PENGEMBANGAN *E – COMMERCE* DENGAN PEMANFAATAN SISTEM *PAYMENT GATEWAY*
(STUDI KASUS: WISATA KAMPUNG SAPI *ADVENTURE*)

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Yuninda Eka Nisrina
NIM: 165150707111005

PROGRAM STUDI TEKNOLOGI INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2020

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PENGESAHAN

PENGEMBANGAN E – COMMERCE DENGAN PEMANFAATAN SISTEM *PAYMENT GATEWAY* (STUDI KASUS: WISATA KAMPUNG SAPI *ADVENTURE*)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
Memperoleh gelar Sarjana Komputer

Disusun Oleh :
Yuninda Eka Nisrina
NIM: 165150707111005

Skripsi ini telah diuji dan dinyatakan lulus pada
3 Januari 2020
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Widhy Hayuhardhika N.P., S.Kom., M.Kom.	Buce Trias H., S.Kom., M.Kom.
NIK: 2017128704092001	NIP: 198904262019031009

Mengetahui
Ketua Jurusan Sistem Informasi

Dr. Eng. Herman Tolle, S.T., M.T.
NIP: 197408232000121001

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

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, 20 Desember 2019

METERAI TEMPEL
881C6ADF141894939
6000
ENAM RIBU RUPIAH

Yuninda Eka Nisrina

NIM: 165150707111005

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PRAKATA

Puji syukur kehadirat Allah Yang Maha Esa atas segala rahmat dan berkat yang telah diberikan kepada penulis sehingga dapat menyelesaikan penelitian yang berjudul "Pengembangan *E – Commerce* Dengan Pemanfaatan Sistem *Payment Gateway* (Studi Kasus: Wisata Kampung *Adventure*)". Penelitian ini sukses berkat dukungan, doa, dan bantuan dari banyak pihak. Penulis ingin menyampaikan rasa terima kasih kepada:

1. Bapak Widhy Hayuhardhika Nugraha Putra, S.Kom., M.Kom. selaku dosen pembimbing satu yang telah bersedia memfasilitasi, membimbing, dan mengarahkan penulis selama pengerjaan penelitian ini.
2. Bapak Buce Trias H., S.Kom., M.Kom. selaku dosen pembimbing dua yang telah bersedia membimbing dan mengarahkan penulis selama pengerjaan penelitian ini.
3. Bapak Issa Arwani, S.Kom., M.Sc. selaku Ketua Program Studi Sistem Informasi.
4. Bapak Dr. Eng. Herman Tolle, S.T., M.T. selaku Ketua Jurusan Sistem Informasi.
5. Ibunda dan Ayahanda saya senantiasa memberikan kasih saying, nasihat, semangat, dan doa yang tulus.
6. Bapak Aribowo selaku pengelola Wisata Kampung Sapi *Adventure* yang telah bekerja sama untuk menyukseskan penelitian ini.
7. Keluarga Besar Mahasiswa Teknologi Informasi (KBMTI) yang telah banyak memberi bantuan, hiburan, dan dukungan selama perkuliahan maupun pengerjaan penelitian ini.
8. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang telah memberikan saya dukungan, menjadi teman yang baik dan menyenangkan selama masa perkuliahan.

Semoga Allah Yang Maha Esa selalu memberikan rahmat serta hidayah-Nya kepada semua pihak yang telah terlibat dalam penelitian ini. Penulis menerima sara atau ide yang dapat digunakan sebagai referensi untuk penelitian selanjutnya. Penulis mengharapkan skripsi ini dapat bermanfaat bagi siapapun yang membutuhkannya.

Malang, 20 Desember 2019

Penulis

yunindaeka@student.ub.ac.id

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

ABSTRAK

Yuninda Eka Nisrina, Pengembangan *E – Commerce* Dengan Pemanfaatan Sistem *Payment Gateway* (Studi Kasus: Wisata Kampung Sapi *Adventure*)

Pembimbing: Widhy Hayuhardhika Nugraha Putra, S.Kom., M.Kom. dan Buce Trias H., S.Kom., M.Kom.

