface* Mengelola Produk (Form Tambah)

Perancangan dari user interface halaman mengelola produk yang ditunjukkan pada Gambar 4.56. kelola produk dapat diakses pada halaman dashboard pelanggan, sub menu produk. Dan akan ditampilkan tabel daftar produk, yang terdiri dari kolom gambar, nama, kategori, harga, status dan 3 tombol action, yaitu image untuk melakukan pengelolaan gambar produk, edit untuk edit produk, hapus untuk hapus produk. Pada bagian atas terdapat tombol tambah yang akan menampilkan form input menambahkan produk seperti gambar 4.57.

4.11.11 User Interface Melihat Data Pesanan

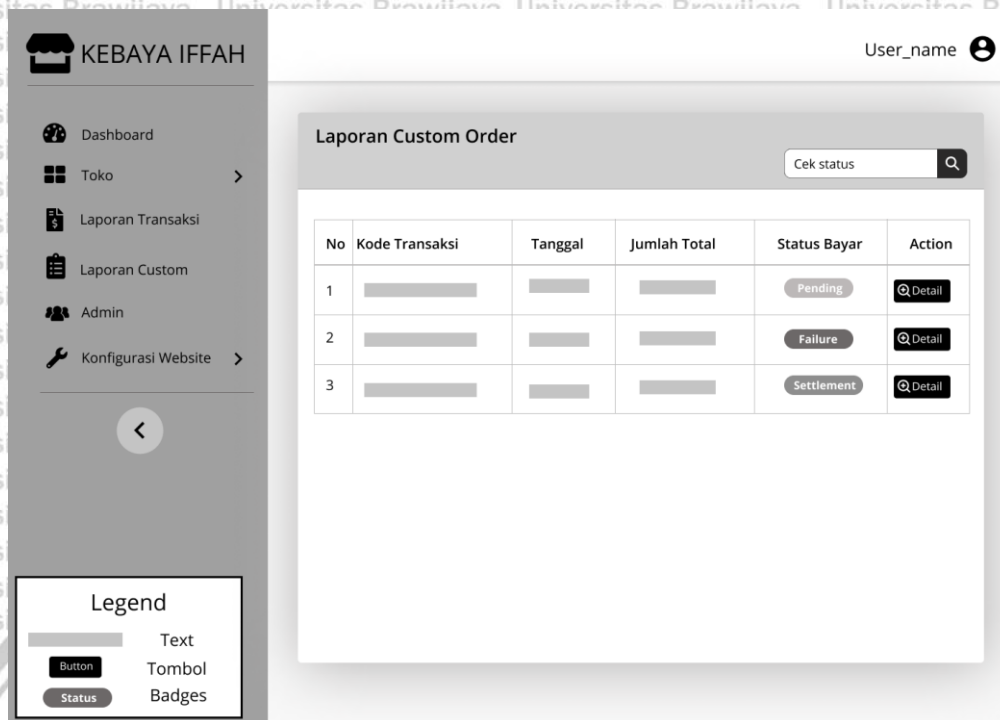


Gambar 4. 58 Perancangan User Interface Melihat Data Pesanan dan Cek Status

Sumber : (Penulis, 2021)

Perancangan dari user interface halaman melihat data pesanan yang ditunjukkan pada Gambar 4.58. Riwayat belanja dapat diakses pada halaman dashboard pelanggan, sub menu riwayat transaksi. Dan akan ditampilkan tabel riwayat transaksi yang telah pelanggan lakukan sebelumnya, terdapat kolom kode transaksi, tanggal transaksi dilakukan, jumlah total, jumlah item, dan status pembayaran produk, serta terdapat tombol action 'Detail' jika tombol ditekan maka akan melanjutkan pada halaman detail transaksi terpilih untuk ditampilkan informasi lebih rinci dari transaksi tersebut.

4.11.12 User Interface Melihat Data Custom Order

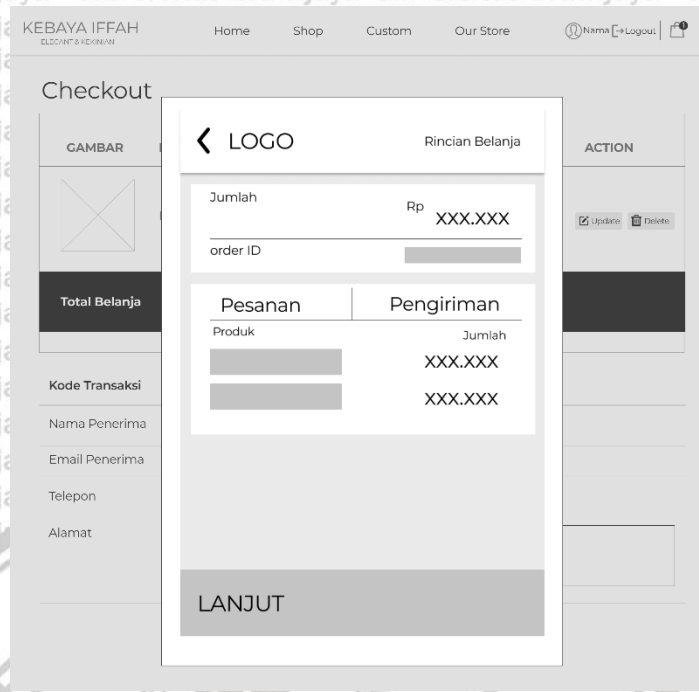


Gambar 4. 59 Perancangan User Interface Melihat Data Custom Order dan Cek Status

Sumber : (Penulis, 2021)

Perancangan dari *user interface* halaman melihat data custom order yang ditunjukkan pada Gambar 4.59. dapat diakses pada halaman dashboard pelanggan, sub menu riwayat transaksi. Dan akan ditampilkan tabel riwayat transaksi yang telah pelanggan lakukan sebelumnya, terdapat kolom kode transaksi, tanggal transaksi dilakukan, jumlah total, jumlah item, dan sytatus pembayaran produk, serta terdapat tombol action 'Detail' jika tombol ditekan maka akan melanjutkan pada halaman detail transaksi terpilih untuk ditampilkan informasi lebih rinci dari transaksi tersebut.

4.11.13 User Interface Pembayaran



Gambar 4. 61 Perancangan User Interface Pembayaran Midtrans

Perancangan dari user interface halaman pembayaran yang ditunjukkan pada Gambar 4.61. dapat diakses setelah pelanggan sudah berada pada halaman checkout, baik produk ready maupun custom, saat menekan tombol buat pesanan akan ditampilkan snap payment dari midtrans yang berisikan atribut pembayaran dan proses lanjutnya seperti memilih metode hingga selesai.

BAB 5 IMPLEMENTASI

Pada bab ini akan dilakukan proses pengembangan sistem dengan menerapkan rancangan sistem yang sudah dirancang pada bab sebelumnya. Pada Bab ini akan dijelaskan mengenai Implementasi pada Website E-commerce Butik Kebaya Iffah yang meliputi :

1. Spesifikasi lingkungan implementasi
2. Konfigurasi Penggunaan API Midtrans
3. Implementasi *user interface*
4. Implementasi *database*
5. Implementasi kode program

5.1 Spesifikasi Lingkungan Implementasi

Bagian ini akan menjelaskan mengenai spesifikasi lingkungan implementasi yang digunakan dalam pembuatan Website E-commerce Butik Kebaya Iffah. Terdapat dua spesifikasi yang akan dijelaskan yaitu spesifikasi perangkat keras dan spesifikasi perangkat lunak.

5.1.1 Spesifikasi Perangkat Keras

Kebutuhan perangkat keras yang digunakan untuk membangun sistem antara lain memiliki spesifikasi yang dijelaskan pada Tabel 5.1 .

Tabel 5. 1 Spesifikasi Lingkungan Perangkat Keras

Laptop Processor	Intel® Core™ i7-7500U CPU @ 1.70GHz (4 CPUs), ~2.4GHz
Memory (RAM)	4GB
Harddisk	1TB
Graphic Card	Intel HD Graphics 620

Sumber : (Penulis, 2021)

5.1.2 Spesifikasi Perangkat Lunak

Kebutuhan perangkat lunak yang digunakan untuk membangun sistem antara lain memiliki spesifikasi yang dijelaskan pada Tabel 5.2.

Tabel 5. 2 Spesifikasi Lingkungan Perangkat Lunak

Sistem operasi	Windows 10 Pro 64-bit
Text Editor	Visual Code Studio versi 1.57.0
Programming Language	PHP,CSS,HTML dan Javascript

Framework	CodeIgniter
Web Server	XAMPP versi 7.2.11-0
Editor Dokumentasi	Draw.io

Sumber : (Penulis, 2021)

5.2 Konfigurasi API *sandbox Payment Gateway Midtrans*

Bagian ini akan menjelaskan langkah-langkah untuk melakukan konfigurasi API *Sandbox Payment Gateway* Midtrans dengan file codeigniter agar dapat digunakan pada *website e-commerce* butik kebaya iffah. Berikut adalah langkah-langkah yang harus dilakukan :

1. Download library midtrans untuk bahasa pemrograman php
2. Dilakukan penggabungan file library midtrans yang telah didownload dengan file codeIgniter yang kita gunakan
3. Disisi lain, sudah dilakukan registrasi pada midtrans, untuk mendapatkan server-key dan client-key
4. Dilakukan konfigurasi pada controller snap, dengan menginputkan server-key yang sudah didapat pada proses sebelumnya

```
$params = array('server_key' => 'Server-Key', 'production' => false);
$this->load->library('midtrans');
$this->midtrans->config($params);
```

Gambar 5. 1 Konfigurasi Server-Key

Sumber : (Penulis,2021)

5. Dilakukan *frontend reuirement* pada folder view yang akan digunakan dengan menginputkan client-key yang sudah didapat pada proses sebelumnya

```
<script type="text/javascript"
src="https://app.sandbox.midtrans.com/snap/snap.js"
data-client-key="client-key"></script>
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script>
```

Gambar 5. 2 *Frontend Reuirement Client-Key*

Sumber : (Penulis,2021)

5.3 Implementasi *User Interface*

Subbab ini memaparkan hasil implementasi dari *user interface* yang mengacu pada perancangan sebelumnya.

5.3.1 *User Interface Register*

Gambar 5.1 menunjukkan implementasi dari *user interface* halaman Registrasi pelanggan yang mengacu pada perancangan *user interface* halaman

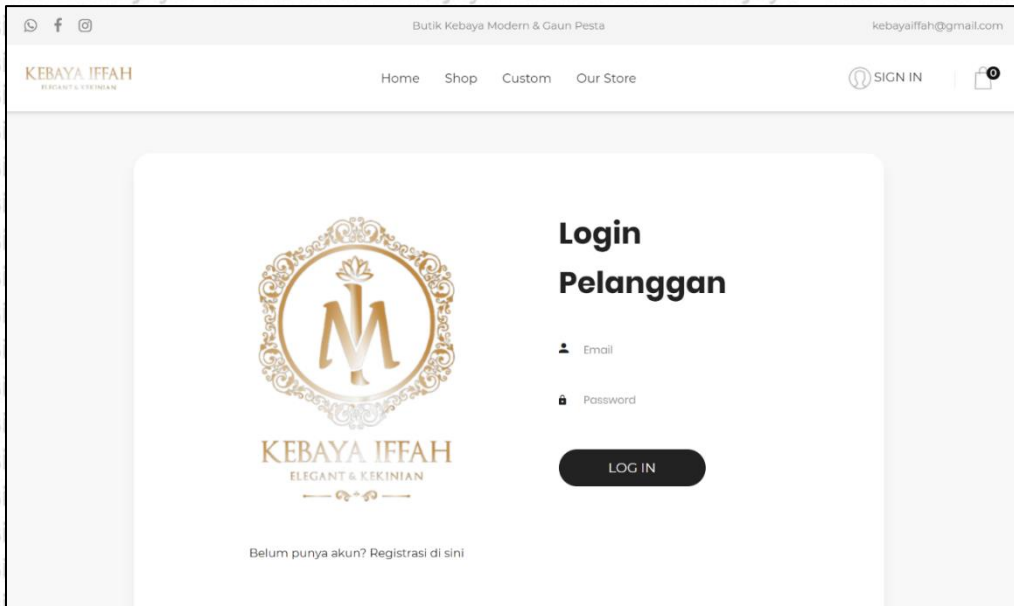
Register, Terdapat form input seperti nama, email, password, telepon, dan alamat untuk diisi oleh pelanggan.

Gambar 5. 3 Implementasi *User Interface* Halaman Registrasi

Sumber : (Penulis,2021)

5.3.2 *User Interface* Login Pelanggan

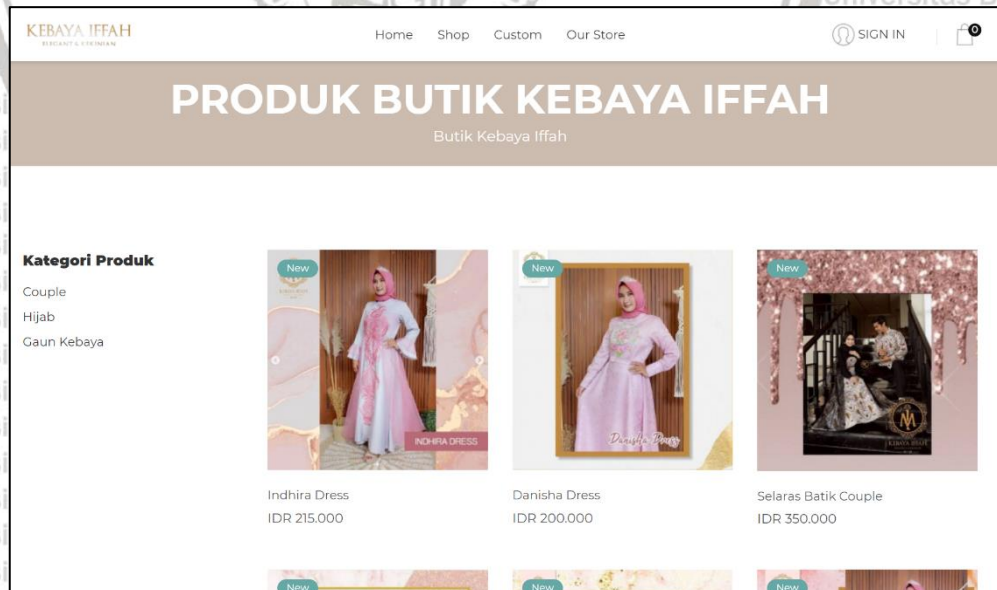
Gambar 5.2 menunjukkan implementasi dari *user interface* halaman Login Admin yang mengacu pada perancangan *user interface* halaman Login Admin, Terdapat form input seperti email dan password untuk diisi oleh pelanggan



Gambar 5. 4 Implementasi *User Interface* Halaman Login Pelanggan
 Sumber : (Penulis,2021)

5.3.3 *User Interface* Melihat Daftar Produk

Gambar 5.3 menunjukkan implementasi dari *user interface* halaman Melihat Daftar Produk yang mengacu pada perancangan *user interface* halaman melihat daftar produk, Terdapat banner title, dan menampilkan produk sesuai kategori yang dipilih.



Gambar 5. 5 Implementasi *User Interface* Halaman Melihat Daftar Produk
 Sumber : (Penulis,2021)

5.3.4 User Interface Melihat Detail Produk

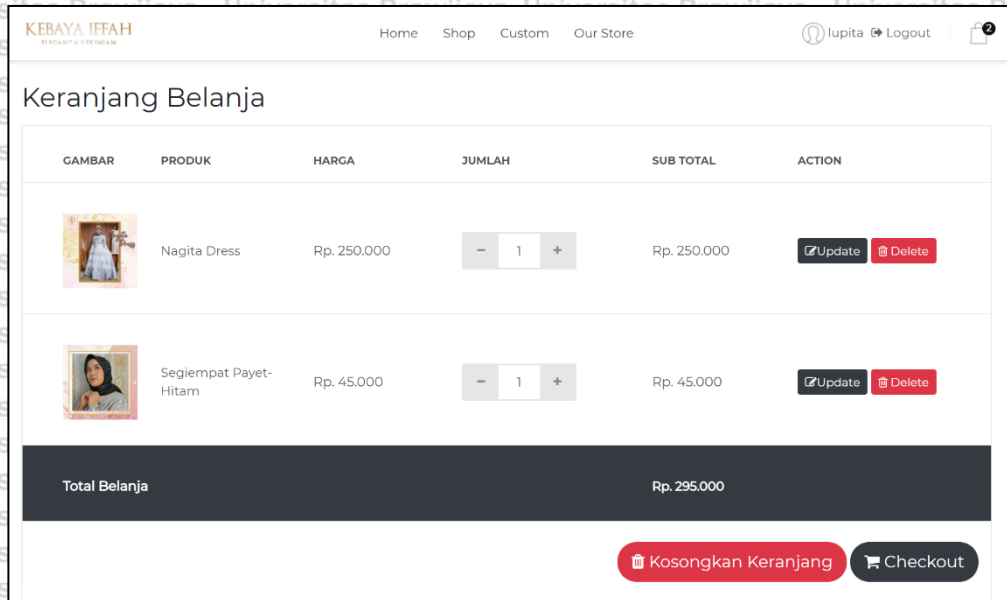
Gambar 5.4 menunjukkan implementasi dari *user interface* halaman Melihat Detail Produk yang mengacu pada perancangan *user interface* halaman melihat detail produk, Terdapat judul nama produk, harga, gambar dan deskripsi serta tombol jumlah dan tambah ke keranjang.



Gambar 5. 6 Implementasi *User Interface* Halaman Melihat Detail Produk
Sumber : (Penulis,2021)

5.3.5 User Interface Keranjang Belanja

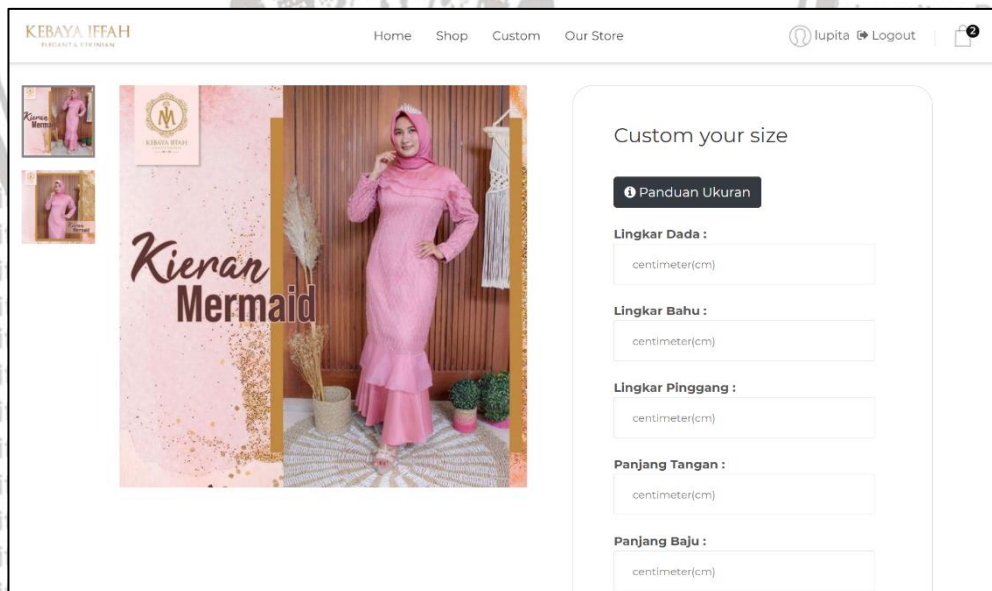
Gambar 5.5 menunjukkan implementasi dari *user interface* halaman keranjang belanja yang mengacu pada perancangan *user interface* halaman Login keranjang belanja, menampilkan isi keranjang belanja dan tombol untuk mengelolanya.



Gambar 5. 7 Implementasi User Interface Halaman Keranjang Belanja
 Sumber : (Penulis,2021)

5.3.6 User Interface Custom Order

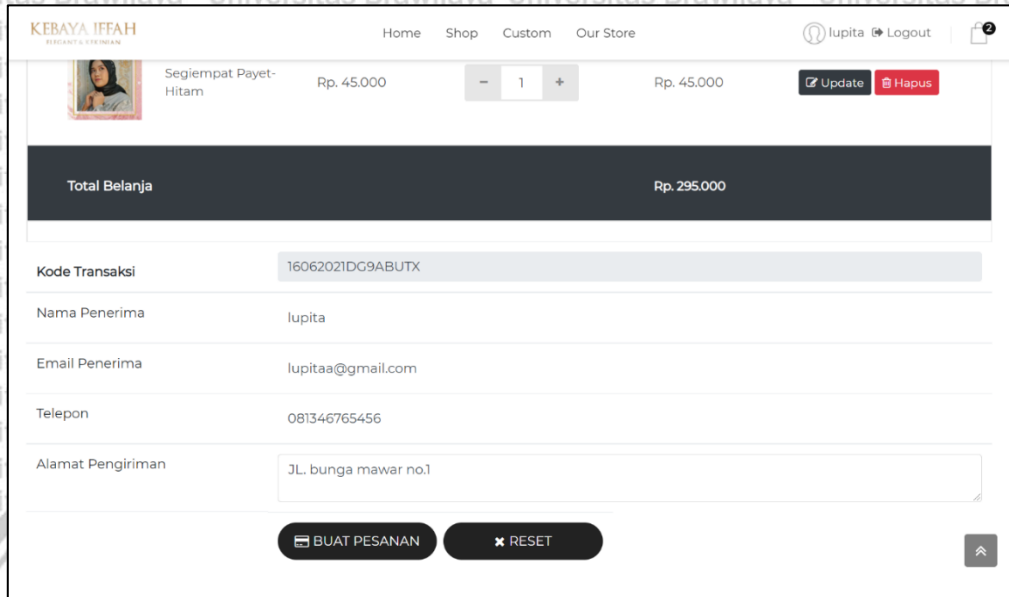
Gambar 5.6 menunjukkan implementasi dari *user interface* halaman Custom yang mengacu pada perancangan *user interface* halaman custom, Terdapat form input mengenai atribut yang akan digunakan untuk custom, dan terdapat tombol modal yang menampilkan cara mengukur ukuran.



Gambar 5. 8 Implementasi User Interface Halaman Custom Order
 Sumber : (Penulis,2021)

5.3.7 User Interface Checkout

Gambar 5.7 menunjukkan implementasi dari *user interface* halaman checkout yang mengacu pada perancangan *user interface* halaman checkout, Terdapat form input yang digunakan untuk pengiriman.



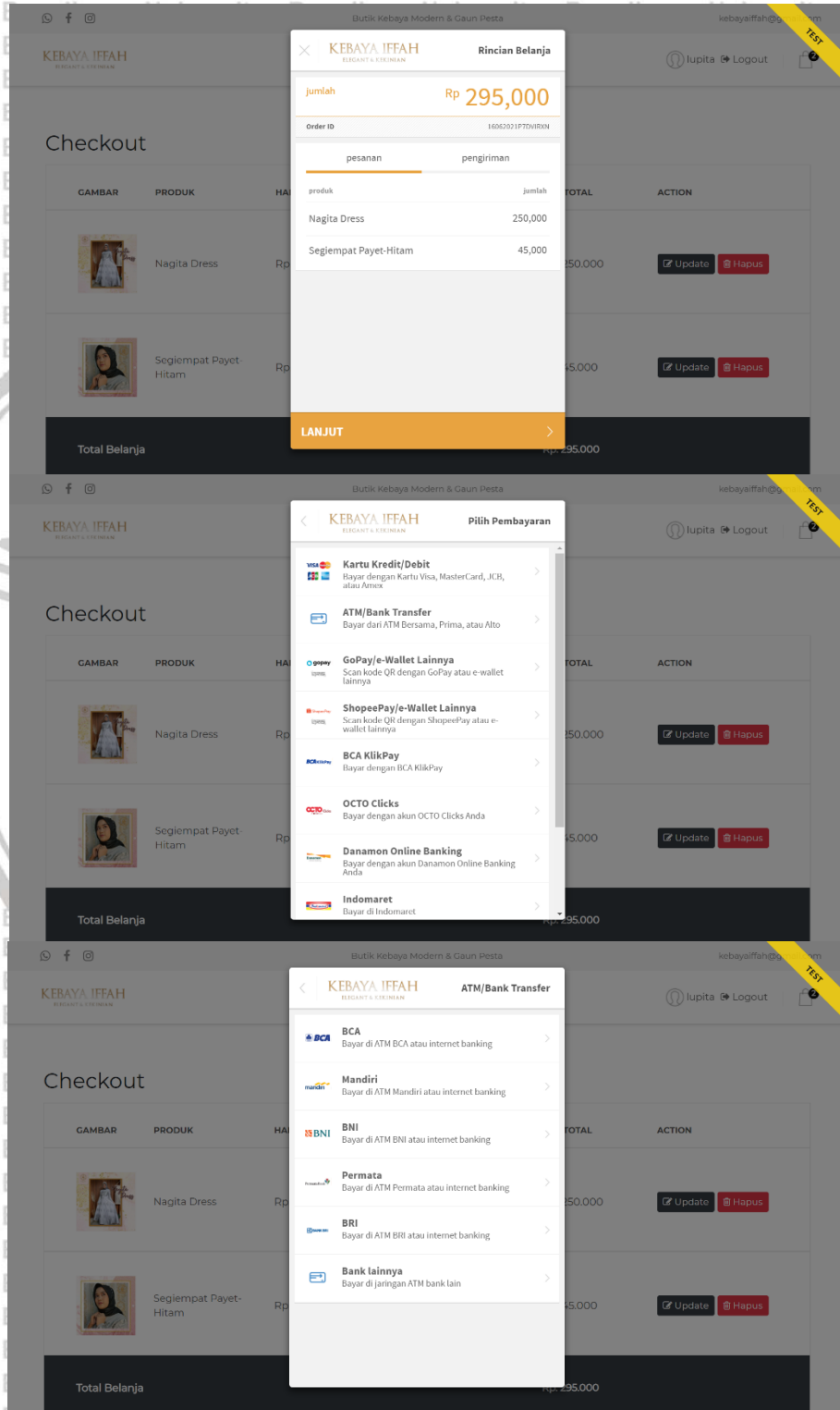
Gambar 5. 9 Implementasi User Interface Halaman Checkout

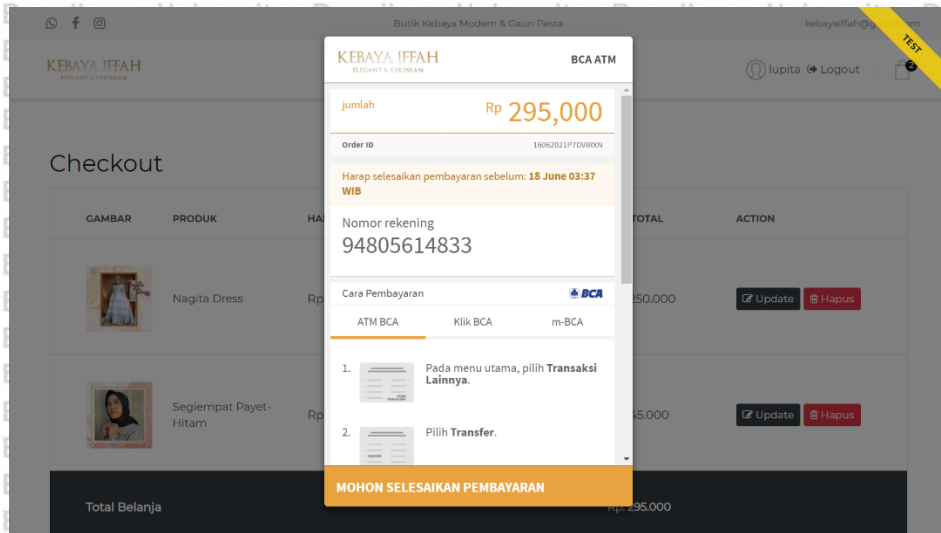
Sumber : (Penulis,2021)



5.3.8 User Interface Pembayaran

Gambar 5.8 menunjukkan implementasi dari *user interface* halaman pembayaran yang mengacu pada perancangan *user interface* halaman pembayaran, dimana bahwa halaman ini merupakan halaman snap yang terintegrasi dengan payment gateway.

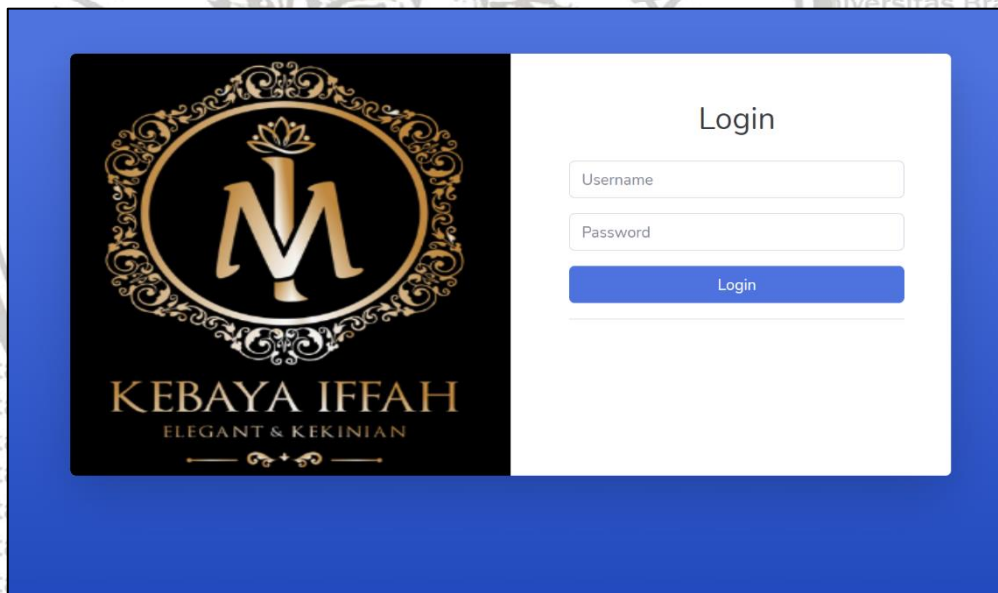




Gambar 5. 10 Implementasi *User Interface* Halaman Pembayaran Midtrans

5.3.9 *User Interface* Login Admin

Gambar 5.9 menunjukkan implementasi dari *user interface* halaman Login Admin yang mengacu pada perancangan *user interface* halaman Login Admin. Terdapat form input username dan password admin.

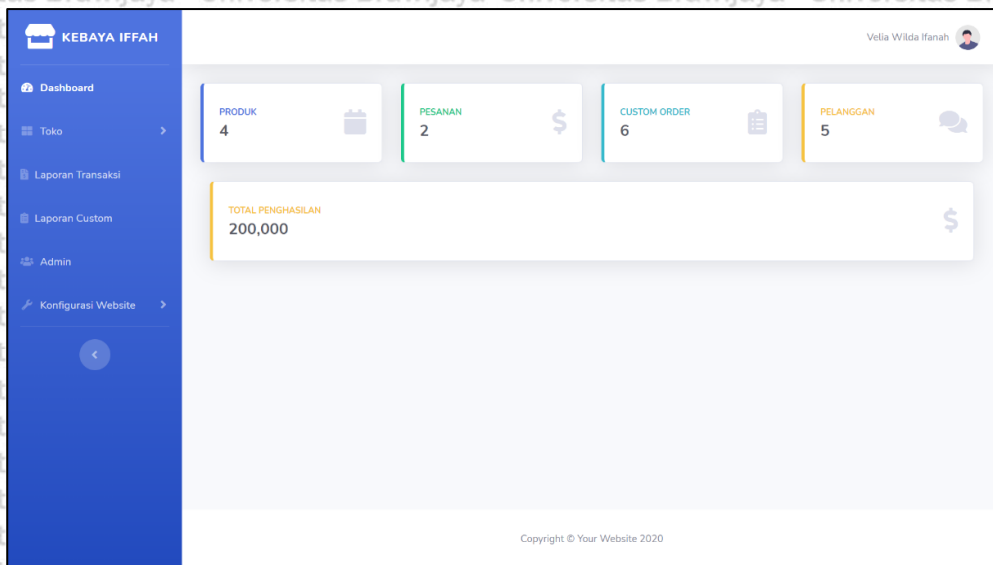


Gambar 5. 11 Implementasi *User Interface* Halaman Login Admin

Sumber : (Penulis,2021)

5.3.10 *User Interface* Halaman Dashboard Admin

Gambar 5.10 menunjukkan implementasi dari *user interface* halaman dashboard admin dimana halaman ini berisikan informasi ringkasan sistem dan total penghasilan. Implementasi *user interface* ini mengacu pada perancangan *user interface* halaman dashboard admin.

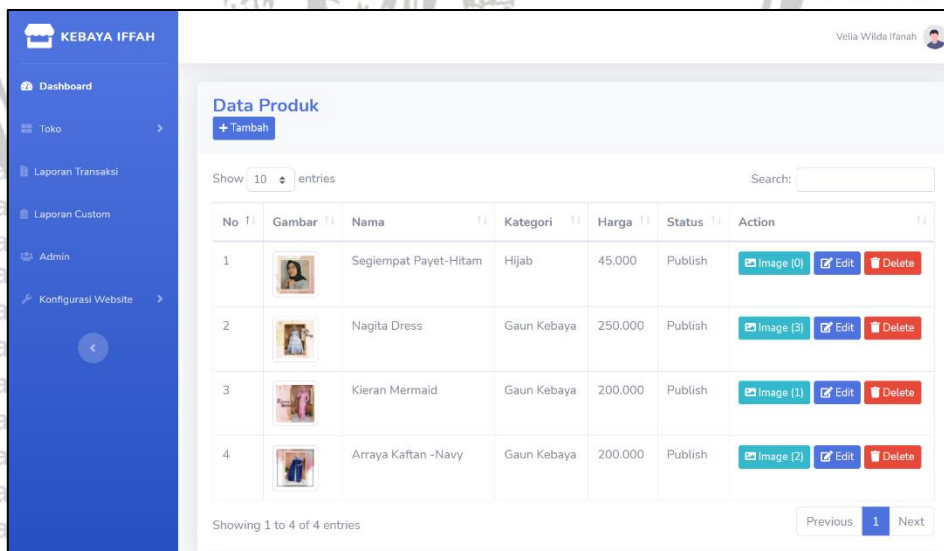


Gambar 5. 12 Implementasi *User Interface* Halaman *Dashboard Admin*

Sumber : (Penulis,2021)

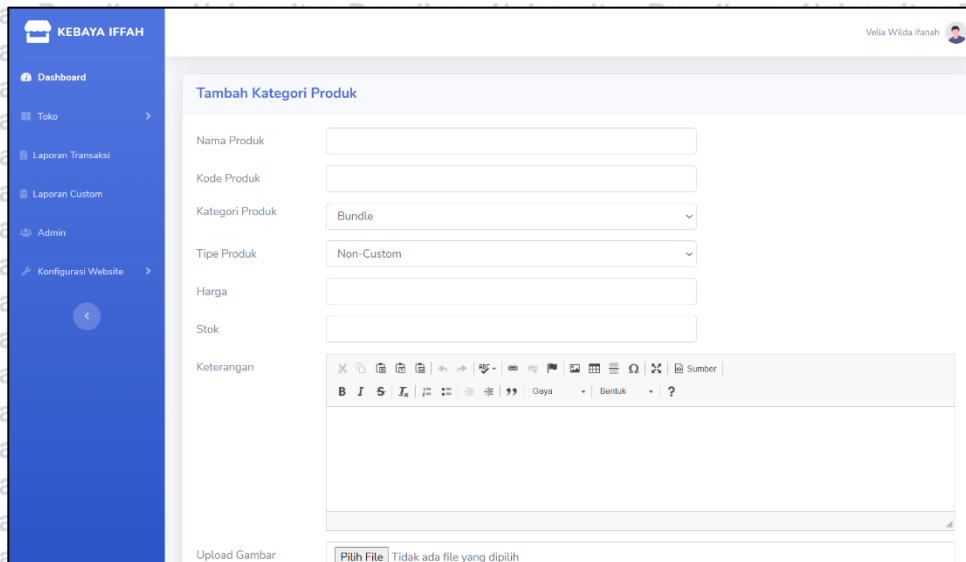
5.3.11 *User Interface* Halaman Mengelola Produk

Pada Gambar 5.11 menunjukkan halaman daftar produk dengan terdapat tombol untuk tambah, edit dan hapus serta tombol image untuk mengelola gambar. Kemudian pada Gambar 5.12 menunjukkan halaman form tambah produk yang akan ditampilkan apabila aktor menekan tombol 'Tambah'.



Gambar 5. 13 Implementasi *User Interface* Halaman Mengelola Produk

Sumber : (Penulis,2021)

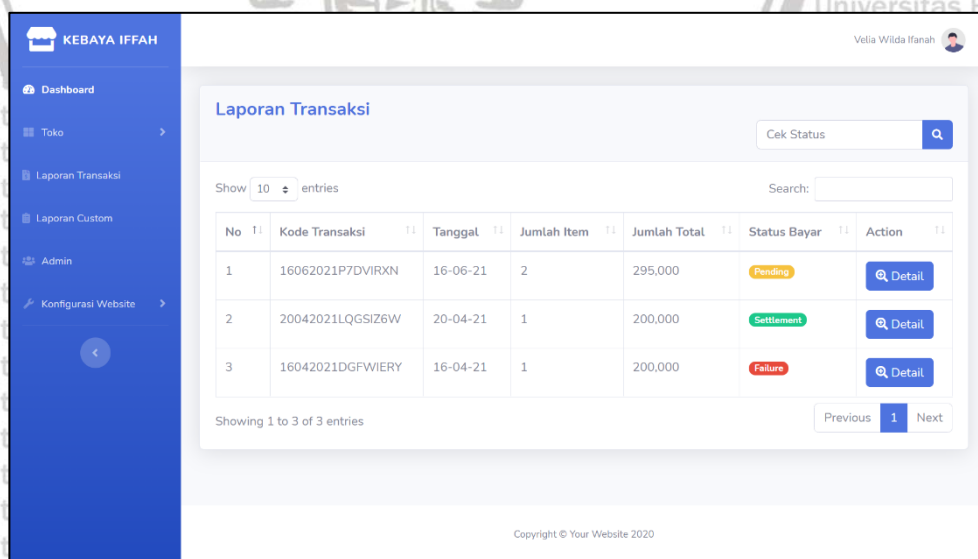


Gambar 5. 14 Implementasi User Interface Halaman Mengelola Produk (Form Tambah)

Sumber : (Penulis,2021)

5.3.12 User Interface Halaman Melihat Data Pesanan dan Update Status

Gambar 5.13 menunjukkan implementasi dari *user interface* halaman melihat data pesanan dimana halaman ini menampilkan daftar pesanan dan pada bagian atasnya terdapat form input untuk melakukan update status bayar, dan terdapat tiga warna indikator yaitu hijau untuk pembayaran settlement, kuning untuk pending dan merah untuk failure. Implementasi *user interface* ini mengacu pada perancangan *user interface* halaman melihat data transaksi.

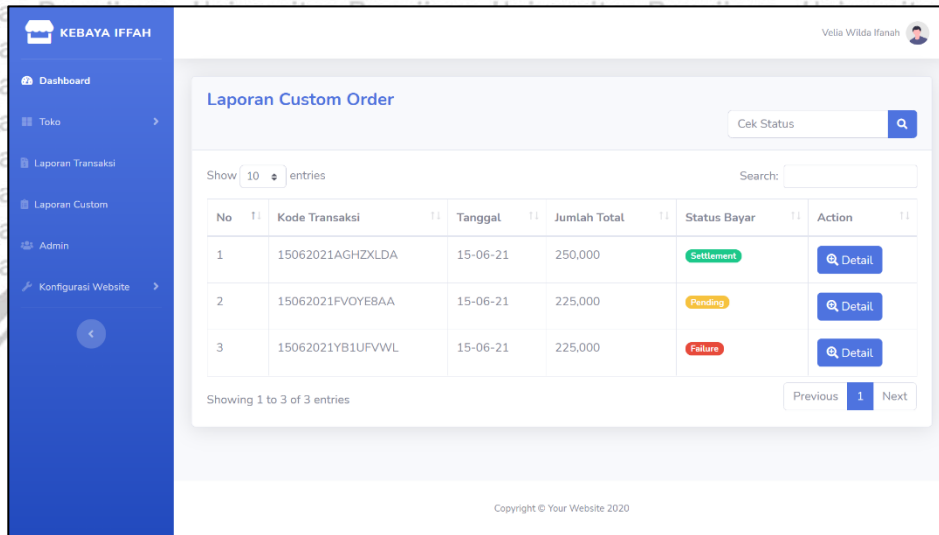


Gambar 5. 15 Implementasi User Interface Halaman Melihat Data Pesanan

Sumber : (Penulis,2021)

5.3.13 User Interface Halaman Melihat Data Custom Order dan Update Status

Gambar 5.14 menunjukkan implementasi dari *user interface* halaman melihat data custom order dimana halaman ini menampilkan daftar custom order dan pada bagian atasnya terdapat form input untuk melakukan update status bayar, dan terdapat tiga warna indikator yaitu hijau untuk pembayaran settlement, kuning untuk pending dan merah untuk failure. Implementasi *user interface* ini mengacu pada perancangan *user interface* halaman melihat data custom order.

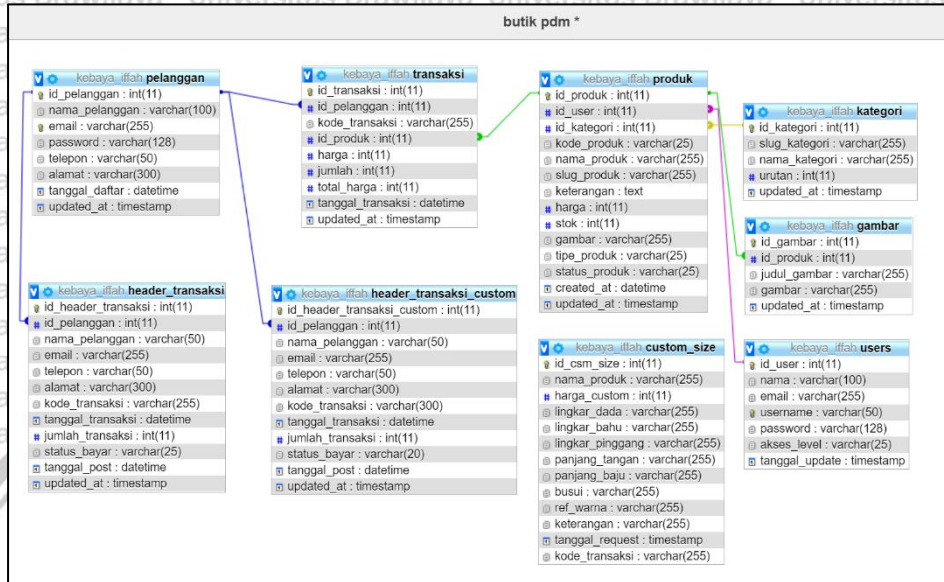


Gambar 5. 16 Implementasi User Interface Halaman Melihat Data Custom Order

Sumber : (Penulis,2021)

5.4 Implementasi Database

Database Management System yang digunakan pada pengembangan website E-commerce Butik Kebaya Iffah adalah MySQL dengan menggunakan tools PHPMYADMIN agar memudahkan proses pengembangan database. Gambar implementasi database akan ditampilkan pada Gambar 5.15



Gambar 5. 17 Implementasi Physical Data Model Database Website E-commerce Butik Kebaya Iffah

Sumber : (Penulis,2021)

Database pada gambar ini adalah database yang dibuat untuk pengembangan website E-commerce Butik Kebaya Iffah dengan atribut atribut sesuai hasil permintaan dan wawancara dengan pemilik bisnis.

5.4.1 Implementasi Table Pelanggan

Tabel Pelanggan dibuat berdasarkan rancangan yang telah dilakukan sebelumnya. Berikut adalah hasil dari implementasi tabel pelanggan dijelaskan pada Tabel 5.3. Tabel ini berfungsi untuk menyimpan data pelanggan.

Tabel 5. 3 Implementasi Tabel Pelanggan

pelanggan.sql	
1	CREATE TABLE `pelanggan` (
2	`id_pelanggan` int(11) NOT NULL,
3	`id_user` int(11) NOT NULL,
4	`nama_pelanggan` varchar(100) NOT
5	NULL,
6	`email` varchar(255) NOT NULL,
7	`password` varchar(128) NOT NULL,
8	`telepon` varchar(50) NOT NULL,
9	`alamat` varchar(300) NOT NULL,
10	`tanggal_daftar` datetime NOT NULL,
11	`updated_at` timestamp NOT NULL


```

12  DEFAULT current_timestamp() ON UPDATE
13  current_timestamp() }

```

5.4.2 Implementasi Table Admin

Tabel users dibuat berdasarkan rancangan yang telah dilakukan sebelumnya. Berikut adalah hasil dari implementasi tabel admin dijelaskan pada Tabel 5.4. Tabel ini berfungsi untuk menyimpan data admin.

Tabel 5. 4 Implementasi Tabel Admin

```

users.sql
1  CREATE TABLE `users` (
2  `id_user` int(11) NOT NULL,
3  `nama` varchar(100) NOT NULL,
4  `email` varchar(255) NOT NULL,
5  `username` varchar(50) NOT NULL,
6  `password` varchar(128) NOT NULL,
7  `akses_level` varchar(25) NOT NULL,
8  `tanggal_update` timestamp NOT NULL
9  DEFAULT current_timestamp() ON UPDATE
10 current_timestamp()
11 )

```

5.4.3 Implementasi Tabel Produk

Tabel Produk dibuat berdasarkan rancangan yang telah dilakukan sebelumnya. Berikut adalah hasil dari implementasi tabel produk dijelaskan pada Tabel 5.5. Tabel ini berfungsi untuk menyimpan data produk.

Tabel 5. 5 Implementasi Tabel Produk

```

produk.sql
1  CREATE TABLE `produk` (
2  `id_produk` int(11) NOT NULL,
3  `id_user` int(11) DEFAULT NULL,
4  `id_kategori` int(11) NOT NULL,
5  `kode_produk` varchar(25) NOT NULL,
6  `nama_produk` varchar(255) NOT NULL,
7  `slug_produk` varchar(255) NOT NULL,
8  `keterangan` text NOT NULL,
9  `harga` int(11) NOT NULL,
10 `stok` int(11) NOT NULL,
11 `gambar` varchar(255) NOT NULL,
12 `tipe_produk` varchar(25) NOT NULL,
13 `status_produk` varchar(25) NOT
14 NULL,
15 `created_at` datetime NOT NULL,
16 `updated_at` timestamp NOT NULL
17 DEFAULT current_timestamp() ON UPDATE
18 current_timestamp()
19 )

```

5.4.4 Implementasi Tabel Header Transaksi Pesanan

Tabel header_transaksi dibuat berdasarkan rancangan yang telah dilakukan sebelumnya. Berikut adalah hasil dari implementasi tabel header_transaksi dijelaskan pada Tabel 5.6. Tabel ini berfungsi untuk menyimpan data pesanan produk ready pada sistem.

Tabel 5. 6 Implementasi Tabel Header Transaksi

```
Header_transaksi.sql
1 CREATE TABLE `header_transaksi` (
2   `id_header_transaksi` int(11) NOT
3   NULL,
4   `id_user` int(11) NOT NULL,
5   `id_pelanggan` int(11) NOT NULL,
6   `nama_pelanggan` varchar(50) DEFAULT
7   NULL,
8   `email` varchar(255) DEFAULT NULL,
9   `telepon` varchar(50) DEFAULT NULL,
10  `alamat` varchar(300) DEFAULT NULL,
11  `kode_transaksi` varchar(255) NOT
12  NULL,
13  `tanggal_transaksi` datetime NOT
14  NULL,
15  `jumlah_transaksi` int(11) NOT NULL,
16  `status_bayar` varchar(25) NOT NULL,
17  `tanggal_post` datetime NOT NULL,
18  `updated_at` timestamp NOT NULL
19  DEFAULT current_timestamp() ON UPDATE
20  current_timestamp()
21  )
```

5.4.5 Implementasi Tabel Transaksi Pesanan

Tabel transaksi dibuat berdasarkan rancangan yang telah dilakukan sebelumnya. Berikut adalah hasil dari implementasi tabel transaksi dijelaskan pada Tabel 5.7. Tabel ini berfungsi untuk menyimpan data pesanan produk ready pada sistem dan detail dari header_transaksi.

Tabel 5. 7 Implmentasi Tabel Transaksi

```

transaksi.sql
1 CREATE TABLE `transaksi` (
2   `id_transaksi` int(11) NOT NULL,
3   `id_user` int(11) NOT NULL,
4   `id_pelanggan` int(11) NOT NULL,
5   `kode_transaksi` varchar(255)
6   NOT NULL,
7   `id_produk` int(11) NOT NULL,
8   `harga` int(11) NOT NULL,
9   `jumlah` int(11) NOT NULL,
10  `total_harga` int(11) NOT NULL,
11  `tanggal_transaksi` datetime NOT
12  NULL,
13  `updated_at` timestamp NOT NULL
14  DEFAULT current_timestamp() ON
15  UPDATE current_timestamp()
16  )
17
18

```

5.4.6 Implementasi Tabel Header Transaksi Custom

Tabel header_transaksi_custom dibuat berdasarkan rancangan yang telah dilakukan sebelumnya. Berikut adalah hasil dari implementasi tabel header_transaksi_custom dijelaskan pada Tabel 5.8. Tabel ini berfungsi untuk menyimpan data pesanan custom order pada sistem.

Tabel 5. 8 Implementasi Tabel Header Transaksi Custom

```

Header_transaksi_custom.sql
1 CREATE TABLE
2   `header_transaksi_custom` (
3     `id_header_transaksi_custom` int(11)
4     NOT NULL,
5     `id_user` int(11) NOT NULL,
6     `id_pelanggan` int(11) NOT NULL,
7     `nama_pelanggan` varchar(50)
8     CHARACTER SET utf8mb4 NOT NULL,
9     `email` varchar(255) CHARACTER SET
10    utf8mb4 NOT NULL,
11    `telepon` varchar(50) CHARACTER SET
12    utf8mb4 NOT NULL,
13    `alamat` varchar(300) CHARACTER SET
14    utf8mb4 NOT NULL,
15    `kode_transaksi` varchar(300)
16    CHARACTER SET utf8mb4 NOT NULL,
17    `tanggal_transaksi` datetime NOT
18    NULL,
19    `jumlah_transaksi` int(11) NOT NULL,
20    `status_bayar` varchar(20) CHARACTER
21    SET utf8mb4 NOT NULL,
22    `tanggal_post` datetime NOT NULL,
23    `updated_at` timestamp NOT NULL

```

```

23 DEFAULT current_timestamp() ON UPDATE
24 current_timestamp()
25 )

```

5.4.7 Implementasi Tabel Custom

Tabel custom dibuat berdasarkan rancangan yang telah dilakukan sebelumnya. Berikut adalah hasil dari implementasi tabel custom dijelaskan pada Tabel 5.9. Tabel ini berfungsi untuk menyimpan data pesanan custom order pada sistem dan detail atribut custom.

Tabel 5. 9 Implementasi Tabel Custom

```

Custom.sql
1 CREATE TABLE `custom_size` (
2   `id_csm_size` int(11) NOT NULL,
3   `nama_produk` varchar(255) NOT NULL,
4   `harga_custom` int(11) NOT NULL,
5   `lingkar_dada` varchar(255) NOT NULL,
6   `lingkar_bahu` varchar(255) NOT NULL,
7   `lingkar_pinggang` varchar(255) NOT NULL,
8   `panjang_tangan` varchar(255) NOT NULL,
9   `panjang_baju` varchar(255) NOT NULL,
10  `busui` varchar(255) NOT NULL,
11  `ref_warna` varchar(255) NOT NULL,
12  `keterangan` varchar(255) NOT NULL,
13  `tanggal_request` timestamp NOT NULL DEFAULT current_timestamp(),
14  `kode_transaksi` varchar(255) CHARACTER SET utf8mb4 NOT NULL
15 )

```

5.4.8 Implementasi Tabel Kategori

Tabel kategori dibuat berdasarkan rancangan yang telah dilakukan sebelumnya. Berikut adalah hasil dari implementasi tabel kategori dijelaskan pada Tabel 5.10. Tabel ini berfungsi untuk menyimpan data kategori.

Tabel 5. 10 Implementasi Tabel Kategori

```

kategori.sql
1 CREATE TABLE `kategori` (
2   `id_kategori` int(11) NOT NULL,
3   `slug_kategori` varchar(255) NOT NULL,
4   `nama_kategori` varchar(255) NOT NULL,
5   `urutan` int(11) DEFAULT NULL,
6   `updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON
7   UPDATE current_timestamp()
8 )

```


5.4.9 Implementasi Tabel Gambar

Tabel gambar dibuat berdasarkan rancangan yang telah dilakukan sebelumnya. Berikut adalah hasil dari implementasi tabel gambar dijelaskan pada Tabel 5.11. Tabel ini berfungsi untuk menyimpan data gambar.

Tabel 5. 11 Implementasi Tabel Gambar

```
gambar.sql
1 CREATE TABLE `gambar` (
2   `id_gambar` int(11) NOT NULL,
3   `id_produk` int(11) NOT NULL,
4   `judul_gambar` varchar(255) NOT NULL,
5   `gambar` varchar(255) NOT NULL,
6   `updated_at` timestamp NOT NULL DEFAULT
7   current_timestamp() ON UPDATE
8   current_timestamp()
9 )
```

5.5 Implementasi Kode Program

Implementasi kode program merupakan tahap penulisan kode program dengan bahasa pemrograman berdasarkan *class diagram* yang telah dirancang. Bahasa pemrograman yang digunakan adalah *PHP*. Pada bagian ini, pembahasan kode program meliputi fungsi-fungsi yang digunakan oleh actor admin dan pelanggan.

5.5.1 Kode Program Pengaturan Umum Midtrans

Terdapat pengaturan umum pada library midtrans, yang dapat dilihat pada Tabel 5.12. Pada baris pertama berfungsi untuk mengatur *server key merchant*. Dan baris kedua digunakan untuk mengatur lingkungan pengembangan apakah menggunakan lingkungan *production* atau lingkungan *sandbox*. Dikarenakan pada implementasi menggunakan lingkungan *sandbox* Midtrans maka konfigurasi *\$isProduction* tersebut diatur dengan nilai *false*.

Tabel 5. 12 Kode Program Pengaturan Umum Midtrans

```
General Setting Midtrans
1 \Midtrans\Config::$serverKey = $params['server_key'];
2 \Midtrans\Config::$isProduction = false;
```

5.5.2 Kode Program Produk

Hasil implementasi kode program untuk proses melihat data produk oleh aktor pelanggan, ditunjukkan pada tabel 5.12. Pada Controller dengan nama Produk ini terdapat method index yang berfungsi dalam menampilkan halaman produk pada Website *E-commerce* Butik Kebaya Iffah. Dan juga terdapat fungsi detail untuk menampilkan halaman detail produk.

Selanjutnya terdapat Controller dengan nama Produk pada sub folder admin dapat dilihat pada tabel 5.13 yang digunakan untuk menampilkan, menambah melakukan edit dan menghapus data admin produk pada halaman admin. Seluruh data mengenai produk akan dikelola atau diambil dari produk_model yang dapat dilihat pada tabel 5.14.