Wisata Kampung Sapi *Adventure* merupakan peternakan sapi yang dijadikan sebagai tempat objek wisata edukasi sekaligus mengembangkan konsep agrowisata. Wisata Kampung Sapi Adventure saat ini belum memiliki sistem informasi yang digunakan untuk media promosi, pemesanan tiket secara online, serta membuat laporan keuangan. Dikarenakan transaksi jual beli tiket dan reservasi melalui media sosial masih dilakukan secara manual yaitu wisatawan mengirimkan foto bukti pembayaran transfer. Kemudian petugas wisata akan menginputkan data transaksi pembelian tiket ke dalam *spreadsheet*. Proses manual ini dapat menyebabkan data tidak valid, karena petugas kurang teliti dalam menginputkan data. pada penelitian ini, akan dibangun aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* dengan memanfaatkan *payment gateway* midtrans untuk mempermudah pendataan wisatawan dalam hal transaksi jual beli tiket dan reservasi serta memudahkan petugas wisata dalam melakukan pencatatan transaksi penjualan tiket. Dalam membangun aplikasi ini, peneliti menggunakan konsep *Service Oriented Architecture*. Hasil dari penelitian yang telah dilakukan, diperoleh analisis kebutuhan, rancangan, implementasi, dan pengujian sistem. Metode pengujian yang digunakan dalam perancangan ini adalah *blackbox* dan *compatibility*. Hasil pengujian pada aplikasi ini menunjukkan adanya keselarasan dengan rancangan sistem serta aplikasi dapat dijalankan pada beberapa *browser*.

Kata kunci: *Wisata Kampung Sapi Adventure, Transaksi Jual Beli Tiket, Reservasi, E-Commerce, Payment Gateway Midtrans, Service Oriented Architecture.*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

ABSTRACT

Yuninda Eka Nisrina, *Development of E – Commerce With The Utilization of Payment Gateway System* (*Case Study*: *Kampung Sapi Adventure Tourism*)

***Supervisors*: Widhy Hayuhardhika Nugraha Putra, S.Kom., M.Kom. *and* Buce Trias H., S.Kom., M.Kom.**

Kampung Sapi Adventure Tourism is a cow farm that used as a place for educational tourism objects and develops the concept of agritourism. Currently it does not have information system used for promotional media, booking tickets online, and making financial reports. Buying and selling tickets and bookings through social media are still done manually, therefore, tourists send photos of proof of transfer payments.Then the tour officer will input the ticket purchase transaction data into spreadsheet. This manual process can cause invalid data, because officers are less careful about entering data. This research, an e-commerce application will be built by utilizing the midtrans payment gateway to facilitate the data collection of tourists in the case of buying and selling tickets and reservations as well as making it easier for tour officials to record ticket sales transaction. This researcher uses a Service Oriented Architecture.This research develop requirement analysis, design, implementation, and testing of the system. The testing methods uses are blackbox and compatibility. The test results on this application indicate alignment between system design and applications, and can be run on multiple browsers.

Keywords: *Kampung Sapi Adventure Tours, Ticket Sales Transactions, Reservations, E-Commerce, Payment Gateway Midtrans, Service Oriented Architecture.*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR ISI