Tabel 5. 13 Kode Program Controller Produk

```

Produk.php
1 //load model
2 public function __construct()
3 {
4     parent::__construct();
5     $this->load->model('Produk_model');
6     $this->load->model('Kategori_model');
7 }
8
9
10 //list data produk
11 public function index()
12 {
13     $site = $this->konfigurasi_model-
14 >get_all_konfigurasi();
15     $listing_kategori = $this->Produk_model-
16 >listing_kategori();
17     //ambil data total
18     $total = $this->Produk_model->total_produk();
19     //paginasi
20     $this->load->library('pagination');
21
22     $config['base_url'] =
23     base_url().'produk/index/';
24     $config['total_rows'] = $total->total;
25     $config['use_page_numbers'] = TRUE;
26     $config['per_page'] = 3;
27     $config['uri_segment'] = 3;
28     $config['num_links'] = 5;
29     $config['full_tag_open'] = '<nav><ul
30 class="pagination">';
31     $config['full_tag_close'] = '</ul></nav>';
32     $config['first_link'] = 'First';
33     $config['first_tag_open'] = '<li class="page-item">';
34     $config['first_tag_close'] = '</li>';
35     $config['last_link'] = 'Last';
36     $config['last_tag_open'] = '<li class="page-item">';
37     $config['last_tag_close'] = '</li>';
38     $config['next_link'] = '&raquo;';
39     $config['next_tag_open'] = '<li class="page-item">';
40     $config['next_tag_close'] = '</li>';
41     $config['prev_link'] = '&laquo;';
42     $config['prev_tag_open'] = '<li class="page-item">';
43     $config['prev_tag_close'] = '</li>';
44     $config['cur_tag_open'] = '<li class="page-item
45 active"><a class="page-link" href="#">';
46     $config['cur_tag_close'] = '</a></li>';

```



```

47     $config['num_tag_open'] = '<li class="page-item">';
48     $config['num_tag_close'] = '</li>';
49     $config['attributes'] = array('class' => 'page-
50 link');
51     $config['first_url'] = base_url().'/produk/';
52
53     //styling
54
55
56 //initialize
57 $this->pagination->initialize($config);
58
59 //ambil data produk
60 $page = ($this->uri->segment(3)) ? ($this->uri-
61 >segment(3)-1) * $config['per_page']:0;
62 $produk = $this->Produk_model-
63 >produk($config['per_page'], $page);
64
65
66     $data = array('title' => 'Produk
67 '.$site->namaweb,
68 'site' => $site,
69 'listing_kategori' =>
70 $listing_kategori,
71 'produk' => $produk,
72 'pagin' => $this-
73 >pagination->create_links(),
74 'isi' =>
75 'produk/list' );
76     $this->load->view('layout/wrapper', $data, FALSE);
77 }
78
79 //listing kategori
80 public function kategori($slug_kategori)
81 {
82     $kategori = $this->Kategori_model-
83 >read($slug_kategori);
84     $id_kategori = $kategori->id_kategori;
85
86     $site = $this->konfigurasi_model-
87 >get_all_konfigurasi();
88     $listing_kategori = $this->Produk_model-
89 >listing_kategori();
90     //ambil data total
91     $total = $this->Produk_model-
92 >total_kategori($id_kategori);
93 //paginasi
94 $this->load->library('pagination');
95
96     $config['base_url'] =
97 base_url().'/produk/kategori/'.$slug_kategori.'/index/';
98     $config['total_rows'] = $total->total;
99     $config['use_page_numbers'] = TRUE;
100    $config['per_page'] = 6;
101    $config['uri_segment'] = 5;
102    $config['num_links'] = 5;
103    $config['full_tag_open'] = '<ul class="pagination">';

```

```

104 $config['full_tag_close'] = '</ul>';
105 $config['first_link'] = 'First';
106 $config['first_tag_open'] = '<li>';
107 $config['first_tag_close'] = '</li>';
108 $config['last_link'] = 'Last';
109 $config['last_tag_open'] = '<li class="disabled"><li
110 class="active"><a href="a">';
111 $config['last_tag_close'] = '<span class="sr-
112 only"></a></li></li>';
113 $config['next_link'] = '&gt;';
114 $config['next_tag_open'] = '<div>';
115 $config['next_tag_close'] = '</div>';
116 $config['prev_link'] = '&lt;';
117 $config['prev_tag_open'] = '<div>';
118 $config['prev_tag_close'] = '</div>';
119 $config['cur_tag_open'] = '<b>';
120 $config['cur_tag_close'] = '</b>';
121 $config['first_url']
122 base_url().'produk/kategori/'.$slug_kategori;
123 $this->pagination->initialize($config);
124
125 //ambil data produk
126 $page = ($this->uri->segment(5)) ? ($this->uri-
127 >segment(5)-1) * $config['per_page']:0;
128 $produk = $this->Produk_model->kategori($id_kategori,
129 $config['per_page'], $page);
130
131
132 $data = array('title' =>
133 $kategori->nama_kategori,
134 'site' => $site,
135 'listing_kategori' =>
136 $listing_kategori,
137 'produk' => $produk,
138 'pagin' => $this-
139 >pagination->create_links(),
140 'isi' =>
141 'produk/list' );
142 $this->load->view('layout/wrapper', $data, FALSE);
143 }
144
145
146 //detail produk
147 public function detail($slug_produk)
148 {
149 $site = $this->konfigurasi_model-
150 >get_all_konfigurasi();
151 $produk = $this->Produk_model->read($slug_produk);
152 $id_produk = $produk->id_produk;
153 $gambar = $this->Produk_model->gambar($id_produk);
154 $related_produk = $this->Produk_model->home();
155
156 $data = array('title' => $produk-
157 >nama_produk,
158 'site' => $site,
159 'produk' => $produk,
160 'related_produk' =>

```



```
161     $related_produk,  
162     'gambar' => $gambar,  
163     'isi' =>  
164     'produk/detail');  
165     $this->load->view('layout/wrapper', $data, FALSE);  
166 }
```



Tabel 5. 14 Kode Program Controller Mengelola Produk

```

Admin/ Produk.php
1 //load model
2 public function __construct()
3 {
4     parent::__construct();
5     $this->load->model('Produk_model');
6     $this->load->model('Kategori_model');
7 }
8 public function index()
9 {
10    $produk = $this->Produk_model->get_all_produk();
11    $data = array ('title' => 'Data Produk',
12                 'produk' => $produk,
13                 'isi' => 'admin/produk/list' );
14    $this->load->view('admin/layout/main', $data, FALSE);
15 }
16
17 //gambar
18 public function gambar($id_produk){
19     $produk = $this->Produk_model->detail($id_produk);
20     $gambar = $this->Produk_model->gambar($id_produk);
21
22     // Validation input
23     $this->form_validation->set_rules('judul_gambar','Judul
24 Gambar','required');
25
26     if($this->form_validation->run())
27     {
28         $config['upload_path'] = './assets/upload/image/';
29         $config['allowed_types'] = 'gif|jpg|png|jpeg';
30         $config['max_size'] = '2400'; //kb
31         $config['max_width'] = '2024';
32         $config['max_height'] = '2024';
33
34         $this->load->library('upload', $config);
35
36         if (! $this->upload->do_upload('gambar')){
37
38             $data = array('title' => 'Tambah Gambar Produk:
39 '.$produk->nama_produk,
40                         'produk' => $produk,
41                         'gambar' => $gambar,
42                         'error' => $this->upload-
43 >display_errors(),
44                         'isi' => 'admin/Produk/gambar');
45             $this->load->view('admin/layout/main', $data, FALSE);
46         }else{
47             $upload_gambar = array('upload_data' => $this->upload-
48 >data());
49
50             //thumbnail
51             $config['image_library'] = 'gd2';
52             $config['source_image'] =
53 './assets/upload/image/'.$upload_gambar['upload_data']['file_name

```



```

54     ];
55     //Lokasi folder thumbnail
56     $config['new_image'] =
57     './assets/upload/image/thumbs/';
58     $config['create_thumb'] = TRUE;
59     $config['maintain_ratio'] = TRUE;
60     $config['width'] = 250;
61     $config['height'] = 250;
62     $config['thumb_marker'] = '';
63
64     $this->load->library('image_lib', $config);
65
66     $this->image_lib->resize();
67
68     $data = array(
69         'id_produk' => $id_produk,
70         'judul_gambar' => $this->input->post('judul_gambar'),
71         'gambar' =>
72         $upload_gambar['upload_data']['file_name'],
73     );
74     $this->Produk_model->tambah_gambar($data);
75     $this->session->set_flashdata('sukses', 'Data gambar
76     telah ditambahkan');
77
78     redirect(base_url('admin/produk/gambar/'.$id_produk), 'refresh!');
79     }
80     }
81     $data = array('title' => 'Tambah Gambar Produk:
82     '.$produk->nama_produk,
83         'produk' => $produk,
84         'gambar' => $gambar,
85         'isi' => 'admin/Produk/gambar');
86     $this->load->view('admin/layout/main', $data, FALSE);
87     }
88
89     //Tambah produk
90     public function tambah()
91     {
92         $kategori = $this->Kategori_model->get_all_kategori();
93         // Validation input
94         $this->form_validation->set_rules('nama_produk', 'Nama
95     Produk', 'required');
96         $this->form_validation->set_rules('kode_produk', 'Kode
97     Produk', 'required|is_unique[produk.kode_produk]');
98
99         if($this->form_validation->run())
100     {
101         $config['upload_path'] = './assets/upload/image/';
102         $config['allowed_types'] = 'gif|jpg|png|jpeg';
103         $config['max_size'] = '2400'; //kb
104         $config['max_width'] = '2024';
105         $config['max_height'] = '2024';
106
107         $this->load->library('upload', $config);
108
109         if (!$this->upload->do_upload('gambar')){
110

```

```
111 $data = array('title' => 'Tambah Produk',
112             'kategori' => $kategori,
113             'error' => $this->upload-
114 >display_errors(),
115             'isi' => 'admin/Produk/tambah'
116             );
117 $this->load->view('admin/layout/main', $data, FALSE);
118 }else{
119 $upload_gambar = array('upload_data' => $this->upload-
120 >data());
121 $slug_produk = url_title($this->input-
122 >post('nama_produk').'-'. $this->input->post('kode_produk'),
123 'dash', TRUE);
124 //thumbnail
125 $config['image_library'] = 'gd2';
126 $config['source_image'] =
127 './assets/upload/image/'. $upload_gambar['upload_data']['file_name
128 '];
129 $config['new_image'] =
130 './assets/upload/image/thumbs/';
131 $config['create_thumb'] = TRUE;
132 $config['maintain_ratio'] = TRUE;
133 $config['width'] = 250;
134 $config['height'] = 250;
135 $config['thumb_marker'] = '';
136
137 $this->load->library('image_lib', $config);
138
139 $this->image_lib->resize();
140 $data = array(
141     'id_user' => $this->session->userdata('id_user'),
142     'id_kategori' => $this->input->post('id_kategori'),
143     'kode_produk' => $this->input->post('kode_produk'),
144     'nama_produk' => $this->input->post('nama_produk'),
145     'slug_produk' => $slug_produk,
146     'keterangan' => $this->input->post('keterangan'),
147     'harga' => $this->input->post('harga'),
148     'stok' => $this->input->post('stok'),
149     'gambar' => $upload_gambar['upload_data']['file_name'],
150     'tipe_produk' => $this->input->post('tipe_produk'),
151     'status_produk' => $this->input->post('status_produk'),
152     'created_at' => date('Y-m-d H:i:s')
153 );
154 $this->Produk_model->tambah($data);
155 $this->session->set_flashdata('sukses', 'Data telah
156 ditambahkan!');
157 redirect(base_url('admin/Produk'), 'refresh');
158 }
159 }
160 }
161 }
162 $data = array('title' => 'Tambah Produk',
163             'kategori' => $kategori,
164             'isi' => 'admin/Produk/tambah');
165 $this->load->view('admin/layout/main', $data, FALSE);
166 }
167 }
```



```

168 //Edit produk
169 public function edit($id_produk)
170 {
171     $produk = $this->Produk_model->detail($id_produk);
172     $kategori = $this->Kategori_model->get_all_kategori();
173
174     $this->form_validation->set_rules('nama_produk', 'Nama
175 Produk', 'required');
176     $this->form_validation->set_rules('kode_produk', 'Kode
177 Produk', 'required');
178
179     if($this->form_validation->run())
180     {
181         //cek jika ganti gambar
182         if(!empty($_FILES['gambar']['name'])) {
183
184             $config['upload_path'] = './assets/upload/image/';
185             $config['allowed_types'] = 'gif|jpg|png|jpeg';
186             $config['max_size'] = '2400'; //kb
187             $config['max_width'] = '2024';
188             $config['max_height'] = '2024';
189
190             $this->load->library('upload', $config);
191
192             if (!$this->upload->do_upload('gambar')) {
193
194                 $data = array('title' => 'Edit Produk: '.$produk->
195 >nama_produk,
196                             'kategori' => $kategori,
197                             'produk' => $produk,
198                             'error' => $this->upload->
199 >display_errors(),
200                             'isi' => 'admin/Produk/edit');
201                 $this->load->view('admin/layout/main', $data, FALSE);
202             }else{
203                 $upload_gambar = array('upload_data' => $this->upload->
204 >data());
205                 //thumbnail
206                 $config['image_library'] = 'gd2';
207                 $config['source_image']
208                 './assets/upload/image/'.$upload_gambar['upload_data']['file_name
209                 '];
210                 $config['new_image']
211                 './assets/upload/image/thumbs/';
212                 $config['create_thumb'] = TRUE;
213                 $config['maintain_ratio'] = TRUE;
214                 $config['width'] = 250;
215                 $config['height'] = 250;
216                 $config['thumb_marker'] = '';
217
218                 $this->load->library('image lib', $config);
219                 $this->image_lib->resize();
220                 $slug_produk = url_title($this->input->
221 >post('nama_produk').'-'.$this->input->post('kode_produk'),
222 >dash', TRUE);
223                 $data = array(
224                 'id_produk' => $id_produk,

```



```

225 'id_user' => $this->session->userdata('id_user'),
226 'id_kategori' => $this->input->post('id_kategori'),
227 'kode_produk' => $this->input->post('kode_produk'),
228 'nama_produk' => $this->input->post('nama_produk'),
229 'slug_produk' => $slug_produk,
230 'keterangan' => $this->input->post('keterangan'),
230 'harga' => $this->input->post('harga'),
231 'stok' => $this->input->post('stok'),
232 'gambar' => $upload_gambar['upload_data']['file_name'],
233 $upload_gambar['upload_data']['file_name'],
234 'tipe_produk' => $this->input->post('tipe_produk'),
235 'status_produk' => $this->input->post('status_produk')
236 );
237 $this->Produk_model->edit($data);
238 $this->session->set_flashdata('sukses', 'Data telah
239 diedit');
240 redirect(base_url('admin/Produk'),'refresh');
241 }}else{
242 //Edit produk tanpa ganti gambar
243 $slug_produk = url_title($this->input-
244 >post('nama_produk').'-'. $this->input->post('kode_produk'),
245 'dash', TRUE);
246 $data = array(
247 'id_produk' => $id_produk,
248 'id_user' => $this->session->userdata('id_user'),
249 'id_kategori' => $this->input->post('id_kategori'),
250 'kode_produk' => $this->input->post('kode_produk'),
251 'nama_produk' => $this->input->post('nama_produk'),
252 'slug_produk' => $slug_produk,
253 'keterangan' => $this->input->post('keterangan'),
254 'harga' => $this->input->post('harga'),
255 'stok' => $this->input->post('stok'),
256 //'gambar' => $upload_gambar['upload_data']['file_name'],
257 $upload_gambar['upload_data']['file_name'],
258 'tipe_produk' => $this->input->post('tipe_produk'),
259 'status_produk' => $this->input->post('status_produk')
260 );
261 $this->Produk_model->edit($data);
262 $this->session->set_flashdata('sukses', 'Data telah
263 diedit');
264 redirect(base_url('admin/Produk'),'refresh');
265 }
266 }
267 $data = array('title' => 'Edit Produk: '.$produk-
268 >nama_produk,
269 'kategori' => $kategori,
270 'produk' => $produk,
271 'isi' => 'admin/Produk/edit');
272 $this->load->view('admin/layout/main', $data, FALSE);
273 }
274 }
275 // delete produk
276 public function delete($id_produk)
277 {
278 //hapus gambar
279 $produk = $this->Produk_model->detail($id_produk);
280 unlink('./assets/upload/image/'.$produk->gambar);

```



```

281 unlink('./assets/upload/image/thumbs/'.$produk->gambar);
282 $data = array('id_produk'=> $id_produk);
283 $this->Produk_model->delete($data);
284 $this->session->set_flashdata('sukses', 'Data telah
285 dihapus');
286 redirect(base_url('admin/produk'), 'refresh');
287 }
288
289 // delete gambar produk
290 public function delete_gambar($id_produk,$id_gambar)
291 {
292 // hapus gambar
293 $gambar = $this->Produk_model->detail_gambar($id_gambar);
294 unlink('./assets/upload/image/'.$gambar->gambar);
295 unlink('./assets/upload/image/thumbs/'.$gambar->gambar);
296 $data = array('id_gambar'=> $id_gambar);
297 $this->Produk_model->delete_gambar($data);
298 $this->session->set_flashdata('sukses', 'Data gambar telah
299 dihapus');
300 redirect(base_url('admin/produk/gambar/'.$id_produk),
301 'refresh');
302 }

```

Tabel 5. 15 Kode Program Model Produk

```

Produk_model.php
1 public function __construct()
2 {
3     parent::__construct();
4     $this->load->database();
5 }
6
7 public function get_all_produk()
8 {
9     $this->db->select('produk.*,
10                     users.nama,
11                     kategori.nama_kategori,
12                     kategori.slug_kategori,
13                     COUNT(gambar.id_gambar) AS
14 total_gambar');
15     $this->db->from('produk');
16     //join
17     $this->db->join('users','users.id_user
18     produk.id_user','left');
19     $this->db->join('kategori','kategori.id_kategori
20     produk.id_kategori','left');
21     $this->db->join('gambar','gambar.id_produk
22     produk.id_produk','left');
23     // end
24     $this->db->group_by('produk.id_produk');
25     $this->db->order_by('id_produk', 'desc');
26     $query = $this->db->get();
27     return $query->result();
28 }
29
30 //HOME

```

```

31 public function home ()
32 {
33     $this->db->select('produk.*,
34         users.nama,
35         kategori.nama_kategori,
36         kategori.slug_kategori,
37         COUNT(gambar.id_gambar) AS
38     total_gambar');
39     $this->db->from('produk');
40     //join
41     $this->db->join('users','users.id_user =
42     produk.id_user','left');
43     $this->db->join('kategori','kategori.id_kategori =
44     produk.id_kategori','left');
45     $this->db->join('gambar','gambar.id_produk =
46     produk.id_produk','left');
47     //end
48     $this->db->where('produk.status_produk','Publish');
49     $this->db->group_by('produk.id_produk');
50     $this->db->order_by('id_produk','desc');
51     $this->db->limit(12);
52     $query = $this->db->get();
53     return $query->result();
54 }
55
56 // REad produk
57 public function read($slug_produk)
58 {
59     $this->db->select('produk.*,
60         users.nama,
61         kategori.nama_kategori,
62         kategori.slug_kategori,
63         COUNT(gambar.id_gambar) AS
64     total_gambar');
65     $this->db->from('produk');
66     //join
67     $this->db->join('users','users.id_user =
68     produk.id_user','left');
69     $this->db->join('kategori','kategori.id_kategori =
70     produk.id_kategori','left');
71     $this->db->join('gambar','gambar.id_produk =
72     produk.id_produk','left');
73     // end
74     $this->db->where('produk.status_produk','Publish');
75     $this->db->where('produk.slug_produk',$slug_produk);
76     $this->db->group_by('produk.id_produk');
77     $this->db->order_by('id_produk','desc');
78     $query = $this->db->get();
79     return $query->row();
80 }
81
82 //PRODUK
83 public function produk($limit,$start)
84 {
85     $this->db->select('produk.*,
86         users.nama,
87         kategori.nama_kategori,

```



```

88     kategori.slug_kategori,
89     COUNT(gambar.id_gambar) AS
90     total_gambar');
91     $this->db->from('produk');
92     //join
93     $this->db->join('users','users.id_user
94     produk.id_user','left');
95     $this->db->join('kategori','kategori.id_kategori
96     produk.id_kategori','left');
97     $this->db->join('gambar','gambar.id_produk
98     produk.id_produk','left');
99     // end
100    $this->db->where('produk.status_produk','Publish');
101    $this->db->group_by('produk.id_produk');
102    $this->db->order_by('id_produk','desc');
103    $this->db->limit($limit,$start);
104    $query = $this->db->get();
105    return $query->result();
106 }
107
108 //total produk
109 public function total_produk()
110 {
111     $this->db->select('COUNT(*) AS total');
112     $this->db->from('produk');
113     $this->db->where('status_produk','Publish');
114     $query = $this->db->get();
115     return $query->row();
116 }
117
118 //kategori
119 public function kategori($id_kategori,$limit,$start)
120 {
121     $this->db->select('produk.*,
122     users.nama,
123     kategori.nama_kategori,
124     kategori.slug_kategori,
125     COUNT(gambar.id_gambar) AS
126     total_gambar');
127     $this->db->from('produk');
128     //join
129     $this->db->join('users','users.id_user
130     produk.id_user','left');
131     $this->db->join('kategori','kategori.id_kategori
132     produk.id_kategori','left');
133     $this->db->join('gambar','gambar.id_produk
134     produk.id_produk','left');
135     // end
136     $this->db->where('produk.status_produk','Publish');
137     $this->db->where('produk.id_kategori', $id_kategori);
138     $this->db->group_by('produk.id_produk');
139     $this->db->order_by('id_produk','desc');
140     $this->db->limit($limit,$start);
141     $query = $this->db->get();
142     return $query->result();
143 }
144

```

```

145 //total produk
146 public function total_kategori($id_kategori)
147 {
148     $this->db->select('COUNT(*) AS total');
149     $this->db->from('produk');
150     $this->db->where('status_produk','Publish');
151     $this->db->where('id_kategori',$id_kategori);
152     $query = $this->db->get();
153     return $query->row();
154 }
155
156 //list kategori
157 public function listing_kategori()
158 {
159     $this->db->select('produk.*,
160                    users.nama,
161                    kategori.nama_kategori,
162                    kategori.slug_kategori,
163                    COUNT(gambar.id_gambar) AS
164                    total_gambar');
165     $this->db->from('produk');
166     //join
167     $this->db->join('users','users.id_user
168     produk.id_user','left');
169     $this->db->join('kategori','kategori.id_kategori =
170     produk.id_kategori','left');
171     $this->db->join('gambar','gambar.id_produk
172     produk.id_produk','left');
173     // end
174     $this->db->group_by('produk.id_kategori');
175     $this->db->order_by('id_produk', 'desc');
176     $query = $this->db->get();
177     return $query->result();
178 }
179
180 public function detail($id_produk)
181 {
182     $this->db->select('*');
183     $this->db->from('produk');
184     $this->db->where('id_produk',$id_produk);
185     $this->db->order_by('id_produk', 'desc');
186     $query = $this->db->get();
187     return $query->row();
188 }
189
190 // detail gambar produk
191 public function detail_gambar($id_gambar)
192 {
193     $this->db->select('*');
194     $this->db->from('gambar');
195     $this->db->where('id_gambar',$id_gambar);
196     $this->db->order_by('id_gambar', 'desc');
197     $query = $this->db->get();
198     return $query->row();
199 }
200
201 //gambar

```



```

202 public function gambar($id_produk){
203     $this->db->select('*');
204     $this->db->from('gambar');
205     $this->db->where('id_produk',$id_produk);
206     $this->db->order_by('id_gambar', 'desc');
207     $query = $this->db->get();
208     return $query->result();
209 }
210
211 // Tambah
212 public function tambah($data)
213 {
214     $this->db->insert('produk', $data);
215 }
216
217 // Tambah gambar
218 public function tambah_gambar($data)
219 {
220     $this->db->insert('gambar', $data);
221 }
222
223 //edit
224 public function edit($data)
225 {
226     $this->db->where('id_produk',$data['id_produk']);
227     $this->db->update('produk',$data);
228 }
229
230 // delete
230 public function delete($data)
231 {
232     $this->db->where('id_produk',$data['id_produk']);
233     $this->db->delete('produk',$data);
234 }
235
236 // delete gambar
237 public function delete_gambar($data)
238 {
239     $this->db->where('id_gambar',$data['id_gambar']);
240     $this->db->delete('gambar',$data);
241 }

```

5.5.3 Kode Program Custom

Hasil implementasi kode program untuk proses custom order oleh aktor pelanggan, ditunjukkan pada tabel 5.15. Pada Controller dengan nama Custom ini terdapat method index yang berfungsi dalam menampilkan daftar produk custom pada Website *E-commerce* Butik Kebaya Iffah. Dan juga terdapat fungsi detail untuk menampilkan halaman form custom order. Pada Controller ini juga terdapat fungsi checkout yang dilakukan untuk produk custom beserta penentuan harga custom.

Selanjutnya terdapat Controller dengan nama Custom pada sub folder admin dapat dilihat pada tabel 5.16 yang digunakan untuk menampilkan data custom

order pada halaman admin. selain itu pada controller ini berisikan juga perintah untuk melakukan update status pembayaran melalui method cek_status. Seluruh data mengenai produk custom akan dikelola atau diambil dari custom_model yang dapat dilihat pada tabel 5.17. Sedangkan untuk data transaksi penjualan custom akan dikelola oleh header_transaksi_custom model yang dapat dilihat pada tabel 5.18.

Tabel 5. 16 Kode Program Controller Custom

```

Custom.php
1 //load model
2 public function __construct()
3 {
4     parent::__construct();
5     $this->load->model('Custom_model');
6     $this->load->model('header_transaksi_custom_model');
7
8     $this->load->model('Produk_model');
9 }
10
11 //list data produk
12 public function index()
13 {
14     $site = $this->konfigurasi_model->
15 >get_all_konfigurasi();
16     //ambil data total
17     $total = $this->Custom_model->total_produk();
18     //paginasi
19     $this->load->library('pagination');
20
21     $config['base_url'] =
22 base_url().'custom/index/';
23     $config['total_rows'] = $total->total;
24     $config['use_page_numbers'] = TRUE;
25     $config['per_page'] = 8;
26     $config['uri_segment'] = 3;
27     $config['num_links'] = 5;
28     $config['full_tag_open'] = '<nav><ul
29 class="pagination">';
30     $config['full_tag_close'] = '</ul></nav>';
31     $config['first_link'] = 'First';
32     $config['first_tag_open'] = '<li class="page-item">';
33     $config['first_tag_close'] = '</li>';
34     $config['last_link'] = 'Last';
35     $config['last_tag_open'] = '<li class="page-item">';
36     $config['last_tag_close'] = '</li>';
37     $config['next_link'] = '&raquo;';
38     $config['next_tag_open'] = '<li class="page-item">';
39     $config['next_tag_close'] = '</li>';
40     $config['prev_link'] = '&laquo;';
41     $config['prev_tag_open'] = '<li class="page-item">';
42     $config['prev_tag_close'] = '</li>';
43     $config['cur_tag_open'] = '<li class="page-item
44 active"><a class="page-link" href="#">';
45     $config['cur_tag_close'] = '</a></li>';

```



```

46     $config['num_tag_open'] = '<li class="page-item">';
47     $config['num_tag_close'] = '</li>';
48     $config['attributes'] = array('class' => 'page-
49 link');
50     $config['first_url'] = base_url().'custom/';
51
52     //styling
53
54
55 //initialize
56 $this->pagination->initialize($config);
57
58 //ambil data produk
59 $page = ($this->uri->segment(3)) ? ($this->uri-
60 >segment(3)-1) * $config['per_page']:0;
61 $produk = $this->Custom_model-
62 >produk($config['per_page'], $page);
63
64 $data = array('title' => 'Produk
65 '.$site->namaweb,
66 'site' => $site,
67 'produk' => $produk,
68 'pagin' => $this-
69 >pagination->create_links(),
70 'isi' =>
71 'custom/list' );
72 $this->load->view('layout/wrapper', $data, FALSE);
73 }
74
75 //detail produk
76 public function detail($slug_produk)
77 {
78     $site = $this->konfigurasi_model-
79 >get_all_konfigurasi();
80     $produk = $this->Produk_model->read($slug_produk);
81     $id_produk = $produk->id_produk;
82     $gambar = $this->Produk_model->gambar($id_produk);
83
84     $data = array('title' => $produk->
85 >nama_produk,
86 'site' => $site,
87 'produk' => $produk,
88 'gambar' => $gambar,
89 'isi' =>
90 'custom/detail' );
91 $this->load->view('layout/wrapper', $data, FALSE);
92 }
93
94 public function checkout()
95 {
96     // Validation input
97     $kode_transaksi = date('dmY')
98     strtoupper(random_string('alnum', 8));
99
100    if($this->session->userdata('email'))
101    {
102        $email = $this->session->userdata('email');

```

```
103     $nama_pelanggan = $this->session->
104     >userdata('nama_pelanggan');
105     $pelanggan = $this->pelanggan_model->
106     >sudah_login($email,$nama_pelanggan);
107
108     // Validation input
109     $this->form_validation->
110     >set_rules('nama_pelanggan','Nama lengkap','required');
111     $this->form_validation->set_rules('telepon','Nomor
112     telepon','required');
113     $this->form_validation->
114     >set_rules('alamat','Alamat','required');
115     $this->form_validation->
116     >set_rules('email','Email','required|valid_email');
117
118     $this->form_validation->set_rules('lingkar_dada','Lingkar
119     Dada','required');
120     $this->form_validation->
121     >set_rules('lingkar_bahu','Lingkar Bahu','required');
122     $this->form_validation->
123     >set_rules('lingkar_pinggang','Lingkar Pinggang','required');
124     $this->form_validation->
125     >set_rules('panjang_tangan','Panjang Tangan','required');
126     $this->form_validation->
127     >set_rules('panjang_baju','Panjang Baju','required');
128     $this->form_validation->
129     >set_rules('busui','Busui','required');
130
131     $config['upload_path'] = './assets/upload/custom size/';
132     $config['allowed_types'] = 'gif|jpg|png|jpeg';
133     $config['max_size'] = '2400'; //kb
134     $config['max_width'] = '2024';
135     $config['max_height'] = '2024';
136
137     $this->load->library('upload', $config);
138
139     if (! $this->upload->do_upload('ref_warna')){
140
141         $data = array('title' => 'Request Kebaya',
142                     'error' => $this->upload->
143         >display_errors(),
144                     'isi' => 'custom/list'
145         );
146         $this->load->view('layout/wrapper', $data, FALSE);
147     }
148     }else{
149     $upload_gambar = array('upload_data' => $this->upload->
150     >data());
151     //thumbnail
152     $img_name = $upload_gambar['upload_data']['file_name'];
153
154     $this->session->set_flashdata('img_name', $img_name);
155
156     $nama_produk = $this->input->post('name');
157     $harga_custom = $this->input->post('price');
158     $lingkar_dada = $this->input->post('lingkar_dada');
159     $busui = $this->input->post('busui');
```




```
160
161 if($lingkar_dada <= 110 && $busui == 'Busui'){
162     $harga_custom = $harga_custom+35000;
163 }else if ($lingkar_dada >= 110 && $busui == 'Busui!'){
164     $harga_custom = $harga_custom+60000;
165 }else if($lingkar dada <= 110 && $busui == 'Non-Busui'){
166     $harga custom = $harga_custom+25000;
167 }else if ($lingkar_dada >= 110 && $busui == 'Non-Busui'){
168     $harga_custom = $harga_custom+50000;}
169
170 if ($this->form_validation->run() === FALSE){
171     //end validasi
172
173     $data = array(
174         'title' => 'Checkout',
175         'pelanggan' => $pelanggan,
176         'isi' => 'belanja/checkout'
177     );
178     $this->session->set_flashdata('price', $harga_custom);
179     $this->session->set_flashdata('nama_produk',
180     $nama_produk );
181     $this->session->set_flashdata('type', 'custom');
182     $this->session->set_flashdata('kode',
183     $kode_transaksi);
184     $this->session->set_flashdata('lingkar_dada', $this->
185     >input->post('lingkar_dada'));
186     $this->session->set_flashdata('lingkar_bahu', $this->
187     >input->post('lingkar_bahu'));
188     $this->session->set_flashdata('lingkar_pinggang',
189     $this->input->post('lingkar_pinggang'));
190     $this->session->set_flashdata('panjang_tangan', $this->
191     >input->post('panjang_tangan'));
192     $this->session->set_flashdata('panjang_baju', $this->
193     >input->post('panjang_baju'));
194     $this->session->set_flashdata('busui', $this->input->
195     >post('busui'));
196     $this->session->set_flashdata('keterangan', $this->
197     >input->post('keterangan'));
198     $this->session->set_flashdata('harga_custom',
199     $harga_custom);
200
201     $this->load->view('layout/wrapper', $data, FALSE);
202     // masuk db
203 } else {
204     $data = array(
205         'nama_produk' => $nama_produk,
206         'harga_custom' => $harga_custom,
207         'lingkar_dada' => $this->input->
208         >post('lingkar_dada'),
209         'lingkar_bahu' => $this->input->
210         >post('lingkar_bahu'),
211         'lingkar_pinggang' => $this->input->
212         >post('lingkar_pinggang'),
213         'panjang_tangan' => $this->input->
214         >post('panjang_tangan'),
215         'panjang_baju' => $this->input->
216         >post('panjang_baju'),
```

```
217 'busui' => $this->input->post('busui'),
218 'ref_warna' => $img_name,
219 'keterangan' => $this->input->post('keterangan'),
220 'kode_transaksi' => $kode_transaksi
221 );
222 $this->Custom_model->tambah($data);
223
224 $idProduk = $this->Custom_model->getId($kode_transaksi);
225
226 $data = array('id_pelanggan' => $pelanggan->
227 >id_pelanggan,
228 'nama_pelanggan' => $this->input->
229 >post('nama_pelanggan'),
230 'email' => $this->input->
230 >post('email'),
231 'telepon' => $this->input->
232 >post('telepon'),
233 'alamat' => $this->input->
234 >post('alamat'),
235 'kode_transaksi' => $this->input->
236 >post('kode_transaksi'),
237 'tanggal_transaksi' => $this->input->
238 >post('tanggal_transaksi'),
239 'jumlah_transaksi' => $this->input->
240 >post('jumlah_transaksi'),
241 'status_bayar' => $result->
242 >transaction_status
243 );
244 // masuk ke header transaksi
245 $this->header_transaksi_custom_model->
246 >tambahCustom($data);
247
248 $this->session->set_flashdata('sukses', 'Checkout
249 berhasil');
250 redirect(base_url('request/sukses'), 'refresh');
251 }
252 }
253
254 //end masuk database
255
256 } else {
257 // klo blm hrs regist
258 $this->session->set_flashdata('sukses', 'Silahkan login atau
259 registrasi terlebih dahulu');
260 redirect(base_url('masuk'), 'refresh');
261 }
262 }
263
264 // sukses
265 public function sukses()
266 {
267 $data = array('title' => 'Halaman Konfirmasi',
268 'isi' => 'custom/sukses');
269 $this->load->view('layout/wrapper', $data, FALSE);
270 }
271
272 public function theDumper() {
```



```

echo $this->request_model->getId('03042021N8MPXEAZ');
exit;
}

```

Tabel 5. 17 Kode program Controller Data Custom Order

```

Admin/ Custom.php
1 //load model
2 public function __construct()
3 {
4     parent::__construct();
5     $this->load->model('Custom_model');
6     $this->load->model('header_transaksi_custom_model');
7     $params = array('server_key' => 'SB-Mid-server-Br1RN71k-
8     jFUfUV8Q0UOGxN1', 'production' => false);
9     $this->load->library('veritrans');
10    $this->veritrans->config($params);
11    //proteksi halaman
12    $this->simple_login->cek_login();
13 }
14
15 public function index()
16 {
17     $header_transaksi_custom = $this-
18 >header_transaksi_custom_model-
19 >get_all_header_transaksi_custom();
20
21     $data = array('title' => 'Laporan Custom',
22                 'header_transaksi_custom' =>
23 $header_transaksi_custom,
24                 'isi' => 'admin/custom/list');
25     $this->load->view('admin/layout/main', $data, FALSE);
26 }
27
28 //Detail
29 public function detail($kode_transaksi)
30 {
31     $header_transaksi_custom = $this-
32 >header_transaksi_custom_model->kode_transaksi($kode_transaksi);
33     $custom_size = $this->Custom_model-
34 >detail($kode_transaksi);
35
36     $data = array('title' => 'Pesanan Custom',
37                 'header_transaksi_custom' =>
38 $header_transaksi_custom,
39                 'custom_size' => $custom_size,
40                 'isi' =>
41 'admin/custom/detail');
42     $this->load->view('admin/layout/main', $data, FALSE);
43 }

```

```

44
45 public function cek_status()
46 {
47     $orderid = $this->input->post('order_id');
48
49     if($orderid){
50         $this->status($orderid);
51     }else{
52         echo 'Order id yang anda cari tidak ada';
53     }
54 }
55
56 private function status($orderid)
57 {
58
59     $result = $this->veritrans->status($orderid);
60     $dataupdate = [
61         'status_bayar' => $result->transaction_status
62     ];
63
64     $where = [
65         'kode_transaksi' => $orderid
66     ];
67
68     $update= $this->header_transaksi_custom_model->
69     >update($dataupdate, $where);
70
71     redirect(base_url('admin/custom'),'refresh');
72
73 }

```

Tabel 5. 18 Kode Program Model Custom

```

Custom_model.php
1 public function __construct()
2 {
3     parent::__construct();
4     $this->load->database();
5 }
6
7 // REad produk
8 public function read($slug_produk)
9 {
10     $this->db->select('produk.*,
11         users.nama,
12         kategori.nama_kategori,
13         kategori.slug_kategori,
14         COUNT(gambar.id_gambar) AS
15         total_gambar');
16     $this->db->from('produk');
17     //join
18     $this->db->join('users','users.id_user
19     =
20     produk.id_user','left');
21     $this->db->join('kategori','kategori.id_kategori
22     =
23     produk.id_kategori','left');
24     $this->db->join('gambar','gambar.id_produk

```



```

23     produk.id_produk', 'left');
24 // end
25     $this->db->where('produk.status_produk', 'Publish');
26     $this->db->where('produk.tipe_produk', 'Custom');
27     $this->db->where('produk.slug_produk', $slug_produk);
28     $this->db->group_by('produk.id_produk');
29     $this->db->order_by('id_produk', 'desc');
30     $query = $this->db->get();
31     return $query->row();
32 }
33
34 //PRODUK
35 public function produk($limit, $start)
36 {
37     $this->db->select('produk.*,
38                     users.nama,
39                     kategori.nama_kategori,
40                     kategori.slug_kategori,
41                     COUNT(gambar.id_gambar) AS
42     total_gambar');
43     $this->db->from('produk');
44     //join
45     $this->db->join('users', 'users.id_user
46     produk.id_user', 'left');
47     $this->db->join('kategori', 'kategori.id_kategori =
48     produk.id_kategori', 'left');
49     $this->db->join('gambar', 'gambar.id_produk
50     produk.id_produk', 'left');
51     // end
52     $this->db->where('produk.status_produk', 'Publish');
53     $this->db->where('produk.tipe_produk', 'Custom');
54     $this->db->group_by('produk.id_produk');
55     $this->db->order_by('id_produk', 'desc');
56     $this->db->limit($limit, $start);
57     $query = $this->db->get();
58     return $query->result();
59 }
60
61 //total produk
62 public function total_produk()
63 {
64     $this->db->select('COUNT(*) AS total');
65     $this->db->from('produk');
66     $this->db->where('status_produk', 'Publish');
67     $this->db->where('tipe_produk', 'Custom');
68     $query = $this->db->get();
69     return $query->row();
70 }
71
72 // detail gambar produk
73 public function detail_gambar($id_gambar)
74 {
75     $this->db->select('*');
76     $this->db->from('gambar');
77     $this->db->where('id_gambar', $id_gambar);
78     $this->db->order_by('id_gambar', 'desc');
79     $query = $this->db->get();

```

```

80     return $query->row();
81 }
82
83 //gambar
84     public function gambar($id_produk){
85         $this->db->select('*');
86         $this->db->from('gambar');
87         $this->db->where('id_produk',$id_produk);
88         $this->db->order_by('id_gambar','desc');
89         $query = $this->db->get();
90     return $query->result();
91     }
92
93     public function get_all_custom()
94     {
95         $this->db->select('*');
96         $this->db->from('custom_size');
97         $this->db->order_by('id_csm_size','desc');
98         $query = $this->db->get();
99         return $query->result();
100    }
101
102    public function getId($kodeTransaksi)
103    {
104        $this->db->select('id_csm_size');
105        $this->db->from('custom_size');
106        $this->db->where('kode_transaksi',$kodeTransaksi);
107        $query = $this->db->get();
108        return $query->row()->id_csm_size;
109    }
110
111    public function detail($kode_transaksi)
112    {
113        $this->db->select('*');
114        $this->db->from('custom_size');
115        $this->db->where('kode_transaksi',$kode_transaksi);
116        $this->db->order_by('kode_transaksi','desc');
117        $query = $this->db->get();
118        return $query->row();
119    }
120
121    // Tambah custom
122    public function tambahCustom($data)
123    {
124        $this->db->insert('custom_size', $data);
125    }

```

Tabel 5. 19 Kode Program Model Header Transaksi Custom

```

Header_transaksi_custom_model.php
1     public function __construct()
2     {
3         parent::__construct();
4         $this->load->database();
5     }

```



```
6
7 public function get_all_header_transaksi_custom()
8 {
9     $this->db->select('*');
10    $this->db->from('header_transaksi_custom');
11    $this->db->order_by('id_header_transaksi_custom', 'desc');
12    $query = $this->db->get();
13    return $query->result();
14 }
15
16 public function pelanggan($id_pelanggan)
17 {
18     $this->db->select('header_transaksi_custom.*,
19                    SUM(custom_size.jumlah) AS
20 total_item');
21    $this->db->from('header_transaksi_custom');
22    $this->db->where('header_transaksi_custom.id_pelanggan',
23 $id_pelanggan);
24    $this->db->join('custom_size', 'custom_size.kode_transaksi
25 = header_transaksi_custom.kode_transaksi', 'left');
26
27    $this->db-
28 >group by('header_transaksi_custom.id_header_transaksi_custom');
29    $this->db->order_by('id_header_transaksi_custom', 'desc');
30    $query = $this->db->get();
31    return $query->result();
32 }
33
34 // tambah custom
35 public function tambahCustom($data)
36 {
37     $this->db->insert('header_transaksi_custom', $data);
38 }
39
40 //UPDATE STATUS BAYAR
41 public function update($dataupdate,$where)
42 {
43     $this->db->update('header_transaksi_custom',
44 $dataupdate,$where );
45    return $this->db->affected_rows();
46 }
47
48 // detail header_transaksi
49 public function detail($id_header_transaksi_custom)
50 {
51     $this->db->select('*');
52     $this->db->from('header_transaksi_custom');
53     $this->db->where('id_header_transaksi_custom',
54 $id_header_transaksi_custom);
55     $this->db->order_by('id_header_transaksi_custom', 'desc');
56     $query = $this->db->get();
57     return $query->row();
58 }
59
60
61 public function kode_transaksi($kode_transaksi)
62 {
```

```

63     $this->db->select('*');
64     $this->db->from('header_transaksi_custom');
65     $this->db->join('custom_size', 'custom_size.kode_transaksi =
66     header_transaksi_custom.kode_transaksi', 'left');
67     $this->db->join('pelanggan', 'pelanggan.id_pelanggan =
68     header_transaksi_custom.id_pelanggan', 'left');
69     $this->db-
70     >group_by('header_transaksi_custom.id_header_transaksi_custom');
71     $this->db->where('custom_size.kode_transaksi',
72     $kode_transaksi);
73     $this->db->order_by('id_header_transaksi_custom', 'desc');
74     $query = $this->db->get();
75     return $query->row();
76     }

```

5.5.4 Kode Program Checkout

Hasil implementasi kode program untuk proses checkout oleh aktor pelanggan ditunjukkan pada tabel 5.19. Pada Controller dengan nama Belanja ini terdapat method index yang berfungsi dalam menampilkan daftar keranjang. Pada Controller ini juga terdapat fungsi checkout yang dilakukan untuk melakukan transaksi pembelian produk, yang nantinya akan diteruskan dengan proses pembayaran. Dan method sukses untuk menampilkan halaman transaksi pembelian berhasil.

Setelah melakukan pemesanan, data tersebut akan dikelola pada model header transaksi dapat dilihat pada tabel 5.21 untuk menyimpan informasi yang akan ditampilkan pada halaman riwayat belanja, dan Pada tabel 5.22 merupakan kelas model untuk transaksi yang berisi informasi detail pesanan produk ready. Fungsi Controller Transaksi pada tabel 5.20 pada folder admin inilah yang akan digunakan untuk menampilkan data pesanan pada halaman admin, selain itu pada controller ini berisikan juga perintah untuk melakukan update status pembayaran melalui method cek_status dan juga dapat mencetak detail pesanan pada method cetak.

Tabel 5. 20 Kode Program Controller Belanja

```

Belanja.php
1 //load model
2 public function __construct()
3 {
4     parent::__construct();
5     $this->load->model('Produk_model');
6     $this->load->model('Kategori_model');
7     $this->load->model('Konfigurasi_model');
8     $this->load->model('pelanggan_model');
9     $this->load->model('header_transaksi_model');
10    $this->load->model('transaksi_model');
11    //load helper
12    $this->load->helper('string');
13 }
14

```



```
15 //halmaan belanja
16 public function index()
17 {
18     $keranjang = $this->cart->contents();
19
20     $data = array('title' => 'Keranjang Belanja',
21                 'keranjang' => $keranjang,
22                 'isi' => 'belanja/list' );
23     $this->load->view('layout/wrapper', $data, FALSE);
24 }
25
26 //sukses belanja
27 public function sukses()
28 {
29
30     $data = array(
31         'title' => 'Belanja berhasil',
32         'isi' => 'belanja/sukses'
33     );
34     $this->load->view('layout/wrapper', $data, FALSE);
35 }
36
37 public function checkout()
38 {
39     if($this->session->userdata('email'))
40     {
41         $email = $this->session->userdata('email');
42         $nama_pelanggan = $this->session-
43 >userdata('nama_pelanggan');
44         $pelanggan = $this->pelanggan_model-
45 >sudah_login($email,$nama_pelanggan);
46
47         $keranjang = $this->cart->contents();
48
49         // Validation input
50         $this->form_validation-
51 >set_rules('nama_pelanggan', 'Nama lengkap', 'required');
52         $this->form_validation->set_rules('telepon', 'Nomor
53 telepon', 'required');
54         $this->form_validation-
55 >set_rules('alamat', 'Alamat', 'required');
56         $this->form_validation-
57 >set_rules('email', 'Email', 'required|valid_email');
58
59         if($this->form_validation->run()===FALSE)
60         {
61             $data = array('title' => 'Checkout',
62                 'keranjang' => $keranjang,
63                 'pelanggan' => $pelanggan,
64                 'isi' => 'belanja/checkout'
65             );
66             $this->load->view('layout/wrapper', $data, FALSE);
67
68             //masuk database
69         }else{
70             $data = array('id_pelanggan' => $pelanggan-
71 >id_pelanggan,
```

```

72     'nama_pelanggan' => $this->input->
73     >post('nama_pelanggan'),
74     'email' => $this->input->
75     >post('email'),
76     'telepon' => $this->input->
77     >post('telepon'),
78     'alamat' => $this->input->
79     >post('alamat'),
80     'kode_transaksi' => $this->input->
81     >post('kode_transaksi'),
82     'tanggal_transaksi' => $this->input->
83     >post('tanggal_transaksi'),
84     'jumlah_transaksi' => $this->input->
85     >post('jumlah_transaksi'),
86     'status_bayar' => $result->
87     >transaction_status,
88     'tanggal_post' => date('Y-m-d
89     H:i:s')
90     );
91     //masuk header_transaksi
92     $this->header_transaksi_model->tambah($data);
93
94     //masuk tabel_transaksi
95     foreach ($keranjang as $keranjang)
96     {
97         $sub_total = $keranjang['price']
98         $keranjang['qty'];
99         $data = array( 'id_pelanggan' =>
100         $pelanggan->id_pelanggan,
101         'kode_transaksi' => $this->
102         >input->post('kode_transaksi'),
103         'id_produk'
104         $keranjang['id'],
105         'harga'
106         $keranjang['price'],
107         'jumlah'
108         $keranjang['qty'],
109         'total_harga'
110         $sub_total,
111         'tanggal_transaksi' => $this->
112         >input->post('tanggal_transaksi')
113         );
114         $this->transaksi_model->tambah($data);
115     }
116     //end
117     //proses kosongkan keranjang
118     $this->cart->destroy();
119
120     $this->session->set_flashdata('sukses', 'Checkout
121     berhasil');
122     redirect(base_url('belanja/sukses'),'refresh');
123 }
124 //end masuk database
125 }else{
126     $this->session->set_flashdata('warning', 'Silahkan
127     login atau registrasi terlebih dahulu');
128     redirect(base_url('masuk'),'refresh');

```



```
129 }
130 }
131
132 //tambah ke keranjang
133 public function add()
134 {
135     $id = $this->input->post('id');
136     $qty = $this->input->post('qty');
137     $price = $this->input->post('price');
138     $name = $this->input->post('name');
139     $redirect_page = $this->input->post('redirect_page');
140
141     //proses add to cart
142     $data = array(
143         'id' => $id,
144         'qty' => $qty,
145         'price' => $price,
146         'name' => $name
147     );
148
149     $this->cart->insert($data);
150     //redirect page
151     redirect($redirect_page, 'refresh');
152 }
153
154 //update
155 public function update_cart($rowid)
156 {
157     //jika ada data
158     if($rowid)
159     {
160         $data = array('rowid' => $rowid,
161                     'qty' => $this->input->post('qty'))
162     );
163     $this->cart->update($data);
164     $this->session->set_flashdata('sukses', 'Data
165 keranjang telah diupdate');
166     redirect(base_url('belanja'), 'refresh');
167 }else{
168     redirect(base_url('belanja'), 'refresh');
169 }
170 }
171
172 //hapus semua isi keranjang
173 public function hapus($rowid='')
174 {
175     if($rowid){
176         $this->cart->remove($rowid);
177         $this->session->set_flashdata('sukses', 'Data
178 keranjang belanja telah dihapus');
179         redirect(base_url('belanja'), 'refresh');
180     }else{
181         //delete all
182         $this->cart->destroy();
183         $this->session->set_flashdata('sukses', 'Data
184 keranjang belanja telah dihapus');
185         redirect(base_url('belanja'), 'refresh');
```

186	}
187	Brawijaya } Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya
188	Brawijaya } Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya

Tabel 5. 21 Kode Program Controller Data Pesanan

```

Admin/ Transaksi.php
1 //load model
2 public function __construct()
3 {
4     parent::__construct();
5     $this->load->model('konfigurasi_model');
6     $this->load->model('header_transaksi_model');
7     $this->load->model('transaksi_model');
8     $params = array('server_key' => 'SB-Mid-server-Br1RN71k-
9     jFUfUV8Q0UOGxN1', 'production' => false);
10    $this->load->library('veritrans');
11    $this->veritrans->config($params);
12    //proteksi halaman
13    $this->simple_login->cek_login();
14 }
15
16 public function index()
17 {
18     $header_transaksi = $this->header_transaksi_model-
19     >get_all_header_transaksi();
20     $data = array ( 'title' => 'Data Transaksi',
21                  'header_transaksi' => $header_transaksi,
22                  'isi' =>
23                  'admin/transaksi/list' );
24     $this->load->view('admin/layout/main', $data, FALSE);
25 }
26
27
28 public function cek_status()
29 {
30     $orderid = $this->input->post('order_id');
31
32     if($orderid){
33         $this->status($orderid);
34     }else{
35         echo 'Order id yang anda cari tidak ada';
36     }
37 }
38
39 private function status($orderid)
40 {
41
42     $result = $this->veritrans->status($orderid) ;
43     $dataupdate = [
44         'status_bayar' => $result->transaction_status
45     ];
46 }
47
48     $where = [
49         'kode_transaksi' => $orderid

```



```
50 ];
51
52 $update= $this->header_transaksi_model->update($dataupdate,
53 $where);
54
55 redirect(base_url('admin/transaksi'),'refresh');
56 }
57
58 //Detail transaksi
59 public function detail($kode_transaksi)
60 {
61     $header_transaksi = $this->header_transaksi_model-
62 >kode_transaksi($kode_transaksi);
63     $transaksi
64 >kode_transaksi($kode_transaksi);
65
66     $data = array('title' => 'Detail Transaksi',
67                 'header_transaksi' => $header_transaksi,
68                 'transaksi' => $transaksi,
69                 'isi' =>
70 'admin/transaksi/detail'
71 );
72     $this->load->view('admin/layout/main',
73 $data, FALSE);
74 }
75
76 //Cetak
77 public function cetak($kode_transaksi)
78 {
79     $header_transaksi = $this->header_transaksi_model-
80 >kode_transaksi($kode_transaksi);
81     $transaksi
82 >kode_transaksi($kode_transaksi);
83     $site
84 >get_all_konfigurasi();
85
86     $data = array('title' => 'Riwayat Belanja',
87                 'header_transaksi' => $header_transaksi,
88                 'transaksi' => $transaksi,
89                 'site' => $site
90 );
91     $this->load->view('admin/transaksi/cetak',
92 $data, FALSE);
93 }
94 }
```

Tabel 5. 22 Kode Program Model Header Transaksi

```

Header_transaksi_model.php
1 public function __construct()
2 {
3     parent::__construct();
4     $this->load->database();
5 }
6
7 public function get_all_header_transaksi()
8 {
9     $this->db->select('header_transaksi.*,
10         pelanggan.nama_pelanggan,
11         SUM(transaksi.jumlah) AS total_item');
12     $this->db->from('header_transaksi');
13     $this->db->join('transaksi', 'transaksi.kode_transaksi =
14         header_transaksi.kode_transaksi', 'left');
15     $this->db->join('pelanggan', 'pelanggan.id_pelanggan =
16         header_transaksi.id_pelanggan', 'left');
17     $this->db->group_by('header_transaksi.id_header_transaksi');
18     $this->db->order_by('id_header_transaksi', 'desc');
19     $query = $this->db->get();
20     return $query->result();
21 }
22
23 public function pelanggan($id_pelanggan)
24 {
25     $this->db->select('header_transaksi.*,
26         SUM(transaksi.jumlah) AS
27         total_item');
28     $this->db->from('header_transaksi');
29     $this->db->where('header_transaksi.id_pelanggan',
30         $id_pelanggan);
31     //join
32     $this->db->join('transaksi', 'transaksi.kode_transaksi =
33         header_transaksi.kode_transaksi', 'left');
34     //end
35     $this->db-
36     >group_by('header_transaksi.id_header_transaksi');
37     $this->db->order_by('id_header_transaksi', 'desc');
38     $query = $this->db->get();
39     return $query->result();
40 }
41
42 public function kode_transaksi($kode_transaksi)
43 {
44     $this->db->select('header_transaksi.*,
45         pelanggan.nama_pelanggan,
46         SUM(transaksi.jumlah) AS total_item');
47     $this->db->from('header_transaksi');
48     $this->db->join('transaksi', 'transaksi.kode_transaksi =
49         header_transaksi.kode_transaksi', 'left');
50     $this->db->join('pelanggan', 'pelanggan.id_pelanggan =
51         header_transaksi.id_pelanggan', 'left');
52     $this->db-
53     >group_by('header_transaksi.id_header_transaksi');

```




```

54 $this->db-
55 >where('transaksi.kode_transaksi',$kode_transaksi);
56 $this->db->order_by('id_header_transaksi','desc');
57 $query = $this->db->get();
58 return $query->row();
59 }
60
61 public function detail($id_header_transaksi)
62 {
63 $this->db->select('*');
64 $this->db->from('header_transaksi');
65 $this->db-
66 >where('id_header_transaksi',$id_header_transaksi);
67 $this->db->order_by('id_header_transaksi','desc');
68 $query = $this->db->get();
69 return $query->row();
70 }
71
72 //UPDATE STATUS BAYAR
73 public function update($dataupdate,$where)
74 {
75 $this->db->update('header_transaksi', $dataupdate,$where );
76 return $this->db->affected_rows();
77 }
78
79 // tambah
80 public function tambah($data)
81 {
82 $this->db->insert('header_transaksi', $data);
83 }
84

```

Tabel 5. 23 Kode Program Model transaksi

Transaksi_model.php

```

1 public function __construct()
2 {
3     parent::__construct();
4     $this->load->database();
5 }
6
7 public function get_all_transaksi()
8 {
9     $this->db->select('*');
10    $this->db->from('transaksi');
11    $this->db->order_by('id_transaksi','desc');
12    $query = $this->db->get();
13    return $query->result();
14 }
15
16 public function kode_transaksi($kode_transaksi)
17 {
18    $this->db->select('transaksi.*,
19                    produk.nama_produk,
20                    produk.kode_produk');
21    $this->db->from('transaksi');

```

```

22 //join
23 $this->db->join('produk','produk.id_produk =
24 transaksi.id_produk', 'left');
25 //end
26 $this->db->where('kode_transaksi', $kode_transaksi);
27 $this->db->order_by('id_transaksi', 'desc');
28 $query = $this->db->get();
29 return $query->result();
30 }
31
32 public function transaksi_pelanggan($kode_transaksi)
33 {
34     $this->db->select('transaksi.*,
35                     produk.nama_produk');
36     $this->db->from('transaksi');
37     //join
38     $this->db->join('produk','produk.id_produk =
39 transaksi.id_produk', 'left');
40     //end
41     $this->db->where('kode_transaksi', $kode_transaksi);
42     $this->db->order_by('id_transaksi', 'desc');
43     $query = $this->db->get();
44     return $query->result();
45 }
46
47 public function detail($id_transaksi)
48 {
49     $this->db->select('*');
50     $this->db->from('transaksi');
51     $this->db->where('id_transaksi',$id_transaksi);
52     $this->db->order_by('id_transaksi', 'desc');
53     $query = $this->db->get();
54     return $query->row();
55 }
56
57 // detail slug transaksi
58 public function read($slug_transaksi)
59 {
60     $this->db->select('*');
61     $this->db->from('transaksi');
62     $this->db->where('slug_transaksi',$slug_transaksi);
63     $this->db->order_by('id_transaksi', 'desc');
64     $query = $this->db->get();
65     return $query->row();
66 }
67
68 //login
69 public function login($username, $password)
70 {
71     $this->db->select('*');
72     $this->db->from('users');
73     $this->db->where(array('username' => $username,
74                          'password'=> SHA1($password)));
75     $this->db->order_by('id_user','desc');
76     $query = $this->db->get();
77     return $query->row();
78 }

```



```

79
80 // tambah
81 public function tambah($data)
82 {
83     $this->db->insert('transaksi', $data);
84 }
    
```

5.5.5 Kode Program Pembayaran

Hasil implementasi kode program untuk proses pembayaran oleh aktor pelanggan dan midtrans. ditunjukkan pada tabel 5.23. Pada Controller dengan nama Snap ini terdapat Server Key pada baris 4-7 yang digunakan untuk melakukan integrasi dengan sand box API payment gateway, kemudian terdapat method token yang berisi informasi data data detail transaksi yang akan ditampilkan pada snap payment midtrans, setelah pelanggan melakukan pembayaran akan dieksekusi fungsi finish pada controller ini yang berisikan perintah untuk menginputkan data pesanan ready maupun custom pada database masing masing.