PENGESAHAN ii
PERNYATAAN ORISINALITAS iii
PRAKATA iv
ABSTRAK v
ABSTRACT vi
DAFTAR ISI vii
DAFTAR TABEL xii
DAFTAR GAMBAR xiv
BAB 1 PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Tujuan 2
1.4 Manfaat 2
1.5 Batasan Masalah 3
1.6 Sistematika Pembahasan 3
BAB 2 TINJAUAN PUSTAKA 5
2.1 Wisata Kampung Sapi *Adventure* 5
2.1.1 Visi 5
2.1.2 Misi 5
2.2 Kajian Pustaka 6
2.3 Dasar Teori 6
2.3.1 Pengertian *Electronic Commerce* (*E-commerce*) 6
2.3.2 Proses Bisnis 8
2.3.3 *Business Process Modeling Notation* (BPMN) 8
2.3.4 *Framework CodeIgniter* 11
2.3.5 *Object Oriented Analysis and Design* (OOAD) 12
2.3.6 *Service Oriented Architecture* (SOA) 12
2.3.7 *Web Service* 14
2.3.8 JSON (*JavaScript Object Notation*) 15
2.3.9 *Representation State Transfer* (REST) 16
2.3.10 Midtrans 16

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

2.3.11 *Software Engineering* 16
2.3.12 *Unifed Model Language* (UML) 17
2.3.13 *System Development Live Cycle* (SDLC) 22
2.3.14 *Waterfall Model* 23
2.3.15 Pengujian *Black-box* 24
BAB 3 METODOLOGI PENELITIAN 25
3.1 Struktur Tahapan Penelitian 25
3.2 Studi Literatur 25
3.3 Analisis Kebutuhan 26
3.4 Perancangan Sistem 26
3.5 Implementasi Sistem 26
3.5.1 Implementasi *E-commerce* 26
3.5.2 Implementasi *Payment Gateway* 26
3.6 Pengujian 27
3.7 Kesimpulan dan Saran 27
BAB 4 ANALISIS KEBUTUHAN DAN PERANCANGAN SISTEM 28
4.1 Arsitektur Aplikasi *E – Commerce* Wisata Kampung Sapi *Adventure* 28
4.2 Pemodelan Proses Bisnis 29
4.2.1 Pemodelan Proses Bisnis *As-Is* 29
4.2.2 Proses Bisnis *To-Be* 35
4.3 Analisis Kebutuhan 38
4.3.1 Identifikasi Aktor 38
4.3.2 Kebutuhan Fungsional 39
4.3.3 Kebutuhan Non-Fungsional 41
4.3.4 Pemodelan Use Case 42
4.4 Spesifikasi *Use Case* 44
4.5 *Activity Diagram* 57
4.5.1 *Activity Diagram* Register 57
4.5.2 *Activity Diagram Login* 57
4.5.3 *Activity Diagram Logout* 59
4.5.4 *Activity Diagram* Melihat Kegiatan 60
4.5.5 *Activity Diagram* Melakukan Pemesanan Tiket 61

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

4.5.6 *Activity Diagram* Melihat Halaman *Dashboard* 62
4.5.7 *Activity Diagram* Melakukan Mengelola Data Kegiatan 63
4.5.8 *Activity Diagram* Mengelola Data Tiket 64
4.5.9 *Activity Diagram* Melihat Daftar Pemesanan 66
4.5.10 *Activity Diagram* Mengelola Data Daftar Pengguna 66
4.5.11 *Activity Diagram* Melihat Laporan Keuangan 67
4.5.12 *Activity Diagram* Mencetak Laporan Keuangan 68
4.6 *Sequence Diagram* 68
4.6.1 *Sequence Diagram* Register 69
4.6.2 *Sequence Diagram Login* 70
4.6.3 *Sequence Diagram Logout* 72
4.6.4 *Sequence Diagram* Melihat Kegiatan 73
4.6.5 *Sequence Diagram* Melakukan Pemesanan Tiket 73
4.6.6 *Sequence Diagram* Melihat Halaman *Dashboard* Admin 76
4.6.7 *Sequence Diagram* Mengelola Data Kegiatan 76
4.6.8 *Sequence Diagram* Mengelola Data Tiket 78
4.6.9 *Sequence Diagram* Melihat Daftar Pemesanan Tiket 80
4.6.10 *Sequence Diagram* Mengelola Data Daftar Pengguna 80
4.6.11 *Sequence Diagram* Melihat Laporan Keuangan 81
4.6.12 *Sequence Diagram* Mencetak Laporan Keuangan 82
4.7 Class Diagram 82
4.8 Physical Data Model 85
4.9 Perancangan Integrasi Layanan API *Sandbox* Pada *Payment Gateway* Midtrans 86
4.10 Perancangan Antarmuka Pengguna 87
4.10.1 Perancangan Antarmuka Halaman Pada Menu Kegiatan 87
4.10.2 Perancangan Antarmuka Halaman Menu Pesan Tiket 88
4.10.3 Perancangan Antarmuka Register 103
4.10.4 Perancangan Antarmuka *Login* 104
4.10.5 Perancangan Antarmuka Halaman *Logout* 105
4.10.6 Perancangan Antarmuka Halaman *Dashboard* Admin 105
4.10.7 Perancangan Antarmuka Halaman Menu Tiket 106
4.10.8 Perancangan Antarmuka Halaman Kegiatan 107