Pada tabel 2.24 merupakan view dari kelas checkout yang merupakan frontend halaman checkout dan pembayaran, dapat dilihat pada baris 135 merupakan dimulainya form yang nantinya data tersebut akan disimpan pada database setelah proses selesai, namun proses dimulai saat pelanggan menekan tombol pada baris 247 yang memiliki id pay-button akan menjalankan proses ajax pada halaman checkout dimulai baris 274 akan mengarah pada fungsi token pada controller snap, selanjutnya proses pada controller snap ini seperti yang sudah dijelaskan diatas.

Tabel 5. 24 Kode Program Controller Pembayaran

```

Snap.php
1 public function __construct()
2 {
3     parent::__construct();
4     $params = array('server_key' => 'SB-Mid-server-Br1RN71k-
5 jFUUV8Q0UOGxN1', 'production' => false);
6     $this->load->library('midtrans');
7     $this->midtrans->config($params);
8     $this->load->helper('url');
9     $this->load->model('Produk_model');
10    $this->load->model('Kategori_model');
11    $this->load->model('Konfigurasi_model');
12    $this->load->model('pelanggan_model');
13    $this->load->model('header_transaksi_model');
14    $this->load->model('transaksi_model');
15    $this->load->model('custom_model');
16    $this->load->model('header_transaksi_custom_model');
17
18    //load helper
19    $this->load->helper('string');
20 }
    
```

```
21
22 public function index()
23 {
24     $this->load->view('belanja/checkout');
25 }
26
27 public function token()
28 {
29     $kode_transaksi = $this->input->get('kode_transaksi');
30     $nama_pelanggan = $this->input->get('nama_pelanggan');
31     $email = $this->input->get('email');
32     $telepon = $this->input->get('telepon');
33     $jumlah_transaksi = $this->input->
34     >get('jumlah_transaksi');
35
36     $keranjang = $this->cart->contents();
37     // Required
38     $transaction_details = array(
39         'order_id' => $kode_transaksi,
40         'gross_amount' => $jumlah_transaksi, // no decimal
41         allowed for creditcard
42     );
43
44     if ($this->session->flashdata('type') == 'custom'){
45         // Optional
46         $item_details = array(
47             'id' => 'al',
48             'price' => $jumlah_transaksi,
49             'quantity' => 1,
50             'name' => "Custom Kebaya"
51         );
52     }else{
53
54         foreach($keranjang as $keranjang){
55             $item_details[] = array(
56                 'id'
57 => $keranjang['id'],
58                 'price'=>
59 $keranjang['price'],
60                 'quantity' =>
61 $keranjang['qty'],
62                 'name' =>
63 $keranjang['name']
64             );
65         }
66     }
67
68     // Optional
69     $billing_address = array(
70         'first_name' => "Andri",
71         'last_name' => "Litani",
72         'address' => "Mangga 20",
73         'city' => "Jakarta",
74         'postal_code' => "16602",
75         'phone' => "081122334455",
76         'country_code' => 'IDN'
77     );
```



```
78
79 // Optional
80 $shipping_address = array(
81     'first_name' => "Obet",
82     'last_name'  => "Supriadi",
83     'address'    => "Manggis 90",
84     'city'       => "Jakarta",
85     'postal_code' => "16601",
86     'phone'      => "08113366345",
87     'country_code' => 'IDN'
88 );
89
90 // Optional
91 $customer_details = array(
92     'first_name' => $nama_pelanggan,
93     // 'last_name' => "Litani",
94     'email'      => $email,
95     'phone'      => $telepon,
96     'billing_address' => $billing_address,
97     'shipping_address' => $shipping_address
98 );
99
100 // Data yang akan dikirim untuk request redirect url.
101 $credit_card['secure'] = true;
102 //ser save_card true to enable oneclick or 2click
103 // $credit_card['save_card'] = true;
104
105 $time = time();
106 $custom_expiry = array(
107     'start_time' => date("Y-m-d H:i:s O", $time),
108     'unit'       => 'day',
109     'duration'   => 1
110 );
111
112 $transaction_data = array(
113     'transaction_details' => $transaction_details,
114     'item_details'       => $item_details,
115     'customer_details'   => $customer_details,
116     'credit_card'        => $credit_card,
117     'expiry'             => $custom_expiry
118 );
119
120 error_log(json_encode($transaction_data));
121 $snapToken = $this->midtrans-
122 >getSnapToken($transaction_data);
123 error_log($snapToken);
124 echo $snapToken;
125 }
126
127
128 public function finish()
129 {
130     $result = json_decode($this->input->post('result_data'));
131     // echo 'RESULT <br><pre>';
132     // var_dump($result);
133     // echo '</pre>';
134 }
```

```
135 if($this->session->userdata('email'))
136 {
137     $email = $this->session->userdata('email');
138     $nama_pelanggan = $this->session->
139     >userdata('nama_pelanggan');
140     $pelanggan = $this->pelanggan_model->
141     >sudah_login($email,$nama_pelanggan);
142
143     if($this->input->post('type') == 'custom') {
144
145         // Validation input
146         $this->form_validation->
147         >set_rules('nama_pelanggan','Nama lengkap','required');
148         $this->form_validation->set_rules('telepon','Nomor
149         telepon','required');
150         $this->form_validation->
151         >set_rules('alamat','Alamat','required');
152         $this->form_validation->
153         >set_rules('email','Email','required|valid_email');
154
155
156         if($this->form_validation->run() == FALSE)
157         {
158             $data = array(
159                 'title' => 'Checkout',
160                 'pelanggan' => $pelanggan,
161                 'isi' => 'belanja/checkout'
162             );
163             $this->session->set_flashdata('type', 'custom');
164             $this->load->view('layout/wrapper', $data,
165             FALSE);
166             //masuk database
167             }else{
168                 $data = array( 'id_pelanggan' =>
169                 $pelanggan->id_pelanggan,
170                 'nama_pelanggan' => $this->
171                 >input->post('nama_pelanggan'),
172                 'email' => $this->
173                 >input->post('email'),
174                 'telepon' => $this->
175                 >input->post('telepon'),
176                 'alamat' => $this->
177                 >input->post('alamat'),
178                 'kode_transaksi' => $this->
179                 >input->post('kode_transaksi'),
180                 'tanggal_transaksi' => $this->
181                 >input->post('tanggal_transaksi'),
182                 'jumlah_transaksi' => $this->
183                 >input->post('jumlah_transaksi'),
184                 'status_bayar' => $result->
185                 >transaction_status
186             );
187             //masuk header_transaksi
188             $this->header_transaksi_custom_model->
189             >tambahCustom($data);
190             // masuk ke tabel CUSTOM
191             $data = array(
```



```
192     'nama_produk' => $this->input->
193     >post('nama_produk'),
194     'harga_custom' => $this->input->
195     >post('harga_custom'),
196     'lingkar_dada' => $this->input->
197     >post('lingkar_dada'),
198     'lingkar_bahu' => $this->input->
199     >post('lingkar_bahu'),
200     'lingkar_pinggang' => $this->input->
201     >post('lingkar_pinggang'),
202     'panjang_tangan' => $this->input->
203     >post('panjang_tangan'),
204     'panjang_baju' => $this->input->
205     >post('panjang_baju'),
206     'busui' => $this->input->post('busui'),
207     'ref_warna' => $this->input->
208     >post('img_name'),
209     'keterangan' => $this->input->
210     >post('keterangan'),
211     'kode_transaksi' => $this->input->
212     >post('kode_transaksi')
213 );
214     $this->custom_model->tambahCustom($data);
215     $this->session->set_flashdata('sukses',
216     'Checkout berhasil');
217     redirect(base_url('custom/sukses'),
218     'refresh');
219 }
220 //end
221
222
223
224     /////// PROSES CHECKOUT BELANJA BIASA ///////
225     } else {
226         $keranjang = $this->cart->contents();
227         // Validation input
228         $this->form_validation->
229         >set_rules('nama_pelanggan', 'Nama lengkap', 'required');
230         $this->form_validation->set_rules('telepon', 'Nomor
231         telepon', 'required');
232         $this->form_validation->
233         >set_rules('alamat', 'Alamat', 'required');
234         $this->form_validation->
235         >set_rules('email', 'Email', 'required|valid_email');
236
237         if($this->form_validation->run() == FALSE)
238         {
239             $data = array('title' => 'Checkout',
240             'keranjang' => $keranjang,
241             'pelanggan' => $pelanggan,
242             'isi' => 'belanja/checkout'
243             );
244             $this->load->view('layout/wrapper', $data, FALSE);
245         }
246         //masuk database
247     } else {
```



```
248     $data = array('id_pelanggan' => $pelanggan->
249 >id_pelanggan,
250     'nama_pelanggan' => $this->input->
251 >post('nama_pelanggan'),
252     'email' => $this->input->
253 >post('email'),
254     'telepon' => $this->input->
255 >post('telepon'),
256     'alamat' => $this->input->
257 >post('alamat'),
258     'kode_transaksi' => $this->input->
259 >post('kode_transaksi'),
260     'tanggal_transaksi' => $this->input->
261 >post('tanggal_transaksi'),
262     'jumlah_transaksi' => $this->input->
263 >post('jumlah_transaksi'),
264     'status_bayar' => $result->
265 >transaction_status,
266     'tanggal_post' => date('Y-m-d
267 H:i:s')
268     );
269     //masuk header_transaksi
270     $this->header_transaksi_model->tambah($data);
271
272     //masuk tabel_transaksi
273     foreach ($keranjang as $keranjang)
274     {
275         $sub_total = $keranjang['price'] *
276 $keranjang['qty'];
277         $data = array('id_pelanggan' =>
278 $pelanggan->id_pelanggan,
279         'kode_transaksi' => $this->
280 >input->post('kode_transaksi'),
281         'id_produk' =>
282 $keranjang['id'],
283         'harga' =>
284 $keranjang['price'],
285         'jumlah' =>
286 $keranjang['qty'],
287         'total_harga' =>
288 $sub_total,
289         'tanggal_transaksi' => $this->
290 >input->post('tanggal_transaksi')
291     );
292     $this->transaksi_model->tambah($data);
293     }
294
295     //proses kosongkan keranjang
296     $this->cart->destroy();
297
298     $this->session->set_flashdata('sukses', 'Checkout
299 berhasil');
300     redirect(base_url('belanja/sukses'),'refresh');
301 }
302 //end masuk database
303 }
304 }else{
```



```

305     $this->session->set_flashdata('sukses', 'Silahkan
306     login atau registrasi terlebih dahulu!');
307     redirect(base_url('registrasi'),'refresh');
308 }
309 }
310

```

Tabel 5. 25 Kode Program View Checkout

```

Checkout.php
1     <!DOCTYPE html>
2     <html lang="en">
3     <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-
6     scale=1">
7
8     <script type="text/javascript"
9         src="https://app.sandbox.midtrans.com/snap/snap.js"
10        data-client-key="SB-Mid-client-
11        lwmbLEprNr6haAI7"></script>
12     <script
13        src="//ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js
14        "></script>
15     </head>
16     <body>
17     <!-- Cart -->
18     <section class="cart bgwhite p-t-70 p-b-100">
19     <div class="container">
20     <div class="container-table-cart pos-relative">
21     <div class="wrap-table-shopping-cart bgwhite">
22
23     <h2><?php echo $title ?></h2> <hr>
24
25     <div class="clearfix"></div>
26
27
28     <?php if ($this->session->flashdata('sukses')) {
29     echo '<div class="alert alert-warning">';
30     echo ' <a href="#" class="close" data-
31     dismiss="alert" aria-label="close">&times;</a>';
32     echo $this->session->flashdata('sukses');
33     echo '</div>';
34     } ?>
35
36     <?php if ($this->session->flashdata('type') !=
37     'custom') { ?>
38
39     <table class="table-shopping-cart">
40     <tr class="table-head">
41     <th class="column-1">GAMBAR</th>
42     <th class="column-2">PRODUK</th>
43     <th class="column-3">HARGA</th>
44     <th class="column-4">JUMLAH</th>
45     <th class="column-5" width="15%">SUB TOTAL</th>

```

```

46 <th class="column-6" width="20%">ACTION</th>
47 </tr>
48
49 <?php
50 // looping data krjng blnj
51 foreach ($keranjang as $keranjang) {
52 // ambil data produk
53 $id_produk = $keranjang['id'];
54 $produk = $this->Produk_model->detail($id_produk);
55 // form update keranjang
56 echo form_open(base_url('belanja/update_cart/'
57 $keranjang['rowid']));
58 >
59 <tr class="table-row">
60 <td class="column-1">
61 <div class="cart-img-product b-rad-4 o-f-hidden">
62 "
65 </div>
66 </td>
67 <td class="column-2"><?php echo $keranjang['name']
68 ?></td>
69 <td class="column-4">Rp. <?php echo
70 number_format($keranjang['price'], '0', ',', '.') ?></td>
71 <td class="column-4">
72 <div class="flex-w bo5 of-hidden w-size17">
73 <button class="btn-num-product-down color1 flex-c-m
74 size7 bg8 eff2">
75 <i class="fs-12 fa fa-minus" aria-
76 hidden="true"></i>
77 </button>
78 <input class="size8 m-text18 t-center num-product"
79 type="number" name="qty" value="<?php echo $keranjang['qty']
80 ?>"
81 >
82 <button class="btn-num-product-up color1 flex-c-m
83 size7 bg8 eff2">
84 <i class="fs-12 fa
85 fa-plus" aria-hidden="true"></i>
86 </button>
87 </div>
88 </td>
89 <td class="column-5">Rp.
90 <?php
91 $sub_total = $keranjang['price']
92 * $keranjang['qty'];
93 echo number_format($sub_total, '0', ',', '.');
94 >
95 </td>
96 <td>
97 <button type="submit" name="update" class="btn btn-
98 dark btn-sm">
99 <i class="fa fa-edit"></i> Update
100 </button>

```




```

103 <a href="php echo base_url('belanja/hapus/' .
104 $keranjang['rowid']) ?" class="btn btn-danger btn-sm">
105 <i class="fa fa-trash-o"></i> Hapus </a>
106 </td>
107 </tr>
108 <?php
109 // echo form close
110 echo form_close();
111 // end loop keranjang belanja
112 }
113 ?>
114
115 <tr class="table-row" style="font-weight: bold;">
116 <td colspan="4" class="column-1 text-white bg-dark">Total
117 Belanja</td>
118 <td colspan="2" class="column-2 text-white bg-dark">Rp.
119 <?php echo number_format($this->cart->total(), '0', ',', '.');
120 ?></td>
121 </tr>
122 <?php } else { ?>
123 <table class="table-shopping-cart">
124 <tr class="table-row" style="font-weight: bold;">
125 <td colspan="4" class="column-1 text-white bg-
126 dark">Total Belanja</td>
127 <td colspan="2" class="column-2 text-white bg-
128 dark">Rp. <?php echo number_format($this->session-
129 >flashdata('price'), '0', ',', '.'); ?></td>
130 </tr>
131 <?php } ?>
132 </table>
133 <br>
134
135 <!------- Snap Pembayaran ----->
136 <form id="payment-form" method="post"
137 action="=base_url()?&gt;/snap/finish"&gt;
138 &lt;input type="hidden" name="result_type"
139 id="result-type" value=""&gt;&lt;/div&gt;
140 &lt;input type="hidden" name="result_data"
141 id="result-data" value=""&gt;&lt;/div&gt;
142
143 &lt;?php echo form_open(base_url('belanja/checkout'));
144
145 if ($this-&gt;session-&gt;flashdata('type') == 'custom') { ?&gt;
146 &lt;input type="hidden" name="id_pelanggan" value="<?php
147 echo $pelanggan-&gt;id_pelanggan; ?&gt;"&gt;
148 &lt;input type="hidden" name="jumlah_transaksi"
149 value="<?php echo $this-&gt;session-&gt;flashdata('price') ?&gt;"&gt;
150 &lt;input type="hidden" name="id_custom" value="<?php
151 echo $this-&gt;session-&gt;flashdata('idProduk') ?&gt;"&gt;
152 &lt;input type="hidden" name="tanggal_transaksi"
153 value="<?php echo date('Y-m-d') ?&gt;"&gt;
154 &lt;?php
155 } else {
156 $kode_transaksi = date('dmY') .
157 strtoupper(random_string('alnum', 8));
158 ?&gt;
159 &lt;input type="hidden" name="id_pelanggan"
</pre

```

```

160 value="<?php echo $pelanggan->id_pelanggan; ?>">
161 <input type="hidden" name="jumlah_transaksi"
162 value="<?php echo $this->cart->total() ?>">
163 <input type="hidden" name="tanggal_transaksi"
164 value="<?php echo date('Y-m-d') ?>"> <?php } ?> <table
165 class="table">
166
167 <?php if($this->session->flashdata('type') == 'custom') {
168     $jmlbayar = $this->session-
169 >flashdata('price');
170 }else{
171     $jmlbayar = $this->cart->total();
172 } ?>
173
174 <thead>
175 <tr>
176 <th width="25%">Kode Transaksi</th>
177 <?php if ($this->session->flashdata('type') ==
178 'custom') { ?>
179 <th><input type="text" name="kode_transaksi"
180 class="form-control" id="kode_transaksi" value="<?php echo
181 $this->session->flashdata('kode') ?>" readonly required</th>
182 <?php } else { ?><th><input type="text"
183 name="kode_transaksi" class="form-control" id="kode_transaksi"
184 value="<?php echo $kode_transaksi ?>" readonly
185 required</th><?php } ?>
186 </tr>
187 </thead>
188 <tbody>
189 <tr>
190 <td>Nama Penerima</td>
191 <td><input type="text" name="nama_pelanggan"
192 id="nama_pelanggan" class="form-control" placeholder="Nama
193 Lengkap" value="<?php echo $pelanggan->nama_pelanggan ?>"
194 required</td>
195 </tr>
196 <tr>
197 <td>Email Penerima</td>
198 <td><input type="email" name="email" id="email"
199 class="form-control" placeholder="Email" value="<?php echo
200 $pelanggan->email ?>" required</td>
201 </tr>
202 <tr>
203 <td>Telepon</td>
204 <td><input type="text" name="telepon"
205 id="telepon" class="form-control" placeholder="Telepon"
206 value="<?php echo $pelanggan->telepon ?>" required</td>
207 </tr>
208 <tr>
209 <td>Alamat Pengiriman</td>
210
211 <td><textarea name="alamat" class="form-control" id="alamat"
212
213 placeholder="Alamat"><?php echo $pelanggan->alamat
214 ?></textarea></td>
215 </tr>
216 <tr>

```



```

217 <td></td>
218 <td class="btn-group">
219
220 <!-- inject forms from flash data -->
221 <input type="hidden" name="type" value="<?php echo
222 $this->session->flashdata('type'); ?>">
223 <input type="hidden" name="img_name" value="<?php
224 echo $this->session->flashdata('img_name'); ?>">
225 <input type="hidden" name="nama_produk"
226 value="<?php echo $this->session->flashdata('nama_produk'); ?>">
227 <input type="hidden" name="kode" value="<?php echo
228 $this->session->flashdata('kode'); ?>">
229 <input type="hidden" name="lingkar_dada"
230 value="<?php echo $this->session->flashdata('lingkar dada');
230 ?>">
231 <input type="hidden" name="lingkar_bahu"
232 value="<?php echo $this->session->flashdata('lingkar_bahu');
233 ?>">
234 <input type="hidden" name="lingkar_pinggang"
235 value="<?php echo $this->session->flashdata('lingkar_pinggang');
236 ?>">
237 <input type="hidden" name="panjang_tangan"
238 value="<?php echo $this->session->flashdata('panjang_tangan');
239 ?>">
240 <input type="hidden" name="panjang_baju"
241 value="<?php echo $this->session->flashdata('panjang_baju');
242 ?>">
243 <input type="hidden" name="busui" value="<?php
244 echo $this->session->flashdata('busui'); ?>">
245 <input type="hidden" name="keterangan"
246 value="<?php echo $this->session->flashdata('keterangan'); ?>">
247 <input type="hidden" name="harga_custom"
248 value="<?php echo $this->session->flashdata('harga_custom');
249 ?>">
250
251 <button class="flex-c-m size12 bg1 bo-rad-23 hov1 m-text3 trans-
252 0-4 mr-2" type="submit" data-amount="<?= $jmlbayar ?>" id="pay-
253 button"><i class="fa fa-credit-card m-r-5"></i> Buat
254 Pesanan</button>
255 <button class="flex-c-m size12 bg1 bo-rad-23 hov1 m-
256 text3 trans-0-4" type="reset">
257 <i class="fa fa-times m-r-5"></i> Reset
258 </button>
259 </td>
260 </tr>
261 </tbody>
262 </table>
263
264 <?php echo form_close(); ?>
265 </div>
266 </div>
267 </form>
268 <div class="flex-w flex-sb-m p-t-25 p-b-25 bo8 p-l-35 p-r-60
269 p-lr-15-sm">
270 <div class="flex-w flex-m w-full-sm">
271
272 </div>

```

```
273 </div>
274
275 </div>
276 </section>
277
278 <!-- Midtrans -->
279 <script>
280 $('#pay-button').click(function (event) {
281     event.preventDefault();
282     $(this).attr("disabled", "disabled");
283
284     var kode_transaksi = $("#kode_transaksi").val();
285     var nama_pelanggan = $("#nama_pelanggan").val();
286     var email = $("#email").val();
287     var telepon = $("#telepon").val();
288     //var alamat = $("#alamat").val();
289
290     var jumlah_transaksi = $(this).data('amount');
291     $.ajax({
292         url: '<?=@base_url()?>/snap/token',
293         data : {
294             kode_transaksi: kode_transaksi,
295             nama_pelanggan: nama_pelanggan,
296             email: email,
297             telepon: telepon,
298             //alamat: alamat,
299             jumlah_transaksi: jumlah_transaksi
300         },
301         cache: false,
302
303         success: function(data) {
304             //location = data;
305
306             console.log('token = '+data);
307
308             var resultType = document.getElementById('result-type');
309             var resultData = document.getElementById('result-data');
310
311             function changeResult(type,data){
312                 $("#result-type").val(type);
313                 $("#result-data").val(JSON.stringify(data));
314                 //resultType.innerHTML = type;
315                 //resultData.innerHTML = JSON.stringify(data);
316             }
317
318             snap.pay(data, {
319
320                 onSuccess: function(result){
321                     changeResult('success', result);
322                     console.log(result.status_message);
323                     console.log(result);
324                     $("#payment-form").submit();
325                 },
326                 onPending: function(result){
327                     changeResult('pending', result);
328                     console.log(result.status_message);
329                     $("#payment-form").submit();

```



```
230     },
231     onError: function(result){
232         changeResult('error', result);
233         console.log(result.status_message);
234         $("#payment-form").submit();
235     }
236 });
237 }
238 });
239 });
240 </script>
241 </body>
242 </html>
```



BAB 6 PENGUJIAN

Pada sub bab ini akan dipaparkan hasil pengujian pada *Website E-commerce* Butik Kebaya Iffah yang penulis kembangkan. Pengujian dilakukan dengan metode *Black Box Testing* untuk pengujian kebutuhan fungsional dan metode *Compatibility Testing* untuk pengujian kebutuhan Non-fungsional *Compatibility Browser*, dan pengujian *User Acceptance Testing (UAT)* untuk mengetahui persentase penerimaan sistem bagi pengguna terakhir.

6.1 Hasil Pengujian Menggunakan *Black Box Testing*

Dibawah ini terdapat beberapa tabel yang berfungsi untuk memaparkan hasil pengujian pada kebutuhan fungsional *Website E-commerce* Butik Kebaya Iffah menggunakan metode *Black Box Testing*.

Tabel 6. 1 Pengujian pada Proses Melakukan Registrasi

Nama	Melakukan Registrasi
Objek Uji	[A-01] F_KI_01
Skenario Pengujian	<p>Skenario 1: Aktor mengosongkan salah satu atau seluruh <i>field</i> pada <i>form</i> registrasi, lalu menekan “Registrasi”</p> <p>Skenario 2: Aktor mengisi <i>field</i> dengan data yang tidak sesuai dengan format yang dibutuhkan, lalu menekan “Registrasi”</p> <p>Skenario 3: Aktor mengisi Nama lengkap, <i>Email</i> yang sudah terdaftar sebagai pelanggan , <i>Password</i>, telepon dan alamat lalu menekan “Registrasi”</p> <p>Skenario 4: Aktor mengisi Nama lengkap, <i>Email</i> yang belum terdaftar sebagai pelanggan , <i>Password</i>, telepon dan alamat dengan format yang sesuai, lalu menekan “Registrasi”</p>
Hasil yang Diharapkan	<p>Skenario 1: Sistem akan menolak akses Registrasi dan menampilkan halaman Registrasi beserta <i>alert</i> untuk mengisi <i>field</i> yang kosong.</p>

	<p>Skenario 2: Sistem akan menolak akses Registrasi dan menampilkan halaman Registrasi beserta <i>alert</i> untuk mengisi kembali <i>field</i> dengan format yang sesuai.</p> <p>Skenario 3: Sistem akan menolak akses Registrasi dan mengembalikan <i>User</i> menuju halaman Registrasi beserta pesan untuk registrasi kembali dengan email yang belum terdaftar.</p> <p>Skenario 4: Sistem akan menerima akses Registrasi dan aktor memasuki halaman sukses registrasi.</p>
<p>Hasil yang Didapatkan</p>	<p>Skenario 1: Sistem berhasil menolak akses Registrasi dan menampilkan halaman Registrasi beserta <i>alert</i> untuk mengisi <i>field</i> yang kosong.</p> <p>Skenario 2: Sistem berhasil menolak akses Registrasi dan menampilkan halaman Registrasi beserta <i>alert</i> untuk mengisi kembali <i>field</i> dengan format yang sesuai.</p> <p>Skenario 3: Sistem berhasil menolak akses Registrasi dan mengembalikan <i>User</i> menuju halaman Registrasi beserta pesan untuk registrasi kembali dengan email yang belum terdaftar.</p> <p>Skenario 4: Sistem berhasil menerima akses Registrasi dan aktor memasuki halaman sukses registrasi.</p>
<p>Kesimpulan</p>	<p>Diterima.</p>

Sumber : (Penulis, 2021)

Tabel 6. 2 Pengujian pada Proses Login

<p>Nama</p>	<p><i>Login</i></p>
<p>Objek Uji</p>	<p>[A-01] F_KI_02</p>
<p>Skenario Pengujian</p>	<p>Skenario 1: Aktor mengosongkan salah satu atau kedua <i>field Email</i> dan <i>Password</i>, lalu menekan tombol “Log In”.</p> <p>Skenario 2: Aktor mengisi <i>field Email</i> dengan format yang tidak sesuai, dan password lalu menekan tombol “Log In”.</p> <p>Skenario 3: Aktor mengisi <i>Email</i> dan <i>Password</i> dengan data kombinasi yang salah, lalu menekan tombol “Log In”.</p> <p>Skenario 4: Aktor mengisi kedua <i>field Email</i> dan <i>Password</i> dengan data yang benar, lalu menekan tombol “Log In”.</p>
<p>Hasil yang Diharapkan</p>	<p>Skenario 1: Sistem akan menolak akses <i>Login</i> dan menampilkan halaman <i>Login</i> beserta <i>alert</i> untuk mengisi <i>field</i> yang kosong.</p> <p>Skenario 2: Sistem akan menolak akses <i>Login</i> dan menampilkan halaman <i>Login</i> beserta <i>alert</i> untuk mengisi kembali <i>field Email</i> dengan format yang sesuai.</p> <p>Skenario 3: Sistem akan menolak akses <i>Login</i> dan mengembalikan <i>User</i> menuju halaman <i>Login</i> beserta pesan kesalahan.</p> <p>Skenario 4: Sistem akan menerima akses <i>login</i> dan aktor memasuki halaman <i>Dashboard</i> Pelanggan.</p>
<p>Hasil yang Didapatkan</p>	<p>Skenario 1: Sistem berhasil menolak akses <i>Login</i> dan menampilkan halaman <i>Login</i> beserta <i>alert</i> untuk mengisi <i>field</i> yang kosong.</p> <p>Skenario 2: Sistem berhasil menolak akses <i>Login</i> dan menampilkan halaman <i>Login</i> beserta <i>alert</i> untuk mengisi kembali <i>field Email</i> dengan format yang sesuai.</p> <p>Skenario 3: Sistem berhasil menolak akses <i>Login</i> dan mengembalikan aktor menuju halaman <i>Login</i> beserta pesan kesalahan.</p> <p>Skenario 4: Sistem berhasil menerima akses <i>login</i> dan memasuki</p>

	halaman <i>Dashboard</i> Pelanggan.
Kesimpulan	Diterima.

Sumber : (Penulis, 2021)

Tabel 6. 3 Pengujian pada Proses Logout

Nama	<i>Logout</i>
Objek Uji	[A-01] F_KI_03
Skenario Pengujian	Aktor menekan tombol “Logout”
Hasil yang Diharapkan	Sistem akan menghapus <i>session</i> aktor dan mengembalikan <i>User</i> menuju halaman <i>Login</i> .
Hasil yang Didapatkan	Sistem berhasil menghapus <i>session</i> aktor dan mengembalikan <i>User</i> menuju halaman <i>Login</i> .
Kesimpulan	Diterima.

Sumber : (Penulis, 2021)

Tabel 6. 4 Pengujian pada Proses Melihat daftar produk

Nama	Melihat Daftar Produk
Objek Uji	[A-01] F_KI_04
Skenario Pengujian	Skenario 1: Aktor memilih menu “Shop” Skenario 2: Aktor memilih menu “Shop” dan memilih kategori produk
Hasil yang Diharapkan	Skenario 1: Sistem akan menampilkan seluruh daftar produk Butik Kebaya Iffah Skenario 2: Sistem akan menampilkan daftar produk Butik Kebaya Iffah sesuai dengan kategori yang dipilih
Hasil yang Didapatkan	Skenario 1: Sistem berhasil menampilkan seluruh daftar produk Butik Kebaya Iffah Skenario 2:

	Sistem berhasil menampilkan daftar produk Butik Kebaya Iffah sesuai dengan kategori yang dipilih
Kesimpulan	Diterima

Sumber : (Penulis, 2021)

Tabel 6. 5 Pengujian pada Proses Mengelola Keranjang Belanja

Nama	Mengelola Keranjang Belanja
Objek Uji	[A-01] F_KI_05
Skenario Pengujian	<p>Skenario 1: Aktor menambahkan produk , dengan menekan tombol “add to cart”</p> <p>Skenario 2: Aktor melihat keranjang belanja namun belum terdapat produk</p> <p>Skenario 3: Aktor melihat keranjang belanja dan terdapat produk</p> <p>Skenario 4: Aktor menambahkan jumlah item produk, lalu menekan tombol “Update”</p> <p>Skenario 5: Aktor mengurangi jumlah item produk, lalu menekan tombol “Update”</p> <p>Skenario 6: Aktor menghapus salah satu produk pada keranjang, dengan menekan tombol “Delete”</p> <p>Skenario 7: Aktor menghapus seluruh produk pada keranjang, dengan menekan tombol “Kosongkan Keranjang”</p>
Hasil yang Diharapkan	<p>Skenario 1: Sistem akan menambahkan produk ke keranjang belanja, dan menampilkan <i>pop up</i> berhasil</p> <p>Skenario 2: Sistem menampilkan teks “Keranjang Belanja Kosong”</p> <p>Skenario 3: Sistem akan menampilkan data produk pada keranjang belanja</p> <p>Skenario 4: Sistem akan menambahkan jumlah item produk dan</p>

	<p>menghitung kembali subtotal pada produk tersebut</p> <p>Skenario 5: Sistem akan mengurangi jumlah item produk dan menghitung kembali subtotal pada produk tersebut</p> <p>Skenario 6: Sistem akan menghapus baris data produk yang terpilih dari keranjang belanja</p> <p>Skenario 7: Sistem akan mengosongkan keranjang belanja</p>
Hasil yang Didapatkan	<p>Skenario 1: Sistem berhasil menambahkan produk ke keranjang belanja, dan menampilkan <i>pop up</i> berhasil</p> <p>Skenario 2: Sistem berhasil menampilkan teks “Keranjang Belanja Kosong”</p> <p>Skenario 3: Sistem berhasil menampilkan data produk pada keranjang belanja</p> <p>Skenario 4: Sistem berhasil menambahkan jumlah item produk dan menghitung kembali subtotal pada produk tersebut</p> <p>Skenario 5: Sistem berhasil mengurangi jumlah item produk dan menghitung kembali subtotal pada produk tersebut</p> <p>Skenario 6: Sistem berhasil menghapus baris data produk yang terpilih dari keranjang belanja</p> <p>Skenario 7: Sistem berhasil mengosongkan keranjang belanja</p>
Kesimpulan	Diterima.

Sumber : (Penulis, 2021)

Tabel 6. 6 Pengujian pada Proses Melakukan Custom Order

Nama	Melakukan <i>Custom Order</i>
Objek Uji	[A-01] F_KI_06
Skenario Pengujian	<p>Skenario 1: Aktor memilih menu “Custom”</p> <p>Skenario 2: Aktor membuka form <i>custom order</i>, dengan menekan tombol “custom size”</p>

	<p>Skenario 3: Aktor mengosongkan salah satu atau seluruh <i>field</i> pada form custom order</p> <p>Skenario 4: Aktor mengupload gambar selain format yang diperbolehkan(gif,jpg,png,jpeg) dan melebihi dari ukuran file maximal</p> <p>Skenario 5: Aktor mengisi seluruh <i>field</i> form custom order dengan format data yang sesuai, dan Aktor mengupload gambar dengan format yang diperbolehkan(gif,jpg,png,jpeg) dan kurang dari ukuran file maximal namun aktor belum melakukan <i>login</i></p> <p>Skenario 6: Aktor sudah login, lalu mengisi seluruh <i>field</i> form custom order dengan format data yang sesuai, dan Aktor mengupload gambar dengan format yang diperbolehkan(gif,jpg,png,jpeg) dan kurang dari ukuran file maximal</p>
<p>Hasil yang Diharapkan</p>	<p>Skenario 1: Sistem akan menampilkan daftar produk Butik Kebaya Iffah yang dapat dilakukan <i>Custom Size</i></p> <p>Skenario 2: Sistem akan menampilkan <i>form custom size</i></p> <p>Skenario 3: Sistem akan menolak masukan data dan menampilkan <i>form custom size</i> beserta <i>alert</i> untuk mengisi <i>field</i> yang kosong.</p> <p>Skenario 4: Sistem akan menolak masukan data dari aktor, dan mengembalikan pada halaman <i>form custom size</i></p> <p>Skenario 5: Sistem akan menolak masukan data dan mengembalikan User dengan menampilkan halaman <i>Login</i></p> <p>Skenario 6: Sistem akan menerima masukan data dari aktor, dan memasuki halaman <i>Checkout Custom</i> beserta menampilkan harga <i>custom order</i> sesuai perhitungan data masukan</p>
<p>Hasil yang Didapatkan</p>	<p>Skenario 1: Sistem berhasil menampilkan daftar produk Butik Kebaya Iffah yang dapat dilakukan <i>Custom Size</i></p> <p>Skenario 2: Sistem berhasil menampilkan <i>form custom size</i></p> <p>Skenario 3: Sistem akan menolak masukan data dan menampilkan <i>form</i></p>

	<p><i>custom size</i> beserta <i>alert</i> untuk mengisi <i>field</i> yang kosong.</p> <p>Skenario 4: Sistem akan menolak masukan data dari aktor, dan mengembalikan pada halaman <i>form custom size</i></p> <p>Skenario 5: Sistem berhasil menolak masukan data dan mengembalikan User dengan menampilkan halaman <i>Login</i></p> <p>Skenario 6: Sistem berhasil menerima masukan data dari aktor, dan memasuki halaman <i>Checkout Custom</i> beserta menampilkan harga <i>custom order</i> sesuai perhitungan data masukan</p>
Kesimpulan	Diterima.

Sumber : (Penulis, 2021)

Tabel 6. 7 Pengujian pada Proses Melakukan Checkout

Nama	Melakukan <i>Checkout</i>
Objek Uji	[A-01] F_KI_07
Skenario Pengujian	<p>Skenario 1: Aktor belum login, dan menekan tombol “Checkout” pada halaman keranjang belanja untuk proses checkout produk ready</p> <p>Skenario 2: Aktor sudah login, dan menekan tombol “Checkout” pada halaman keranjang belanja untuk proses checkout produk ready</p> <p>Skenario 3: Aktor mengosongkan salah satu atau seluruh <i>field</i> pada <i>form Checkout</i> pada halaman <i>checkout</i></p> <p>Skenario 4: Aktor mengisi <i>field</i> dengan data yang tidak sesuai dengan format yang dibutuhkan</p> <p>Skenario 5: Aktor sudah login, dan mengisi seluruh <i>field form checkout</i> dengan format data yang sesuai, lalu menekan tombol “Buat Pesanan”</p>

<p>Hasil yang Diharapkan</p>	<p>Skenario 1: Sistem akan mengembalikan aktor dan menampilkan halaman <i>Login</i></p> <p>Skenario 2: Sistem akan menampilkan halaman checkout untuk proses checkout produk ready</p> <p>Skenario 3: Sistem akan menolak proses <i>checkout</i> dan menampilkan <i>form checkout</i> beserta <i>alert</i> untuk mengisi <i>field</i> yang kosong</p> <p>Skenario 4: Sistem akan menolak proses <i>checkout</i> dan menampilkan <i>form checkout</i> beserta <i>alert</i> untuk mengisi <i>field</i> dengan format yang sesuai</p> <p>Skenario 5: Sistem akan memverifikasi total pesanan, dan proses akan dilanjutkan dengan menampilkan halaman pembayaran midtrans</p>
<p>Hasil yang Didapatkan</p>	<p>Skenario 1: Sistem berhasil mengembalikan aktor dan menampilkan halaman <i>Login</i></p> <p>Skenario 2: Sistem berhasil menampilkan halaman checkout untuk proses checkout produk ready</p> <p>Skenario 3: Sistem berhasil menolak proses <i>checkout</i> dan menampilkan <i>form checkout</i> beserta <i>alert</i> untuk mengisi <i>field</i> yang kosong</p> <p>Skenario 4: Sistem berhasil menolak proses <i>checkout</i> dan menampilkan <i>form checkout</i> beserta <i>alert</i> untuk mengisi <i>field</i> dengan format yang sesuai</p> <p>Skenario 5: Sistem berhasil memverifikasi total pesanan, dan proses akan dilanjutkan dengan menampilkan halaman pembayaran midtrans</p>
<p>Kesimpulan</p>	<p>Diterima.</p>

Sumber : (Penulis, 2021)

Tabel 6. 8 Pengujian pada Proses Melakukan Pembayaran

Nama	Melakukan Pembayaran
Objek Uji	[A-01] F_KI_08, [A-03] F_KI_01
Skenario Pengujian	<p>Skenario 1: Aktor telah menekan tombol “Buat Pesanan”</p> <p>Skenario 2: Aktor menekan Lanjut, kemudian aktor dapat memilih metode pembayaran, lalu lihat nomor rekening dan terakhir, menekan ‘mohon selesaikan pembayaran’</p>
Hasil yang Diharapkan	<p>Skenario 1: Sistem akan menampilkan SNAP Payment midtrans</p> <p>Skenario 2: Sistem akan menampilkan lanjutan dari halaman rincian yaitu dilanjutkan tampil cara pembayaran, kemudian tampil nomor rekening dan batas pembayaran, dan setelah proses terlewati, akan ditampilkan halaman sukses</p>
Hasil yang Didapatkan	<p>Skenario 1: Sistem berhasil menampilkan SNAP Payment midtrans</p> <p>Skenario 2: Sistem berhasil menampilkan lanjutan dari halaman rincian yaitu dilanjutkan tampil cara pembayaran, kemudian tampil nomor rekening dan batas pembayaran, dan setelah proses terlewati, berhasil ditampilkan halaman sukses</p>
Kesimpulan	Diterima.

Sumber : (Penulis, 2021)

Tabel 6. 9 Pengujian pada Proses Melihat Riwayat Transaksi

Nama	Melihat Riwayat Transaksi
Objek Uji	[A-01] F_KI_09
Skenario Pengujian	Aktor memilih menu “Riwayat Belanja” pada <i>dashboard</i> pelanggan
Hasil yang Diharapkan	Sistem akan menampilkan riwayat belanja pembelian produk oleh pelanggan

Hasil yang Didapatkan	Sistem berhasil menampilkan riwayat belanja pembelian produk oleh pelanggan
Kesimpulan	Diterima.

Sumber : (Penulis, 2021)

Tabel 6. 10 Pengujian pada Proses Login Admin

Nama	<i>Login admin</i>
Objek Uji	[A-02] F_KI_01
Skenario Pengujian	<p>Skenario 1: Aktor mengosongkan salah satu atau kedua <i>field Username</i> dan <i>Password</i>, lalu menekan tombol "Login"</p> <p>Skenario 2: Aktor mengisi <i>Username</i> dan <i>Password</i> dengan data kombinasi yang salah, lalu menekan tombol "Login"</p> <p>Skenario 3: Aktor mengisi kedua <i>field Username</i> dan <i>Password</i> dengan data yang benar, lalu menekan tombol "Login"</p>
Hasil yang Diharapkan	<p>Skenario 1: Sistem akan menolak akses <i>Login</i> dan menampilkan halaman <i>Login admin</i> beserta <i>alert</i> untuk mengisi <i>field</i> yang kosong</p> <p>Skenario 2: Sistem akan menolak akses <i>Login</i> dan mengembalikan <i>User</i> menuju halaman <i>Login</i> beserta pesan kesalahan</p> <p>Skenario 3: Sistem akan menerima akses <i>login</i> dan aktor memasuki halaman <i>Dashboard Admin</i></p>
Hasil yang Didapatkan	<p>Skenario 1: Sistem berhasil menolak akses <i>Login</i> dan menampilkan halaman <i>Login admin</i> beserta <i>alert</i> untuk mengisi <i>field</i> yang kosong</p> <p>Skenario 2: Sistem berhasil menolak akses <i>Login</i> dan mengembalikan <i>User</i> menuju halaman <i>Login</i> beserta pesan kesalahan</p> <p>Skenario 3: Sistem berhasil menerima akses <i>login</i> dan memasuki halaman <i>Dashboard Admin</i></p>

Kesimpulan	Diterima
------------	----------

Sumber : (Penulis, 2021)

Tabel 6. 11 Pengujian pada Proses Logout Admin

Nama	Logout
Objek Uji	[A-02] F_KI_02
Skenario Pengujian	Aktor menekan tombol "Logout"
Hasil yang Diharapkan	Sistem akan menghapus <i>session</i> aktor dan mengembalikan <i>User</i> menuju halaman <i>Login admin</i>
Hasil yang Didapatkan	Sistem berhasil menghapus <i>session</i> aktor dan mengembalikan <i>User</i> menuju halaman <i>Login admin</i>
Kesimpulan	Diterima

Sumber : (Penulis, 2021)

Tabel 6. 12 Pengujian pada Proses Melihat Dashboard

Nama	Melihat Dashboard
Objek Uji	[A-02] F_KI_03
Skenario Pengujian	Aktor memilih menu "Dashboard"
Hasil yang Diharapkan	Sistem akan menampilkan halaman dashboard yang berisi informasi ringkasan sistem
Hasil yang Didapatkan	Sistem berhasil menampilkan halaman dashboard yang berisi informasi ringkasan sistem
Kesimpulan	Diterima

Sumber : (Penulis, 2021)

Tabel 6. 13 Pengujian pada Proses Mengelola Produk

<p>Nama</p>	<p>Mengelola Produk</p>
<p>Objek Uji</p>	<p>[A-02] F_KI_04</p>
<p>Skenario Pengujian</p>	<p>Skenario 1: Aktor memilih menu “Produk”</p> <p>Skenario 2: Aktor menekan tombol “Tambah”, pada halaman produk</p> <p>Skenario 3: Aktor mengosongkan salah satu atau seluruh <i>field</i> pada <i>form</i> tambah produk</p> <p>Skenario 4: Aktor mengisi <i>field</i> kode produk dengan data yang sudah terdaftar</p> <p>Skenario 5: Aktor mengupload gambar selain format yang diperbolehkan(gif,jpg,png,jpeg) dan melebihi dari ukuran file maximal</p> <p>Skenario 6: Aktor mengisi seluruh <i>field</i> form tambah produk dengan kode produk yang belum terdaftar, dan Aktor mengupload gambar dengan format yang diperbolehkan (gif,jpg,png,jpeg) dan kurang dari ukuran file maximal</p> <p>Skenario 7: Aktor menekan tombol “Edit” untuk melakukan perubahan data produk pada form edit data produk</p> <p>Skenario 8: Aktor menekan tombol “Delete” untuk menghapus data produk</p>
<p>Hasil yang Diharapkan</p>	<p>Skenario 1: Sistem akan menampilkan seluruh daftar produk</p> <p>Skenario 2: Sistem akan menampilkan <i>form</i> tambah produk</p> <p>Skenario 3: Sistem akan menolak masukan data dan menampilkan <i>form</i> tambah produk beserta <i>alert</i> untuk mengisi <i>field</i> yang kosong</p> <p>Skenario 4: Sistem akan menolak masukan data dan mengembalikan <i>User</i> menuju halaman <i>form</i> tambah produk beserta pesan untuk mengisi kembali <i>field</i> dengan kode produk yang belum terdaftar</p>

	<p>Skenario 5: Sistem akan menolak masukan data dan mengembalikan <i>User</i> menuju halaman <i>form</i> tambah produk beserta pesan kesalahan upload gambar</p> <p>Skenario 6: Sistem akan menerima masukan data dari aktor, dan menampilkan produk pada daftar.</p> <p>Skenario 7: Sistem akan menerima perubahan data produk yang diubah aktor, dan memperbaruinya pada daftar</p> <p>Skenario 8: Sistem akan menghapus data produk terpilih dari daftar produk</p>
<p>Hasil yang Didapatkan</p>	<p>Skenario 1: Sistem berhasil menampilkan daftar produk yang sudah diinputkan sebelumnya.</p> <p>Skenario 2: Sistem berhasil menampilkan <i>form</i> tambah produk</p> <p>Skenario 3: Sistem berhasil menolak masukan data dan menampilkan <i>form</i> tambah produk beserta <i>alert</i> untuk mengisi <i>field</i> yang kosong</p> <p>Skenario 4: Sistem berhasil menolak masukan data dan mengembalikan <i>User</i> menuju halaman <i>form</i> tambah produk beserta pesan untuk mengisi kembali <i>field</i> dengan kode produk yang belum terdaftar</p> <p>Skenario 5: Sistem berhasil menolak masukan data dan mengembalikan <i>User</i> menuju halaman <i>form</i> tambah produk beserta pesan kesalahan upload gambar</p> <p>Skenario 6: Sistem berhasil menerima masukan data dari aktor, dan menampilkan produk pada daftar</p> <p>Skenario 7: Sistem berhasil menerima perubahan data produk yang diubah aktor, dan memperbaruinya pada daftar</p> <p>Skenario 8: Sistem berhasil menghapus data produk terpilih dari daftar produk</p>
<p>Kesimpulan</p>	<p>Diterima.</p>

Sumber : (Penulis, 2021)

Tabel 6. 14 Pengujian pada Proses Melihat Data Pesanan

Nama	Melihat Data Pesanan
Objek Uji	[A-02] F_KI_05
Skenario Pengujian	Aktor memilih menu “Laporan Transaksi”
Hasil yang Diharapkan	Sistem akan menampilkan daftar transaksi pembelian produk ready
Hasil yang Didapatkan	Sistem berhasil menampilkan daftar transaksi pembelian produk ready
Kesimpulan	Diterima.

Sumber : (Penulis, 2021)

Tabel 6. 15 Pengujian pada Proses Melihat Data Custom Order

Nama	Melihat Data Custom Order
Objek Uji	[A-02] F_KI_04
Skenario Pengujian	Aktor memilih menu “Laporan Custom”
Hasil yang Diharapkan	Sistem akan menampilkan daftar transaksi pembelian produk custom
Hasil yang Didapatkan	Sistem berhasil menampilkan daftar transaksi pembelian produk custom
Kesimpulan	Diterima.

Sumber : (Penulis, 2021)

Tabel 6. 16 Pengujian pada Proses Update Status Bayar

Nama	Update Status Bayar
Objek Uji	[A-02] F_KI_07, [A-03] F_KI_01

Skenario Pengujian	<p>Skenario 1: Aktor mengosongkan <i>field</i> cek status, lalu menekan tombol ‘cari’</p> <p>Skenario 2: Aktor memasukkan kode transaksi yang tidak terdaftar pada <i>field</i> cek status, lalu menekan tombol ‘cari’</p> <p>Skenario 3: Aktor memasukkan kode transaksi yang terdaftar pada <i>field</i> cek status, lalu menekan tombol ‘cari’</p>
Hasil yang Diharapkan	<p>Skenario 1-2: Sistem akan menampilkan pesan “Order id yang anda cari tidak ada”</p> <p>Skenario 3: Sistem akan mengganti dengan status terbaru, dan menampilkan warna sesuai status bayar, merah untuk <i>failure</i>, kuning untuk <i>pending</i>, dan hijau untuk <i>settlement</i></p>
Hasil yang Didapatkan	<p>Skenario 1-2: Sistem berhasil menampilkan pesan “Order id yang anda cari tidak ada”</p> <p>Skenario 3: Sistem berhasil mengganti dengan status terbaru, dan menampilkan warna sesuai status bayar, merah untuk <i>failure</i>, kuning untuk <i>pending</i>, dan hijau untuk <i>settlement</i></p>
Kesimpulan	Diterima.

Sumber : (Penulis, 2021)

Tabel 6. 17 Pengujian pada Proses Mengelola Admin

Nama	Mengelola Admin
Objek Uji	[A-02] F_KI_08
Skenario Pengujian	<p>Skenario 1: Aktor memilih menu “Admin”</p> <p>Skenario 2: Aktor menekan tombol “Tambah”, pada halaman admin</p> <p>Skenario 3: Aktor mengosongkan salah satu atau seluruh <i>field</i> pada form tambah admin</p> <p>Skenario 4: Aktor mengisi <i>field</i> dengan data yang tidak sesuai dengan</p>

	<p>format yang dibutuhkan pada <i>form</i> tambah admin</p> <p>Skenario 5: Aktor mengisi seluruh <i>field</i> form tambah admin dengan format data yang sesuai</p> <p>Skenario 6: Aktor menekan tombol “Edit” untuk melakukan perubahan data admin pada form edit data admin</p> <p>Skenario 7: Aktor menekan tombol “Delete” untuk menghapus data admin</p>
<p>Hasil yang Diharapkan</p>	<p>Skenario 1: Sistem akan menampilkan daftar admin yang sudah diinputkan sebelumnya</p> <p>Skenario 2: Sistem akan menampilkan <i>form</i> tambah admin</p> <p>Skenario 3: Sistem akan menolak masukan data dan menampilkan <i>form</i> tambah admin beserta <i>alert</i> untuk mengisi <i>field</i> yang kosong</p> <p>Skenario 4: Sistem akan menolak masukan data dan menampilkan <i>form</i> tambah admin beserta <i>alert</i> untuk mengisi kembali <i>field Email</i> dengan format yang sesuai</p> <p>Skenario 5: Sistem akan menerima masukan data dari aktor, dan menampilkan admin pada daftar</p> <p>Skenario 6: Sistem akan menerima perubahan data admin yang diubah aktor, dan memperbaruinya pada daftar</p> <p>Skenario 7: Sistem akan menghapus data admin terpilih dari daftar</p>
<p>Hasil yang Didapatkan</p>	<p>Skenario 1: Sistem berhasil menampilkan daftar admin yang sudah diinputkan sebelumnya.</p> <p>Skenario 2: Sistem berhasil menampilkan <i>form</i> tambah admin</p> <p>Skenario 3: Sistem berhasil menolak masukan data dan menampilkan <i>form</i> tambah admin beserta <i>alert</i> untuk mengisi <i>field</i> yang kosong</p> <p>Skenario 4: Sistem berhasil menolak masukan data dan menampilkan <i>form</i> tambah admin beserta <i>alert</i> untuk mengisi kembali <i>field Email</i> dengan format yang sesuai</p>

	<p>Skenario 5: Sistem berhasil menerima masukan data dari aktor, dan menampilkan admin pada daftar.</p> <p>Skenario 6: Sistem berhasil menerima perubahan data admin yang diubah aktor, dan memperbaruinya pada daftar</p> <p>Skenario 7: Sistem berhasil menghapus data admin terpilih dari daftar</p>
Kesimpulan	Diterima

Sumber : (Penulis, 2021)

Tabel 6. 18 Pengujian pada Proses Mencetak Detail Pesanan

Nama	Mencetak Detail Pesanan
Objek Uji	[A-02] F_KI_09
Skenario Pengujian	Aktor membuka detail transaksi, lalu menekan tombol “cetak”
Hasil yang Diharapkan	Sistem akan menampilkan halaman unduh file cetak detail pesanan
Hasil yang Didapatkan	Sistem berhasil menampilkan halaman unduh file cetak detail pesanan
Kesimpulan	Diterima

Sumber : (Penulis, 2021)

Berdasarkan hasil pengujian menggunakan metode black box untuk masing-masing fungsi, persentase hasil pengujian dapat diperoleh dengan membagi jumlah pengujian yang mencapai status valid dengan jumlah total pengujian yang dilakukan. Hasil uji black box dihitung sebagai berikut :

$$\text{Presentase} = (\text{Total Kasus Uji Berstatus Valid} / \text{Total Kasus Uji}) \times 100\%$$

$$\text{Presentase} = (18 / 18) \times 100\%$$

$$\text{Presentase} = 100\% \text{ (Valid)}$$

Dari hasil perhitungan tersebut dapat disimpulkan bahwa *Website e-commerce* Butik Kebaya Iffah dapat beroperasi dengan baik dan telah memenuhi kebutuhan sistem yang telah ditentukan sebelumnya, karena hasil pengujian menunjukkan persentase 100% Valid.

6.2 Hasil Pengujian Menggunakan Browser Compatibility Testing

Compatibility Testing dilakukan dengan menggunakan perangkat lunak SortSite. Pengujian ini bertujuan untuk memastikan bahwa website e-commerce butik kebaya iffah ini dapat berjalan di beberapa browser yang berbeda dengan baik. Browser yang digunakan ditunjukkan pada Tabel 6.19

Tabel 6. 19 Browser Compatibility Testing

No.	Nama Browser	Versi Browser
1	Internet Explorer	11
2	Edge	88
3	Firefox	84
4	Safari	14
5	Opera	73
6	Chrome	88
7	iOS	<= 13, dan 14
8	Android	88

Pada pengujian *Compatibility Testing* ini menggunakan perangkat lunak SortSite yang membagi masalah menjadi 3 jenis yaitu *critical issues*, *major issues* dan *minor issues*. Dari pengujian yang dilakukan dapat diketahui bahwa tidak terdapat *critical issues*, kesalahan mayor, maupun kesalahan minor. Dikarenakan pada sistem sudah mendukung penggunaan tampilan sistem seperti format CSS, HTML, dan Bootstrap. Hasil dari pengujian *compatibility testing* menyatakan bahwa sistem mampu berjalan dengan baik sesuai dengan prosedur yang telah ditentukan. Berikut adalah hasil pengujian *Compatibility* pada gambar 6.1.