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

4.10.9 Perancangan Antarmuka Halaman Menu Daftar Pengguna 108
4.10.10 Perancangan Antarmuka Halaman Menu Daftar Pemesanan 109
4.10.11 Perancangan Antarmuka Halaman Menu Laporan Keuangan 110
BAB 5 IMPLEMENTASI DAN PENGUJIAN 112
5.1 Spesifikasi Lingkungan Implementasi 112
5.1.1 Spesifikasi Perangkat Keras 112
5.1.2 Spesifikasi Perangkat Lunak 112
5.2 Batasan Implementasi 113
5.3 Konfirgurasi Penggunaan Layanan API *Sandbox* Pada *Payment Gateway* Midtrans 113
5.4 Implementasi Kode Program 114
5.5 Implementasi Antarmuka 132
5.5.1 Implementasi Antarmuka Halaman Kegiatan 132
5.5.2 Implementasi Antarmuka Halaman Pesan Tiket 133
5.5.3 Implementasi Antarmuka Halaman Register 141
5.5.4 Implementasi Antarmuka Halaman *Login* 142
5.5.5 Implementasi Antarmuka Halaman *Logout* 142
5.5.6 Implementasi Antarmuka Halaman *Dashboard* Admin 143
5.5.7 Implementasi Antarmuka Halaman Menu Tiket 143
5.5.8 Implementasi Antarmuka Halaman Menu Kegiatan 144
5.5.9 Implementasi Antarmuka Halaman Menu Daftar Pengguna 144
5.5.10 Implementasi Antarmuka Halaman Menu Daftar Pemesanan 145
5.5.11 Implementasi Antarmuka Halaman Menu Laporan Keuangan 145
5.6 Pengujian 146
5.6.1 Pengujian Fungsional 146
5.6.2 Pengujian Non-Fungsional 156
BAB 6 PENUTUP 158
6.1 Kesimpulan 158