This tab shows pages that exhibit browser-specific behavior, or trigger browser bugs.

Browser	IE	Edge	Firefox	Safari	Opera	Chrome	iOS	Android	
Version	11	88	84	14	73	88	≤ 13 14	88	
Critical Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓
Major Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓
Minor Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓

Key

- Missing content or functionality
- Major layout or performance problems
- Minor layout or performance problems

Gambar 6. 1 Hasil Compatibility Testing

Sumber : (Penulis,2021)

6.3 User Acceptance Test

Pada tahap pengujian ini akan dilakukan Uji Penerimaan Pengguna yang dilakukan oleh pengguna sistem untuk memastikan bahwa fungsi dalam sistem bekerja dengan baik dan memenuhi kebutuhan pengguna. Pengujian ini melibatkan 1 Admin, dan 3 Pelanggan. Pengujian dilakukan dengan cara mempresentasikan terlebih dahulu kerja sistem kepada pengguna sistem, kemudian pengguna menjawab beberapa pertanyaan berupa Kuesioner berskala likert.

6.3.1 Pengujian oleh Admin

Pada bagian ini akan dijelaskan mengenai hasil dari kuisisioner pengujian yang dibagikan kepada owner butik yang berperan sebagai admin, beserta membahas hasil perhitungannya. Hasil kuisisioner pengujian UAT untuk admin dapat dilihat pada Tabel 6.20.

Tabel 6. 20 Hasil Kuisisioner UAT User Admin

No.	Pertanyaan	Jawaban				
		SS	S	N	TS	STS
1.	Apakah desain tampilan sistem menarik dan sesuai?	1				
2.	Apakah dengan dibuatnya sistem ini mempercepat dan mempermudah proses pengelolaan daftar produk?		1			
3.	Form untuk menginputkan data produk, apakah sudah lengkap seperti pencatatan manual?		1			
4.	Apakah proses pengelolaan transaksi pembelian produk dan pembayaran lebih mudah, aman dan cepat?	1				
5.	Dengan adanya fitur update status, apakah dapat berjalan aman, dan meminimalisir penipuan?	1				
6.	Apakah proses bisnis transaksi pembelian produk custom dapat berjalan baik, dengan adanya sistem ini?			1		
7.	Apakah penggunaan fitur fitur pada sistem mudah dipahami?	1				
8.	Apakah terdapat kendala saat anda menjalankan sistem ini?	1				
9.	Apakah sistem yang dibuat pada aplikasi sudah sesuai dengan apa yang diharapkan?		1			

Total Jawaban Responden	5	3	1		
--------------------------------	---	---	---	--	--

Sumber : (Penulis, 2021)

Dari hasil kuisisioner dengan responden admin, akan dilakukan perhitungan menggunakan skala likert dengan cara mengalikan jumlah jawaban responden dengan bobot nilai jawaban responden, yang mengacu pada persamaan 2.1 yang dapat dilihat pada Tabel 6.21.

Tabel 6. 21 Tabel Perhitungan Skala Likert User Admin

Jawaban	Bobot Nilai	Jumlah Jawaban Responden	Hasil Total Skor Responden
SS	5	5	25
S	4	3	12
K	3	1	3
TS	2	0	0
STS	1	0	0
Total Nilai			40

Kemudian setelah dilakukan perhitungan total nilai, dilakukan perhitungan nilai Y sebagai total tertinggi keseluruhan kuisisioner jumlah skala (N1), jumlah responden (n), dan jumlah pertanyaan (U). Mengacu pada persamaan 2.2, Berikut perhitungan nilai Y :

$$Y = N1 \times n \times U$$

$$Y = 9 \times 1 \times 5 = 45$$

Setelah diketahui nilai Y, kemudian dilakukan perhitungan presentase akhir pengujian UAT mengacu persamaan 2.3.

$$\text{Rumus Index} = (\text{Total Nilai} / Y) \times 100\%$$

$$= (40 / 45) \times 100\%$$

$$= 88,9 \%$$

Berdasarkan kualifikasi interpretasi skor berdasarkan interval pengujian pada user admin didapatkan nilai 88,9 % yang termasuk kategori sangat setuju, Sehingga dapat disimpulkan bahwa pendapat user admin sistem yang dikembangkan telah dapat diterima dengan hasil sangat setuju.

6.3.2 Pengujian oleh Pelanggan

Pada bagian ini akan dijelaskan mengenai hasil dari kuisioner pengujian yang dibagikan kepada pelanggan Butik Kebaya Iffah dimana kuesioner dibagikan kepada 3 Pelanggan, beserta membahas hasil perhitungannya. Hasil kuesioner pengujian UAT untuk admin dapat dilihat pada Tabel 6.22.

Tabel 6. 22 Hasil Kuesioner UAT User Pelanggan

No.	Pertanyaan	Jawaban				
		SS	S	N	TS	STS
1.	Apakah desain dan layout tampilan sistem menarik, sesuai serta mudah dipahami?	1	2			
2.	Apakah penggunaan fitur fitur pada sistem mudah dipelajari ?	2	1			
3.	Apakah dengan dibuatnya sistem ini mempercepat transaksi pembelian anda?	3				
4.	Dengan adanya fitur custom order memudahkan keinginan pelanggan, dan dapat berjalan dengan baik ?	1	1	1		
5.	Apakah proses pembayaran lebih efisien, aman dan cepat?	3				
6.	Apakah tidak terdapat kendala saat anda menjalankan sistem ini?	2		1		
7.	Apakah sistem ini sangat bermanfaat?	3				
8.	Apakah sistem yang dibuat pada aplikasi sudah sesuai dengan apa yang diharapkan?	1	2			
Total Jawaban Responden		16	6	2		

Sumber : (Penulis, 2021)

Dari hasil kuisioner dengan responden pelanggan, akan dilakukan perhitungan menggunakan skala likert dengan cara mengalikan jumlah jawaban responden dengan bobot nilai jawaban responden, yang mengacu pada persamaan 2.1 yang dapat dilihat pada Tabel 6.22.

Tabel 6. 23 Tabel Perhitungan Skala Likert User Pelanggan

Jawaban	Bobot Nilai	Jumlah Jawaban Responden	Hasil Total Skor Responden
SS	5	16	80
S	4	6	24
K	3	2	6
TS	2	0	0

STS	1	0	0
Total Nilai			110

Kemudian setelah dilakukan perhitungan total nilai, dilakukan perhitungan nilai Y sebagai total tertinggi keseluruhan kuesioner jumlah skala (N1), jumlah responden (n), dan jumlah pertanyaan (U). Mengacu pada persamaan 2.2, Berikut perhitungan nilai Y :

$$Y = N1 \times n \times U$$

$$Y = 8 \times 3 \times 5 = 120$$

Setelah diketahui nilai Y, kemudian dilakukan perhitungan presentase akhir pengujian UAT mengacu persamaan 2.3.

$$\text{Rumus Index} = (\text{Total Nilai} / Y) \times 100\%$$

$$= (110 / 120) \times 100\%$$

$$= 91,7\%$$

Berdasarkan kualifikasi interpretasi skor berdasarkan interval pengujian pada user pelanggan didapatkan nilai 91,7 yang termasuk kategori sangat setuju Sehingga dapat disimpulkan bahwa pendapat user pelanggan sistem yang dikembangkan telah dapat diterima dengan hasil sangat setuju .

6.3.3 Hasil Akhir Pengujian *User Acceptance Testing*

Berdasarkan hasil pengujian User Acceptance Testing (UAT) yang telah dilakukan tersebut, diperoleh hasil sebagai berikut:

1. Pengujian UAT yang dilakukan oleh admin dengan jumlah responden sebanyak 1 orang menghasilkan nilai presentase UAT sebesar 88,9% dengan kriteria interpretasi penerimaan sangat setuju.
2. Pengujian UAT yang dilakukan oleh user pelanggan dengan jumlah responden sebanyak 3 orang menghasilkan nilai presentase UAT sebesar 91,7% dengan kriteria interpretasi penerimaan sangat setuju.

Dari nilai presentase pengujian UAT diatas, dapat ditarik hasil akhir pengujian dengan menghitung jumlah rata-rata nilai presentase pengujian user acceptance testing (UAT) pada semua responden, sebagai berikut:

$$\text{Hasil Akhir UAT} = ((88,9+91,7) / 2) \times 100\%$$

$$= (180,6 / 2) \times 100\%$$

$$= 90,3\%$$

Pengujian UAT yang telah dilakukan memperoleh hasil akhir dengan presentase nilai sebesar 90,3%. Berdasarkan kriteria interpretasi skor menurut interval yang didefinisikan pada Tabel 2.1, maka dapat disimpulkan bahwa sistem yang dikembangkan dapat diterima dengan hasil sangat setuju.

BAB 7 PENUTUP

7.1 Kesimpulan

Tahap-tahapan pengembangan Website E-commerce Butik Kebaya Iffah dengan menggunakan payment gateway midtrans telah dilalui mulai dari melakukan analisis kebutuhan dan perancangan sistem, kemudian dilanjutkan dengan melakukan implementasi dan pengujian sistem. Berdasarkan kegiatan tersebut, dapat diambil kesimpulan sebagai berikut :

1. Pada proses analisis kebutuhan pertama tama didapatkan pendefinisian proses bisnis as-is yang didapat setelah melakukan wawancara dengan pemilik butik, serta digambarkan oleh penulis mengenai proses bisnis to-be yang diharapkan memberikan perbaikan. Selanjutnya dilakukan analisi kebutuhan fungsional dan non-fungsional Website E-commerce Butik Kebaya Iffah, didapatkan 3 aktor, 9 kebutuhan fungsional untuk aktor pelanggan, 9 kebutuhan fungsional untuk aktor admin, dan 2 kebutuhan fungsional untuk aktor midtrans, serta 1 kebutuhan non fungsional *Compatibility testing*. Kemudian dari situlah dapat digambarkan *Use Case Diagram*, yang dijelaskan lebih detail langkah-langkahnya pada *use case scenario* yang selanjutnya scenario tersebut digambarkan menggunakan pemodelan Unified Modelling Language (UML). Selain itu juga dirancang mengenai gambaran user interface sistem, pengintegrasian sistem dengan API sandbox payment gateway midtrans, dan Pemodelan PDM (*Physical Data Model*) yang terdiri dari 9 tabel.
2. Pada proses implementasi diterapkan sistem yang sudah dirancang sebelumnya, yaitu implementasi *user interface*, implementasi *database*, dan implementasi kode program. Implementasi *user interface* menampilkan front-end dari Website E-commerce Butik Kebaya Iffah baik halaman e-commerce maupun halaman untuk admin . Implementasi *database* dilakukan dengan menggunakan MySQL dan phpMyAdmin yang terdiri atas 9 tabel yang dibutuhkan dalam sistem sesuai dengan perancangan. Serta pada bagian ini dijelaskan langkah konfigurasi API sandbox payment gateway midtrans untuk diterapkan pada codeIgniter. Dilanjutkan Implementasi kode program dilakukan dengan menggunakan bahasa pemrograman PHP menggunakan framewaork codeigniter dengan format Model-View-Controller (MVC).
3. Pada proses pengujian dilakukan metode *Black Box Testing* yang menguji validitas dari fungsi-fungsi yang ada pada Website E-commerce Butik Kebaya Iffah dapat berjalan dengan baik. Hasil dari pengujian

Black Box Testing didapatkan bahwa setiap fungsionalitas sistem telah terpenuhi dengan hasil 100% valid. Sedangkan untuk hasil pengujian *Compatibility Testing* dibantu dengan *tools SortSite* dan didapatkan bahwa sistem telah memenuhi kriteria *compatibility* tanpa terdapat *critical issues, minor issue, dan major issue*. Kemudian dilakukan juga pengujian *UAT (User Acceptance Test)* dengan hasil akhir dengan presentase nilai sebesar 90,3% bahwa sistem yang dikembangkan dapat diterima dengan hasil sangat setuju.

7.2 Saran

Berdasarkan perancangan Website E-commerce Butik Kebaya Iffah dengan menggunakan payment gateway midtrans, yang telah dilakukan bersama dengan objek penelitian terkait, Penulis ingin memberikan beberapa saran, yang dapat digunakan sebagai referensi untuk penelitian selanjutnya :

1. Hasil perancangan dan implementasi Website E-commerce Butik Kebaya Iffah dengan menggunakan payment gateway midtrans diharapkan kedepannya dapat dilakukan pengembangan lebih lanjut untuk menyempurnakan fitur custom, agar dapat ditambahkan fitur dimana pelanggan dapat melakukan custom model dari keinginan pelanggan yang lebih detail.
2. Diharapkan juga saat sistem dilakukan pengembangan lebih lanjut dapat ditambahkan integrasi dengan API Raja Ongkir untuk memudahkan penggunaan e-commerce hingga proses pengiriman.

DAFTAR REFERENSI

- Alatas, H. (2013). *Responsive Web Design*. Yogyakarta: Lokomedia.
- Arief, M. R. (2011). *Pemrograman web dinamis menggunakan PHP dan MySQL*. Yogyakarta: Andi.
- Damanik, E. (2012). *Perancangan Sistem Informasi Pembayaran*. JSM STMIK Mikroskil.
- Dwanoko, Y. S. (2016). Implementasi Software Development Life Cycle (Sdlc) Dalam Penerapan Pembangunan Aplikasi Perangkat. *Jurnal Teknologi Informasi*.
- Hanggara, B. T., & Ratnawati, D. E. (2020). Pengembangan Sistem Informasi Pemesanan Paket Wisata (Studi Kasus : Agen Wisata Liburan Sekolah). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 4(8), 2443–2449.
- Hartono, H. (2014). Pengertian Website Dan Fungsinya. *Jurnal Ilmu Teknologi Informasi*.
- Irmawati, D. (2011). Pemanfaatan e-commerce dalam dunia bisnis. *Jurnal Ilmiah Orasi Bisnis-ISSN, 2085(1375)*, 161–171.
- Janry Haposan U. P. Simanungkalit, S.Si., M. S. (2012). KONSEP DASAR SISTEM INFORMASI (Review). *Lecture Notes : Sistem Informasi*.
- Jardinia Tanuwidjaja, C., & Setiawan, A. (2017). Perancangan dan Pembuatan Website E-Commerce pada Toko Aksesoris Komputer di Surabaya. *Jurnal Infra*.
- Maulana, R., & Rachmawati, R. F. (2017). *Membangun Website E-Commerce Menggunakan Framework Codeigniter Pada Chemistry Merch*. 5(2), 86–96.
- Midtrans, 2020. Sejarah Midtrans. Midtrans. [online] Available at: [Accessed 15 September 2020] Midtrans, 2021. Snap Integration Guide. Midtrans. [online] Available at: [Accessed 7 April 2021] Midtrans, 2021. Testing Payment on Sandbox. Midtrans. Available at: [Accessed 7 April 2021]
- Nanda, A. P., & Maharani, A. (2018). Aplikasi Electronic Commerce Sebagai Media Penjualan Produk Makanan Ringan Business Development Center Kabupaten Pringsewu. *Jurnal TAM (Technology Acceptance Model)*.
- Nathan, A. J., & Scobell, A. (2012). IMPLEMENTASI DAN USER ACCEPTANCE TEST (UAT) TERHADAP APLIKASI E-LEARNING PADA MADRASAH ALIYAH NEGERI (MAN) 3 KOTA BANDA ACEH. *Foreign Affairs*.
- Nurhayati, L., & Setiadi, D. (2017). Pemodelan Proses Bisnis (Studi Kasus PD. Simpati Sumedang). *Infoman's*. <https://doi.org/10.33481/infomans.v11i1.20>
- Pradana, M. (2015). Klasifikasi Jenis-Jenis Bisnis E-Commerce. *Jurnal Neo-Bis*.
- Pratama, P. A. E. (2018). UAT Sistem Pendataan Penduduk Pendaftar di

Kabupaten Gianyar Berbasis Hybrid Cloud. *Journal of Chemical Information and Modeling*.

Rahmadiansyah, D., & Irwan, D. (2012). Implementasi Metode Model View Controller Menggunakan Framework Code Igniter dalam Pengembangan Aplikasi Manajemen Depo Petikemas pada Unit Usaha Belawan Logistics Center. *Snastikom*.

Rouf, A. (2012). *Pengujian Perangkat Lunak Dengan Menggunakan Metode White Box dan Back Box*, vol 8 no1, 1–7.
<http://www.ejournal.himsya.ac.id/index.php/HIMSYATECH/article/view/28/27>

Solichin, A. (2016). *Pemrograman web dengan PHP dan MySQL*. Penerbit Budi Luhur.

Sommerville, I. (2011). *Software engineering / Ian Sommerville*. — 9th ed. In *Addison-Wesley*.

Sukanto dan, & Shalahuddin(2014:28). (2014). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika Bandung. *Jurnal Pilar Nusa Mandiri*.

Supaartagorn, C. (2011). PHP Framework for database management based on MVC pattern. *International Journal of Computer Science & Information Technology (IJCSIT)*, 3(2), 251–258.

Supono, V. P. (2016). *Pemrograman Web dengan menggunakan PHP dan FRAMEWORK CODEIGNITER*. Yogyakarta: Deepublish.

Supriatna, R. (2019). *Implementasi Dan User Acceptance Test (UAT) Terhadap Aplikasi E-Learning pada Madrasah Aliyah Negeri (MAN) 3 Kota Banda Aceh*. UIN Ar-Raniry Banda Aceh.

Susanto, R., & Andriana, A. D. (2016). PERBANDINGAN MODEL WATERFALL DAN PROTOTYPING UNTUK PENGEMBANGAN SISTEM INFORMASI. *Majalah Ilmiah UNIKOM*. <https://doi.org/10.34010/miu.v14i1.174>

Tjandra, S., & Pickerling, C. (2015). APLIKASI METODE-METODE SOFTWARE TESTING PADA CONFIGURATION , COMPATIBILITY DAN USABILITY PERANGKAT LUNAK. *IDeaTech*.

DAFTAR LAMPIRAN

LAMPIRAN A HASIL WAWANCARA

HASIL WAWANCARA

Hari,Tanggal : Senin, 21 September 2020
 Tempat : Butik Kebaya Iffah
 Narasumber : Iffah Maulidiyah, S.T
 Jabatan : Owner Butik Kebaya Iffah

Daftar Pertanyaan

1. Kapan awal mula butik kebaya iffah berdiri?
Jawab : Kebaya iffah mulai berdiri sekitar Desember tahun 2015, oleh Saya sendiri sebagai pemiliknya.
2. Bisnis yang seperti apa, yang dijalankan butik kebaya iffah?
Jawab : *Brand* yang bergerak pada bisnis fashion terutama kebaya masa kini. Kami menjual kebaya yang lebih mengikuti model zaman sekarang, jadi lebih ringan dan dapat dipakai disegala acara.
3. Apa saja jenis-jenis produk yang dijual pada butik kebaya iffah?
Jawab : Di butik ini memang produk awal atau produk utamanya terfokus pada kebaya masa kini atau bisa disebut *dress* kebaya. Butik ini menjual berbagai macam *dress* kebaya, mulai dari kebaya pesta,kebaya lamaran, sampai kebaya wisuda, dan kebaya yang Kami jual merupakan hasil produksi jahitan sendiri. Namun, mulai tahun 2017 Butik ini juga menambah produk dengan menjual tidak hanya menjual kebaya saja tetapi juga menjual, *hijab*, *dress couple*, bahkan kemeja untuk laki laki dll.
4. Apa saja tipe penjualan yang dapat dilakukan oleh pembeli untuk melakukan transaksi pembelian produk butik kebaya iffah ?
Jawab : Jadi, di Butik kebaya iffah sendiri, awal mulanya memang hanya menjual produk *ready* atau siap pakai saja. Tetapi Kami mengembangkan bisnis dengan menambah tipe pembelian, yaitu Pelanggan dapat melakukan *custom* ukuran, jadi Kami mendengarkan permintaan Pelanggan , agar dapat menjual produk yang sesuai dengan permintaan ukuran yang diminta terutama untuk *big size*, yang cukup sulit untuk mendapatkan produk dengan ukuran yang pas. Selain itu beberapa bulan terakhir ini, Kami juga menambah tipe baru yaitu pembuatan sesuai permintaan pembeli, baik mulai dari modelnya, ukuran, bahan, motif kain brokat sampai jadi keseluruhan sesuai permintaan. Namun, memang tipe tersebut baru saja kami mulai beberapa bulan terakhir dengan beberapa pembeli saja dan permintaan khusus pembeli yang tidak semua Kami terima, karena model yang dikirim pelanggan harus ditanyakan dahulu ke pihak penjahit, jika bisa, pesanan akan diterima.
5. Untuk tipe pembelian produk *custom* ukuran, apa saja yang perlu dilakukan pengukuran? Dan bagaimana penentuan harga dari produk *custom* tersebut?
Jawab : Untuk produk *custom* ukuran, pelanggan biasanya datang langsung ke butik untuk melakukan pengukuran antara lain, lingkaran dada, lingkaran bahu, lingkaran pinggang, panjang tangan, dan panjang baju dalam satuan centimeter (cm). Selain itu juga bisa mau request *busui friendly* (tambah resleting) ataupun tidak. Untuk penentuan harga produk *custom* ukuran akan dikenakan biaya :
 - a. Biaya *custom* ukuran jika Lingkaran dada tidak lebih dari sama dengan 110 cm : Rp. 25.000
 - b. Biaya *custom* ukuran Jika Lingkaran dada lebih dari 110 cm : Rp.50.000
 - c. Biaya request *Busui friendly* (resleting) : Rp. 10.000
6. Siapa saja target market pembeli produk butik kebaya iffah?

Lampiran 1 Transkrip Wawancara

Jawab : Target utama pembeli butik kebaya iffah sendiri adalah perempuan dewasa dan anak remaja yang hendak wisuda, Namun karena kami menjual *dress* kebaya yang memang dapat digunakan diberbagai acara sehingga pelanggan kami tidak hanya itu saja bahkan banyak ibu-ibu yang juga membeli produk kami.

7. Melalui media apa saja pihak butik kebaya iffah melakukan promosi produk kepada pelanggan?

Jawab : Sejauh ini untuk melakukan promosi produk, Kami menggunakan *media social* Facebook, Instagram, dan Whatsapp. *Media social* utama sebagai katalog dan yang selalu *update* adalah Instagram.

8. Apakah melalui ketiga *media social* tersebut dirasa sudah cukup efektif dalam promosi bisnis anda?

Jawab : Belum, dikarenakan Saya sendiri (pemilik) juga masih belajar untuk mempergunakan *media social* sebagai sarana *digital marketing* promosi yang baik, dan dikarenakan saya juga belum menerapkan fitur iklan yang disediakan oleh platform, jadi saya hanya upload gambar pada feed sebagai katalog saja, dan karena jumlah yang melihat kiriman tersebut tidak terlalu banyak, saya juga mempromosikan lewat fitur cerita, namun kita tahu bahwa fitur tersebut hanya bertahan selama 24 jam saja, jadi memang saya rasa itu kurang cukup menarik minat pembeli dan kurang efektif.

9. Bagaimana mekanisme penjualan dan pembelian produk pada butik kebaya iffah?

Jawab : Mekanisme penjualan dan pembelian produk selama ini dapat dilakukan secara 2 cara yaitu *offline* atau datang langsung ke butik yang berada Di jalan lasem no.5 Gresik kota baru Kabupaten Gresik Jawa Timur. Atau yang kedua secara *online* yaitu lewat *media social* Kami Whatsapp.

10. Bagaimana mekanisme transaksi pembelian produk melalui whatsapp?

Jawab : Untuk transaksi pembelian secara online memang hanya dapat dilakukan melalui whatsapp, Langkah pertama pelanggan harus melihat katalog produk yang diinginkan pada instagram kemudian pelanggan dapat menghubungi whatsapp butik yang tertera pada biodata instagram kemudian mengirimkan gambar produk yang diinginkan, jika pelanggan ingin bertanya tanya terlebih dahulu maka akan dijawab melalui pesan whatsapp, jika pelanggan sudah yakin membeli produk, pelanggan dapat mengirimkan format order berisi nama, nomor HP, alamat, dll. Kemudian Admin akan membuatkan totalan transaksi, dan pelanggan dapat melakukan pembayaran kemudian mengirimkan bukti transfer ke whatsapp, lalu admin mengkonfirmasi pesanan dan produk akan dikirim.

11. Apakah dari kedua mekanisme tersebut memiliki permasalahan yang mengganggu berjalannya bisnis, penjualan, maupun pembelian produk?

Jawab : Sejauh ini, untuk metode *offline* atau datang langsung ketoko tidak mengalami kendala. Namun, untuk metode *online* memang masih dirasa kurang efektif dan efisien. Selain masih dilakukan secara manual, juga kurangnya pemanfaatan pada *social media* kami, dan juga untuk pemesanan custom ukuran, pelanggan harus datang dulu ke butik untuk melakukan pengukuran. Selain itu kami juga punya permasalahan dalam pembayaran karena terdapat beberapa pelanggan yang meminta untuk menambah metode pembayaran yang lebih mudah seperti lewat minimarket. Juga bagi admin masih melakukan pengecekan secara manual karena tidak seluruh metode memiliki *internet banking* jadi tidak dapat memantau secara *real-time* transaksi pembayaran pelanggan, sehingga pernah hampir terdapat penipuan dengan pelanggan mengirimkan bukti transfer palsu.

12. Apakah sudah ada pembangunan sistem informasi pada butik kebaya iffah untuk media promosi maupun penjualan dan pembelian produk?

Lampiran 2 Transkrip Wawancara Lembar Ke-2

Jawab : Sampai saat ini belum, karena proses penjualan dan pembelian produk masih dilakukan secara manual seperti yang sudah dijelaskan sebelumnya.

13. Apabila dibuatkan sistem e-commerce dengan fitur mengelola data produk, mempermudah proses penjualan dan pembelian produk, serta fitur *custom* ukuran, dan dilengkapi dengan sistem pembayaran dengan metode pembayaran yang lengkap serta aman apakah dapat membantu menyelesaikan permasalahan tersebut?

Jawab : Sangat membantu, Proses bisnis yang berjalan saat ini memang masih manual, jadi Saya rasa jika diterapkan dengan teknologi akan lebih memudahkan. Dan Fitur yang ditawarkan juga menarik dan sudah cukup lengkap karena menurut saya dapat membantu menyelesaikan permasalahan seperti yang telah saya sebutkan, sudah terwakili dengan fitur yang ada. Selain itu saya rasa dengan adanya sistem ini, akan dapat menjangkau lebih banyak pelanggan dan memudahkan pelanggan tanpa perlu datang ke butik serta menjawab keinginan pelanggan.

Menyetujui,
Owner Butik Kebaya Iffah



Iffah Maulidiyah S.T

LAMPIRAN B HASIL PENGUJIAN BLACK-BOX TESTING

HASIL PENGUJIAN BLACK-BOX TESTING

Keterangan Hasil Uji :

V = Valid, yaitu ketika hasil yang diharapkan sesuai dengan output yang dihasilkan.

TV = Tidak Valid, yaitu ketika hasil yang diharapkan tidak sesuai dengan output yang dihasilkan.

Berikan tanda centang (✓) pada pilihan hasil uji yang dianggap sesuai.