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

6.2 Saran 159
DAFTAR REFERENSI 160
LAMPIRAN A HASIL WAWANCARA 162
LAMPIRAN B HASIL PENGUJIAN VALIDASI 165

DAFTAR TABEL

Tabel 2.1 Simbol-simbol *Use Case Diagram* 18
Tabel 2.2 Simbol-simbol *Class Diagram* 19
Tabel 2.3 Simbol-simbol *Activity Diagram* 20
Tabel 2.4 Simbol-simbol *Sequence Diagram* 21
Tabel 2.5 Simbol-simbol *Sequence Diagram* 22
Tabel 4.1 Identifikasi Aktor 38
Tabel 4.2 Kebutuhan Fungsional Wisatawan 39
Tabel 4.3 Kebutuhan Fungsional Admin 40
Tabel 4.4 Kebutuhan Non-Fungsional 41
Tabel 4.5 *Use Case Diagram* Aplikasi *E-commerce* Wisata Kampung Sapi *Adventure* 43
Tabel 4.6 Spesifikasi *Use Case* Melakukan Register 44
Tabel 4.7 Spesifikasi *Use Case* Melakukan Login 45
Tabel 4.8 Spesifikasi *Use Case* Melakukan *Logout* 46
Tabel 4.9 Spesifikasi *Use Case* Melihat Kegiatan 47
Tabel 4.10 Spesifikasi *Use Case* Melakukan Pemesanan Tiket 48
Tabel 4.11 Spesifikasi *Use Case Login* 50
Tabel 4.12 Spesifikasi *Use Case* Melakukan *Logout* 51
Tabel 4.13 Spesifikasi *Use Case* Melihat Halaman *Dashboard* Admin 51
Tabel 4.14 Spesifikasi *Use Case* Mengelola Data Kegiatan 52
Tabel 4.15 Spesifikasi *Use Case* Mengelola Data Tiket 53
Tabel 4.16 Spesifikasi *Use Case* Melihat Daftar Pemesanan 54
Tabel 4.17 Spesifikasi *Use Case* Mengelola Data Daftar Pengguna 54
Tabel 4.18 Spesifikasi *Use Case* Melihat Laporan Keuangan 55
Tabel 4.19 Spesifikasi *Use Case* Mencetak Laporan Keuangan 56
Tabel 5.1 Spesifikasi Perangkat Keras Komputer 112
Tabel 5.2 Spesifikasi Perangkat Lunak Komputer 112
Tabel 5.3 Kode Program *Payment Gateway* Midtrans 115
Tabel 5.4 Kode Program Register, Login, dan Logout 115
Tabel 5.5 Kode Program Kegiatan 119
Tabel 5.6 Kode Program Pemesanan Tiket 120

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 5.7 Kode Program Pada Halaman Admin 124
Tabel 5.8 Kode Program Cek Status Pembayaran Tiket 131
Tabel 5.9 Kasus Uji Validasi Melakukan Register 147
Tabel 5.10 Kasus Uji Validasi Melakukan *Login* Pada Wisatawan 147
Tabel 5.11 Kasus Uji Validasi Melakukan *Logout* Pada Wisatawan 148
Tabel 5.12 Kasus Uji Validasi Melihat Kegiatan Pada Wisatawan 149
Tabel 5.13 Kasus Uji Validasi Melakukan Pemesanan Tiket 149
Tabel 5.14 Kasus Uji Validasi *Login* Pada Admin 150
Tabel 5.15 Kasus Uji Validasi *Logout* Pada Admin 151
Tabel 5.16 Kasus Uji Melihat Halaman *Dashboard* Admin 152
Tabel 5.17 Kasus Uji Mengelola Data Tiket 152
Tabel 5.18 Kasus Uji Mengelola Data Kegiatan 153
Tabel 5.19 Kasus Uji Melihat Daftar Pemesanan Tiket 154
Tabel 5.20 Kasus Uji Mengelola Data Daftar Pengguna 154
Tabel 5.21 Kasus Uji Melihat Laporan Keuangan 155
Tabel 5.22 Kasus Uji Mencetak Laporan Keuangan 155
Tabel 5.23 Kasus Uji Mencetak Laporan Keuangan 156

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR GAMBAR

Gambar 2.1 Logo Wisata Kampung Sapi *Adventure* 5
Gambar 2.2 Komponen *E-commerce* 7
Gambar 2.3 *Type Event* pada BPMN 9
Gambar 2.4 *Activity* pada BPMN 9
Gambar 2.5 *Type Gateway* pada BPMN 10
Gambar 2.6 *Connecting Object* pada BPMN 10
Gambar 2.7 *Artefacts* pada BPMN 11
Gambar 2.8 *Swimlanes* pada BPMN 11
Gambar 2.9 Arsitektur dari *Service Interface* dan *Service Implementation* 12
Gambar 2.10 Enkapsulasi *Logic* dalam *Service* 13
Gambar 2.11 *Layering* pada SOA 14
Gambar 2.12 Operasi *Web Service* 15
Gambar 2.13 Penulisan JSON 16
Gambar 2.14 Tingkatan Pada Rekayasa Perangkat Lunak 17
Gambar 2.15 *Waterfall Model* 23
Gambar 3.1 Diagram Alir Metode Penelitian Aplikasi *E-commerce* Wisata Kampung Sapi *Adventure* 25
Gambar 4.1 Arsitektur Aplikasi *E – Commerce* Wisata Kampung Sapi *Adventure* 28
Gambar 4.2 Proses Bisnis Saat Ini Pada Pemesanan Tiket Wisata Kampung Sapi Adventure Melalui Via Whatsapp 32
Gambar 4.3 Proses Bisnis Saait Ini Pada Pemesanan Tiket Wisata Kampung Sapi Adventure Secara Langsung Ke Loket Tiket 34
Gambar 4.4 Proses Bisnis Usulan Pemesanan Tiket Kampung Sapi *Adventure* 37
Gambar 4.5 Aturan Penomoran Identifikasi Aktor 38
Gambar 4.6 Aturan Penomoran Pada Kebutuhan Fungsional 39
Gambar 4.7 Aturan Penomoran Kebutuhan Non-Fungsional 41
Gambar 4.8 Aturan Penomoran Pada *Use Case Diagram* 42
Gambar 4.9 *Use Case Diagram* Aplikasi *E-commerce* Wisata Kampung Sapi *Adventure* 43
Gambar 4.10 *Activity Diagram* Register 57
Gambar 4.11 *Activity Diagram Login* Pada Wisatawan 58
Gambar 4.12 *Activity Diagram Login* Pada Admin 59

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Gambar 4.13 *Activity Diagram Logout* Pada Wisatawan Dan Admin 60
Gambar 4.14 *Activity Diagram* Melihat Kegiatan 60
Gambar 4.15 *Activity Diagram* Melakukan Pemesanan Tiket 61
Gambar 4.16 *Activity Diagram* Melihat Halaman *Dashboard* 62
Gambar 4.17 *Activity Diagram* Melakukan Mengelola Data Kegiatan 63
Gambar 4.18 *Activity Diagram* Mengelola Data Tiket 65
Gambar 4.19 *Activity Diagram* Melihat Daftar Pemesanan 66
Gambar 4.20 *Activity Diagram* Mengelola Data Daftar Pengguna 67
Gambar 4.21 *Activity Diagram* Melihat Laporan Keuangan 67
Gambar 4.22 Activity Diagram Mencetak Laporan Keuangan 68
Gambar 4.23 Sequence Diagram Register 69
Gambar 4.24 *Sequence Diagram Login* Pada Wisatawan 70
Gambar 4.25 *Sequence Diagram Login* Pada Admin 71
Gambar 4.26 *Sequence Diagram Logout* Pada Wisatawan 72
Gambar 4.27 *Sequence Diagram Logout* Pada Admin 72
Gambar 4.28 *Sequence Diagram* Melihat Kegiatan 73
Gambar 4.29 *Sequence Diagram* Melakukan Pemesanan Tiket 75
Gambar 4.30 Sequence Diagram Melihat Halaman *Dashboard* Admin 76
Gambar 4.31 *Sequence Diagram* Mengolah Data Kegiatan 77
Gambar 4.32 *Sequence Diagram* Mengelola Data Tiket 79
Gambar 4.33 *Sequence Diagram* Melihat Daftar Pemesanan 80
Gambar 4.34 *Sequence Diagram* Mengubah Data Daftar Pengguna 81
Gambar 4.35 *Sequence Diagram* Melihat Laporan Keuangan 81
Gambar 4.36 *Sequence Diagram* Mencetak Laporan Keuangan 82
Gambar 4.37 Class Diagram Aplikasi *E-Commerce* Wisata Kampung Sapi *Adventure* 84
Gambar 4.38 *Physical Data Model* Aplikasi *E-Commerce* Wisata Kampung Sapi *Adventure* 85
Gambar 4.39 Perancangan Integrasi API *Sandbox* Pada *Payment Gateway* Midtrans 86
Gambar 4.40 Rancangan Antarmuka Halaman Pada Menu Kegiatan 87
Gambar 4.41 Rancangan Antamuka Halaman Memilih Kategori tiket 88
Gambar 4.42 Rancangaan Antarmuka Halaman Informasi Paket Tiket Go Show 89