No	Nama Fungsional	Skenario Uji	Hasil Yang Diharapkan	Hasil Uji	
				V	TV
1	Melakukan Registrasi	Aktor mengosongkan salah satu atau seluruh <i>field</i> pada <i>form</i> registrasi, lalu menekan "Registrasi"	Sistem akan menolak akses Registrasi dan menampilkan halaman Registrasi beserta <i>alert</i> untuk mengisi <i>field</i> yang kosong.	✓	
		Aktor mengisi <i>field</i> dengan data yang tidak sesuai dengan format yang dibutuhkan, lalu menekan "Registrasi"	Sistem akan menolak akses Registrasi dan menampilkan halaman Registrasi beserta <i>alert</i> untuk mengisi kembali <i>field</i> dengan format yang sesuai.	✓	
		Aktor mengisi Nama lengkap, <i>Email</i> yang sudah terdaftar sebagai pelanggan, <i>Password</i> , telepon dan alamat lalu menekan "Registrasi"	Sistem akan menolak akses Registrasi dan mengembalikan <i>User</i> menuju halaman Registrasi beserta pesan untuk registrasi kembali dengan email yang belum terdaftar.	✓	
		Aktor mengisi Nama lengkap, <i>Email</i> yang belum terdaftar sebagai pelanggan, <i>Password</i> , telepon dan alamat lalu menekan "Registrasi"	Sistem akan menerima akses Registrasi dan aktor memasuki halaman sukses registrasi.	✓	
2	Login	Aktor mengosongkan salah satu atau kedua <i>field Email</i> dan <i>Password</i> , lalu menekan tombol "Log In".	Sistem akan menolak akses <i>Login</i> dan menampilkan halaman <i>Login</i> beserta <i>alert</i> untuk mengisi <i>field</i> yang kosong.	✓	
		Aktor mengisi <i>field Email</i> dengan format yang tidak sesuai, dan <i>password</i> lalu menekan tombol "Log In".	Sistem akan menolak akses <i>Login</i> dan menampilkan halaman <i>Login</i> beserta <i>alert</i> untuk mengisi kembali <i>field Email</i> dengan format yang sesuai.	✓	
		Aktor mengisi <i>Email</i> dan <i>Password</i> dengan data kombinasi yang salah, lalu menekan tombol "Log In".	Sistem akan menolak akses <i>Login</i> dan mengembalikan <i>User</i> menuju halaman <i>Login</i> beserta pesan kesalahan.	✓	

Lampiran 4 Hasil Pengujian Black-Box Testing oleh Owner Lembar Ke-1

		Aktor mengisi kedua <i>field Email</i> dan <i>Password</i> dengan data yang benar, lalu menekan tombol "Log In".	Sistem akan menerima akses <i>login</i> dan aktor memasuki halaman <i>Dashboard</i> Pelanggan.	✓	
3	<i>Logout</i>	Aktor menekan tombol "Logout"	Sistem akan menghapus <i>session</i> aktor dan mengembalikan <i>User</i> menuju halaman <i>Login</i>	✓	
4	Melihat Daftar Produk	Aktor memilih menu "Shop"	Sistem akan menampilkan seluruh daftar produk Butik Kebaya Iffah	✓	
		Aktor memilih menu "Shop" dan memilih kategori produk	Sistem akan menampilkan daftar produk Butik Kebaya Iffah sesuai dengan kategori yang dipilih	✓	
5	Mengelola Keranjang Belanja	Aktor menambahkan produk, dengan menekan tombol "add to cart"	Sistem akan menambahkan produk ke keranjang belanja, dan menampilkan <i>pop up</i> berhasil	✓	
		Aktor melihat keranjang belanja namun belum terdapat produk	Sistem menampilkan teks "Keranjang Belanja Kosong"	✓	
		Aktor melihat keranjang belanja dan terdapat produk	Sistem akan menampilkan data produk pada keranjang belanja	✓	
		Aktor menambahkan jumlah item produk, lalu menekan tombol "Update"	Sistem akan menambahkan jumlah item produk dan menghitung kembali subtotal pada produk tersebut	✓	
		Aktor mengurangi jumlah item produk, lalu menekan tombol "Update"	Sistem akan mengurangi jumlah item produk dan menghitung kembali subtotal pada produk tersebut	✓	
		Aktor menghapus salah satu produk pada keranjang, dengan menekan tombol "Delete"	Sistem akan menghapus baris data produk yang terpilih dari keranjang belanja	✓	
		Aktor menghapus seluruh produk pada keranjang, dengan menekan tombol "Kosongkan Keranjang"	Sistem akan mengosongkan keranjang belanja	✓	
6	Melakukan Custom Order	Aktor memilih menu "Custom"	Sistem akan menampilkan daftar produk Butik Kebaya Iffah yang dapat dilakukan <i>Custom Size</i>	✓	
		Aktor membuka form <i>custom order</i> , dengan menekan tombol "custom size"	Sistem akan menampilkan <i>form custom size</i>	✓	

Lampiran 5 Hasil Pengujian Black-Box Testing oleh Owner Lembar Ke-2

		Aktor mengosongkan salah satu atau seluruh <i>field</i> pada form custom order	Sistem akan menolak masukan data dan menampilkan <i>form custom size</i> beserta <i>alert</i> untuk mengisi <i>field</i> yang kosong.	✓	
		Aktor mengupload gambar selain format yang diperbolehkan (gif, jpg, png, jpeg) dan melebihi dari ukuran file maksimal	Sistem akan menolak masukan data dari aktor, dan mengembalikan pada halaman <i>form custom size</i>	✓	
		Aktor mengisi seluruh <i>field</i> form custom order dengan format data yang sesuai, dan Aktor mengupload gambar dengan format yang diperbolehkan (gif, jpg, png, jpeg) dan kurang dari ukuran file maksimal namun aktor belum melakukan <i>login</i>	Sistem akan menolak masukan data dan mengembalikan User dengan menampilkan halaman <i>Login</i>	✓	
		Aktor sudah login, lalu mengisi seluruh <i>field</i> form custom order dengan format data yang sesuai, dan Aktor mengupload gambar dengan format yang diperbolehkan (gif, jpg, png, jpeg) dan kurang dari ukuran file maksimal	Sistem akan menerima masukan data dari aktor, dan memasuki halaman <i>Checkout Custom</i> beserta menampilkan harga <i>custom order</i> sesuai perhitungan data masukan	✓	
7	Melakukan <i>Checkout</i>	Aktor belum login, dan menekan tombol "Checkout" pada halaman keranjang belanja untuk proses checkout produk ready	Sistem akan mengembalikan aktor dan menampilkan halaman <i>Login</i>	✓	
		Aktor sudah login, dan menekan tombol "Checkout" pada halaman keranjang belanja untuk proses checkout produk ready	Sistem akan menampilkan halaman checkout untuk proses checkout produk ready	✓	
		Aktor mengosongkan salah satu atau seluruh <i>field</i> pada form <i>Checkout</i> pada halaman <i>checkout</i>	Sistem akan menolak proses <i>checkout</i> dan menampilkan form <i>checkout</i> beserta <i>alert</i> untuk mengisi <i>field</i> yang kosong	✓	
		Aktor mengisi <i>field</i> dengan data yang tidak sesuai dengan format yang dibutuhkan	Sistem akan menolak proses <i>checkout</i> dan menampilkan form <i>checkout</i> beserta <i>alert</i> untuk mengisi <i>field</i> dengan format yang sesuai	✓	

Lampiran 6 Hasil Pengujian Black-Box Testing oleh Owner Lembar Ke-3

		Aktor sudah login, dan mengisi seluruh <i>field</i> form <i>checkout</i> dengan format data yang sesuai, lalu menekan tombol "Buat Pesanan"	Sistem akan memverifikasi total pesanan, dan proses akan dilanjutkan dengan menampilkan halaman pembayaran midtrans	✓	
8	Melakukan Pembayaran	Aktor telah menekan tombol "Buat Pesanan"	Sistem akan menampilkan SNAP Payment midtrans	✓	
		Aktor menekan Lanjut, kemudian aktor dapat memilih metode pembayaran, lalu lihat nomor rekening dan terakhir, menekan 'mohon selesaikan pembayaran'	Sistem akan menampilkan lanjutan dari halaman rincian yaitu dilanjutkan tampil cara pembayaran, kemudian tampil nomor rekening dan batas pembayaran, dan setelah proses terlewati, akan ditampilkan halaman sukses	✓	
9	Melihat Riwayat Transaksi	Aktor memilih menu "Riwayat Belanja" pada <i>dashboard</i> pelanggan	Sistem akan menampilkan riwayat belanja pembelian produk oleh pelanggan	✓	
10	<i>Login admin</i>	Aktor mengosongkan salah satu atau kedua <i>field</i> <i>Username</i> dan <i>Password</i> , lalu menekan tombol "Login"	Sistem akan menolak akses <i>Login</i> dan menampilkan halaman <i>Login admin</i> beserta <i>alert</i> untuk mengisi <i>field</i> yang kosong	✓	
		Aktor mengisi <i>Username</i> dan <i>Password</i> dengan data kombinasi yang salah, lalu menekan tombol "Login"	Sistem akan menolak akses <i>Login</i> dan mengembalikan <i>User</i> menuju halaman <i>Login</i> beserta pesan kesalahan	✓	
		Aktor mengisi kedua <i>field</i> <i>Username</i> dan <i>Password</i> dengan data yang benar, lalu menekan tombol "Login"	Sistem akan menerima akses <i>login</i> dan aktor memasuki halaman <i>Dashboard Admin</i>	✓	
11	<i>Logout</i>	Aktor menekan tombol "Logout"	Sistem akan menghapus <i>session</i> aktor dan mengembalikan <i>User</i> menuju halaman <i>Login admin</i>	✓	
12	Melihat <i>Dashboard</i>	Aktor memilih menu "Dashboard"	Sistem akan menampilkan halaman <i>dashboard</i> yang berisi informasi ringkasan sistem	✓	
13	Mengelola Produk	Aktor memilih menu "Produk"	Sistem akan menampilkan seluruh daftar produk	✓	
		Aktor menekan tombol "Tambah", pada halaman produk	Sistem akan menampilkan <i>form</i> tambah produk	✓	
		Aktor mengosongkan salah satu atau seluruh <i>field</i> pada <i>form</i> tambah produk	Sistem akan menolak masukan data dan menampilkan <i>form</i> tambah	✓	

Lampiran 7 Hasil Pengujian Black-Box Testing oleh Owner Lembar Ke-4

			produk beserta <i>alert</i> untuk mengisi <i>field</i> yang kosong		
		Aktor mengisi <i>field</i> kode produk dengan data yang sudah terdaftar	Sistem berhasil menolak masukan data dan mengembalikan <i>User</i> menuju halaman <i>form</i> tambah produk beserta pesan untuk mengisi kembali <i>field</i> dengan kode produk yang belum terdaftar	✓	
		Aktor mengupload gambar selain format yang diperbolehkan (gif, jpg, png, jpeg) dan melebihi dari ukuran file maksimal	Sistem berhasil menolak masukan data dan mengembalikan <i>User</i> menuju halaman <i>form</i> tambah produk beserta pesan kesalahan upload gambar	✓	
		Aktor mengisi seluruh <i>field</i> form tambah produk dengan kode produk yang belum terdaftar, dan Aktor mengupload gambar dengan format yang diperbolehkan (gif, jpg, png, jpeg) dan kurang dari ukuran file maksimal	Sistem berhasil menerima masukan data dari aktor, dan menampilkan produk pada daftar	✓	
		Aktor menekan tombol "Edit" untuk melakukan perubahan data produk pada form edit data produk	Sistem berhasil menerima perubahan data produk yang diubah aktor, dan memperbaruinya pada daftar	✓	
		Aktor menekan tombol "Delete" untuk menghapus data produk	Sistem berhasil menghapus data produk terpilih dari daftar produk	✓	
14	Melihat Data Pesanan	Aktor memilih menu "Laporan Transaksi"	Sistem akan menampilkan daftar transaksi pembelian produk ready	✓	
15	Melihat Data Custom Order	Aktor memilih menu "Laporan Custom"	Sistem akan menampilkan daftar transaksi pembelian produk custom	✓	
16	Update Status Bayar	Aktor mengosongkan <i>field</i> cek status, lalu menekan tombol 'cari'	Sistem akan menampilkan pesan "Order id yang anda cari tidak ada"	✓	
		Aktor memasukkan kode transaksi yang tidak terdaftar pada <i>field</i> cek status, lalu menekan tombol 'cari'	Sistem akan menampilkan pesan "Order id yang anda cari tidak ada"	✓	
		Aktor memasukkan kode transaksi yang terdaftar pada <i>field</i> cek status, lalu menekan tombol 'cari'	Sistem akan mengganti dengan status terbaru, dan menampilkan warna sesuai status bayar, merah untuk <i>failure</i> , kuning untuk <i>pending</i> , dan hijau untuk <i>settlement</i>	✓	

Lampiran 8 Hasil Pengujian Black-Box Testing oleh Owner Lembar Ke-5

17	Mengelola Admin	Aktor memilih menu "Admin"	Sistem akan menampilkan daftar admin yang sudah diinputkan sebelumnya	✓	
		Aktor menekan tombol "Tambah", pada halaman admin	Sistem akan menampilkan form tambah admin	✓	
		Aktor mengosongkan salah satu atau seluruh field pada form tambah admin	Sistem akan menolak masukan data dan menampilkan form tambah admin beserta alert untuk mengisi field yang kosong	✓	
		Aktor mengisi field dengan data yang tidak sesuai dengan format yang dibutuhkan pada form tambah admin	Sistem akan menolak masukan data dan menampilkan form tambah admin beserta alert untuk mengisi kembali field Email dengan format yang sesuai	✓	
		Aktor mengisi seluruh field form tambah admin dengan format data yang sesuai	Sistem akan menerima masukan data dari aktor, dan menampilkan admin pada daftar	✓	
		Aktor menekan tombol "Edit" untuk melakukan perubahan data admin pada form edit data admin	Sistem akan menerima perubahan data admin yang diubah aktor, dan memperbaruinya pada daftar	✓	
		Aktor menekan tombol "Delete" untuk menghapus data admin	Sistem akan menghapus data admin terpilih dari daftar	✓	
18	Mencetak Detail Pesanan	Aktor membuka detail transaksi, lalu menekan tombol "cetak"	Sistem akan menampilkan halaman unduh file cetak detail pesanan	✓	

Gresik, 29 Mei 2021

Menyetujui,
Owner Butik Kebaya Iffah

Iffah Maulidiyah S.T

HASIL PENGUJIAN BLACK-BOX TESTING

Keterangan Hasil Uji :

V = Valid, yaitu ketika hasil yang diharapkan sesuai dengan output yang dihasilkan.

TV = Tidak Valid, yaitu ketika hasil yang diharapkan tidak sesuai dengan output yang dihasilkan.

Berikan tanda centang (✓) pada pilihan hasil uji yang dianggap sesuai.

No	Nama Fungsional	Skenario Uji	Hasil Yang Diharapkan	Hasil Uji	
				V	TV
1	Melakukan Registrasi	Aktor mengosongkan salah satu atau seluruh <i>field</i> pada <i>form</i> registrasi, lalu menekan "Registrasi"	Sistem akan menolak akses Registrasi dan menampilkan halaman Registrasi beserta <i>alert</i> untuk mengisi <i>field</i> yang kosong.	✓	
		Aktor mengisi <i>field</i> dengan data yang tidak sesuai dengan format yang dibutuhkan, lalu menekan "Registrasi"	Sistem akan menolak akses Registrasi dan menampilkan halaman Registrasi beserta <i>alert</i> untuk mengisi kembali <i>field</i> dengan format yang sesuai.	✓	
		Aktor mengisi Nama lengkap, <i>Email</i> yang sudah terdaftar sebagai pelanggan, <i>Password</i> , telepon dan alamat lalu menekan "Registrasi"	Sistem akan menolak akses Registrasi dan mengembalikan <i>User</i> menuju halaman Registrasi beserta pesan untuk registrasi kembali dengan email yang belum terdaftar.	✓	
		Aktor mengisi Nama lengkap, <i>Email</i> yang belum terdaftar sebagai pelanggan, <i>Password</i> , telepon dan alamat dengan format yang sesuai, lalu menekan "Registrasi"	Sistem akan menerima akses Registrasi dan aktor memasuki halaman sukses registrasi.	✓	
2	Login	Aktor mengosongkan salah satu atau kedua <i>field Email</i> dan <i>Password</i> , lalu menekan tombol "Log In".	Sistem akan menolak akses <i>Login</i> dan menampilkan halaman <i>Login</i> beserta <i>alert</i> untuk mengisi <i>field</i> yang kosong.	✓	
		Aktor mengisi <i>field Email</i> dengan format yang tidak sesuai, dan <i>password</i> lalu menekan tombol "Log In".	Sistem akan menolak akses <i>Login</i> dan menampilkan halaman <i>Login</i> beserta <i>alert</i> untuk mengisi kembali <i>field</i>	✓	

Lampiran 10 Hasil Pengujian Black-Box Testing oleh Pelanggan Lembar Ke-1

			Email dengan format yang sesuai.		
		Aktor mengisi Email dan Password dengan data kombinasi yang salah, lalu menekan tombol "Log In".	Sistem akan menolak akses Login dan mengembalikan User menuju halaman Login beserta pesan kesalahan.	✓	
		Aktor mengisi kedua field Email dan Password dengan data yang benar, lalu menekan tombol "Log In".	Sistem akan menerima akses login dan aktor memasuki halaman Dashboard Pelanggan.	✓	
3	Logout	Aktor menekan tombol "Logout"	Sistem akan menghapus session aktor dan mengembalikan User menuju halaman Login	✓	
4	Melihat Daftar Produk	Aktor memilih menu "Shop"	Sistem akan menampilkan seluruh daftar produk Butik Kebaya Iffah	✓	
		Aktor memilih menu "Shop" dan memilih kategori produk	Sistem akan menampilkan daftar produk Butik Kebaya Iffah sesuai dengan kategori yang dipilih	✓	
5	Mengelola Keranjang Belanja	Aktor menambahkan produk , dengan menekan tombol "add to cart"	Sistem akan menambahkan produk ke keranjang belanja, dan menampilkan pop up berhasil	✓	
		Aktor melihat keranjang belanja namun belum terdapat produk	Sistem menampilkan teks "Keranjang Belanja Kosong"	✓	
		Aktor melihat keranjang belanja dan terdapat produk	Sistem akan menampilkan data produk pada keranjang belanja	✓	
		Aktor menambahkan jumlah item produk, lalu menekan tombol "Update"	Sistem akan menambahkan jumlah item produk dan menghitung kembali subtotal pada produk tersebut	✓	
		Aktor mengurangi jumlah item produk, lalu menekan tombol "Update"	Sistem akan mengurangi jumlah item produk dan menghitung kembali subtotal pada produk tersebut	✓	
		Aktor menghapus salah satu produk pada keranjang, dengan menekan tombol "Delete"	Sistem akan menghapus baris data produk yang terpilih dari keranjang belanja	✓	
		Aktor menghapus seluruh produk pada keranjang, dengan menekan tombol "Kosongkan Keranjang"	Sistem akan mengosongkan keranjang belanja	✓	

Lampiran 11 Hasil Pengujian Black-Box Testing oleh Pelanggan Lembar Ke-2

6	Melakukan Custom Order	Aktor memilih menu "Custom"	Sistem akan menampilkan daftar produk Butik Kebaya Iffah yang dapat dilakukan Custom Size	✓	
		Aktor membuka form custom order , dengan menekan tombol "custom size"	Sistem akan menampilkan form custom size	✓	
		Aktor mengosongkan salah satu atau seluruh field pada form custom order	Sistem akan menolak masukan data dan menampilkan form custom size beserta alert untuk mengisi field yang kosong.	✓	
		Aktor mengupload gambar selain format yang diperbolehkan(gif,jpg,png,jpeg) dan melebihi dari ukuran file maksimal	Sistem akan menolak masukan data dari aktor, dan mengembalikan pada halaman form custom size	✓	
		Aktor mengisi seluruh field form custom order dengan format data yang sesuai, dan Aktor mengupload gambar dengan format yang diperbolehkan(gif,jpg,png,jpeg) dan kurang dari ukuran file maksimal namun aktor belum melakukan login	Sistem akan menolak masukan data dan mengembalikan User dengan menampilkan halaman Login	✓	
		Aktor sudah login, lalu mengisi seluruh field form custom order dengan format data yang sesuai, dan Aktor mengupload gambar dengan format yang diperbolehkan(gif,jpg,png,jpeg) dan kurang dari ukuran file maksimal	Sistem akan menerima masukan data dari aktor, dan memasuki halaman Checkout Custom beserta menampilkan harga custom order sesuai perhitungan data masukan	✓	
7	Melakukan Checkout	Aktor belum login, dan menekan tombol "Checkout" pada halaman keranjang belanja untuk proses checkout produk ready	Sistem akan mengembalikan aktor dan menampilkan halaman Login	✓	
		Aktor sudah login, dan menekan tombol "Checkout" pada halaman keranjang belanja untuk proses checkout produk ready	Sistem akan menampilkan halaman checkout untuk proses checkout produk ready	✓	
		Aktor mengosongkan salah satu atau seluruh field pada form Checkout pada halaman checkout	Sistem akan menolak proses checkout dan menampilkan form checkout beserta alert	✓	

Lampiran 12 Hasil Pengujian Black-Box Testing oleh Pelanggan Lembar Ke-3

			untuk mengisi <i>field</i> yang kosong		
		Aktor mengisi <i>field</i> dengan data yang tidak sesuai dengan format yang dibutuhkan	Sistem akan menolak proses <i>checkout</i> dan menampilkan <i>form checkout</i> beserta <i>alert</i> untuk mengisi <i>field</i> dengan format yang sesuai	✓	
		Aktor sudah login, dan mengisi seluruh <i>field</i> form <i>checkout</i> dengan format data yang sesuai, lalu menekan tombol "Buat Pesanan"	Sistem akan memverifikasi total pesanan, dan proses akan dilanjutkan dengan menampilkan halaman pembayaran midtrans	✓	
8	Melakukan Pembayaran	Aktor telah menekan tombol "Buat Pesanan"	Sistem akan menampilkan SNAP Payment midtrans	✓	
		Aktor menekan Lanjut, kemudian aktor dapat memilih metode pembayaran, lalu lihat nomor rekening dan terakhir, menekan 'mohon selesaikan pembayaran'	Sistem akan menampilkan lanjutan dari halaman rincian yaitu dilanjutkan tampil cara pembayaran, kemudian tampil nomor rekening dan batas pembayaran, dan setelah proses terlewati, akan ditampilkan halaman sukses	✓	
9	Melihat Riwayat Transaksi	Aktor memilih menu "Riwayat Belanja" pada <i>dashboard</i> pelanggan	Sistem akan menampilkan riwayat belanja pembelian produk oleh pelanggan	✓	

Gresik, 3 Juni 2021

Menyetujui,
Pelanggan Butik Kebaya Iffah



Radina Tri Cahyanti, S.Hum

Lampiran 13 Hasil Pengujian Black-Box Testing oleh Pelanggan Lembar Ke-4

**FORM VALIDASI USER ACCEPTANCE TEST
PENGEMBANGAN WEBSITE E-COMMERCE BUTIK KEBAYA IFFAH
MENGUNAKAN PAYMENT GATEWAY MIDTRANS**

Nama : Iffah Maulidiyah, S.T						
Jabatan : Owner butik kebaya Iffah						
No.	Pertanyaan	Jawaban				
		Sangat Setuju	Setuju	Netral	Tidak Setuju	Sangat Tidak Setuju
1.	Apakah desain tampilan sistem menarik dan sesuai?	✓				
2.	Apakah dengan dibuatnya sistem ini mempercepat dan mempermudah proses pengelolaan daftar produk?		✓			
3.	Form untuk menginputkan data produk, apakah sudah lengkap seperti pencatatan manual?		✓			
4.	Apakah proses pengelolaan transaksi pembelian produk dan pembayaran lebih mudah, aman dan cepat?	✓				
5.	Dengan adanya fitur update status, apakah dapat berjalan aman, dan meminimalisir penipuan?	✓				
6.	Apakah proses bisnis transaksi pembelian produk custom dapat berjalan baik, dengan adanya sistem ini?			✓		
7.	Apakah penggunaan fitur pada sistem mudah dipahami?	✓				
8.	Apakah terdapat kendala saat anda menjalankan sistem ini?	✓				
9.	Apakah sistem yang dibuat pada aplikasi sudah sesuai dengan apa yang diharapkan?		✓			
Saran : - Dapat ditambahkan form warna pada inputan - Dapat dikembangkan update status otomatis.						

Gresik, 29 Mei 2021

Menyetujui,
Owner Butik Kebaya Iffah

Iffah Maulidiyah S.T

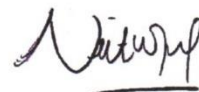
Lampiran 14 Hasil Pengujian UAT Responden Admin 1

**FORM VALIDASI USER ACCEPTANCE TEST
PENGEMBANGAN WEBSITE E-COMMERCE BUTIK KEBAYA IFFAH
MENGUNAKAN PAYMENT GATEWAY MIDTRANS**

Nama : Anita Wardah f.						
Jabatan : Pelanggan (30 tahun, Ibu rumah tangga)						
No.	Pertanyaan	Jawaban				
		Sangat Setuju	Setuju	Netral	Tidak Setuju	Sangat Tidak Setuju
1.	Apakah desain dan layout tampilan sistem menarik, sesuai serta mudah dipahami?		✓			
2.	Apakah penggunaan fitur-fitur pada sistem mudah dipelajari ?		✓			
3.	Apakah dengan dibuatnya sistem ini mempercepat transaksi pembelian anda?	✓				
4.	Dengan adanya fitur custom order memudahkan keinginan pelanggan, dan dapat berjalan dengan baik ?			✓		
5.	Apakah proses pembayaran lebih efisien, aman dan cepat?	✓				
6.	Apakah terdapat kendala saat anda menjalankan sistem ini?			✓		
7.	Apakah sistem ini sangat bermanfaat?	✓				
8.	Apakah sistem yang dibuat pada aplikasi sudah sesuai dengan apa yang diharapkan?	✓				
Saran : - Dapat ditambahkan petunjuk penggunaan						

Gresik, 3 Juni 2021

Menyetujui,
Pelanggan Butik Kebaya Iffah



Anita Wardah Fitriyah, S.Si

Lampiran 15 Hasil Pengujian UAT Responden Pelanggan 1

**FORM VALIDASI USER ACCEPTANCE TEST
PENGEMBANGAN WEBSITE E-COMMERCE BUTIK KEBAYA IFFAH
MENGUNAKAN PAYMENT GATEWAY MIDTRANS**

Nama : Radina Tri Cahyanti						
Jabatan : Pelanggan (Pegawai Swasta, 25 tahun)						
No.	Pertanyaan	Jawaban				
		Sangat Setuju	Setuju	Netral	Tidak Setuju	Sangat Tidak Setuju
1.	Apakah desain dan layout tampilan sistem menarik, sesuai serta mudah dipahami?	✗	✓			
2.	Apakah penggunaan fitur fitur pada sistem mudah dipelajari ?	✓				
3.	Apakah dengan dibuatnya sistem ini mempercepat transaksi pembelian anda?	✓				
4.	Dengan adanya fitur custom order memudahkan keinginan pelanggan, dan dapat berjalan dengan baik ?	✓				
5.	Apakah proses pembayaran lebih efisien, aman dan cepat?	✓				
6.	Apakah terdapat kendala saat anda menjalankan sistem ini?	✓				
7.	Apakah sistem ini sangat bermanfaat?	✓				
8.	Apakah sistem yang dibuat pada aplikasi sudah sesuai dengan apa yang diharapkan?		✓			
Saran : - Waktu pembayaran dapat diperpanjang						

Gresik, 3 Juni 2021

Menyetujui,
Pelanggan Butik Kebaya Iffah



Radina Tri Cahyanti, S.Hum

**FORM VALIDASI USER ACCEPTANCE TEST
PENGEMBANGAN WEBSITE E-COMMERCE BUTIK KEBAYA IFFAH
MENGUNAKAN PAYMENT GATEWAY MIDTRANS**

Nama : Afrilita Hidayatul Lail						
Jabatan : Pelanggan (Mahasiswa, 18 tahun)						
No.	Pertanyaan	Jawaban				
		Sangat Setuju	Setuju	Netral	Tidak Setuju	Sangat Tidak Setuju
1.	Apakah desain dan layout tampilan sistem menarik, sesuai serta mudah dipahami?	✓				
2.	Apakah penggunaan fitur-fitur pada sistem mudah dipelajari ?	✓				
3.	Apakah dengan dibuatnya sistem ini mempercepat transaksi pembelian anda?	✓				
4.	Dengan adanya fitur custom order memudahkan keinginan pelanggan, dan dapat berjalan dengan baik ?		✓			
5.	Apakah proses pembayaran lebih efisien, aman dan cepat?	✓				
6.	Apakah terdapat kendala saat anda menjalankan sistem ini?	✓				
7.	Apakah sistem ini sangat bermanfaat?	✓				
8.	Apakah sistem yang dibuat pada aplikasi sudah sesuai dengan apa yang diharapkan?		✓			
Saran : —						

Gresik, 3 Juni 2021

Menyetujui,
Pelanggan Butik Kebaya Iffah



Afrilita Hidayatul Lail

Lampiran 17 Hasil Pengujian UAT Responden Pelanggan 3