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Gambar 4.43 Rancangan Antarmuka Halaman Kategori Tiket Reservasi 90
Gambar 4.44 Perancangan Antarmuka Informasi Paket Tiket Reservasi 92
Gambar 4.45 Rancangan Antarmuka Halaman Form Pemesanan Tiket 93
Gambar 4.46 Perancangan Antarmuka Halaman Detail Tiket 94
Gambar 4.47 Perancangan Antarmuka Halaman Pembayaran Pada Midtrans ... 95
Gambar 4.48 Perancangan Antarmuka Halaman Memilih Metode Pembayaran Pada Midtrans 96
Gambar 4.49 Perancangan Antarmuka Halaman Metode Pembayaran Menggunakan ATM/Bank Transfer Pada Midtras 97
Gambar 4.50 Perancangan Antarmuka Halaman Cara Pembayaran Melalui Permata Pada Midtrans 98
Gambar 4.51 Perancangan Antarmuka Halaman Cara Pembayaran Melalui Permata Bank Pada Midtrans 99
Gambar 4.52 Perancangan Antarmuka Halaman Metode Pembayaran Melalui Permata Bank Pada Midtrans 100
Gambar 4.53 Implementasi Antarmuka Halaman Pada Menu Tiket 101
Gambar 4.54 Perancangan Antarmuka Register 103
Gambar 4.55 Perancangan Antarmuka *Login* 104
Gambar 4.56 Perancangan Antarmuka Halaman *Logout* Pada Wisatawan 105
Gambar 4.57 Perancangan Antarmuka Halaman *Dashboard* Admin 106
Gambar 4.58 Perancangan Antarmuka Halaman Menu Tiket 107
Gambar 4.59 Perancangan Antarmuka Halaman Kegiatan 108
Gambar 4.60 Perancangan Antarmuka Halaman Daftar Pengguna 109
Gambar 4.61 Perancangan Antarmuka Halaman Daftar Pemesanan 110
Gambar 4.62 Perancangan Antarmuka Halaman Menu Laporan Keuangan 111
Gambar 5.1 *Installasi Composer* 113
Gambar 5.2 *Library Midtrans* 114
Gambar 5.3 Konfigurasi *Composer Autoload* 114
Gambar 5.4 Implementasi Antarmuka Halaman Kegiatan Aplikasi *E-commerce* Wisata Kampung Sapi *Adventure* 133
Gambar 5.5 Implementasi Antamuka Halaman Pesan Tiket Aplikasi *E-commerce* Wisata Kampung Sapi *Adventure* 133
Gambar 5.6 Implementasi Antarmuka Halaman Informasi Paket Tiket Go Show 134
Gambar 5.7 Implementasi Antarmuka Halaman Kategori Tiket Reservasi 135

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Gambar 5.8 Implementasi Antarmuka Informasi Paket Tiket Reservasi 136
Gambar 5.9 Implementasi Antarmuka Halaman Form Pemesanan Tiket 136
Gambar 5.10 Implementasi Antarmuka Halaman Detail Tiket 137
Gambar 5.11 Implementasi Antarmuka Halaman Pembayaran Pada Midtrans 137
Gambar 5.12 Implementasi Antarmuka Halaman Memilih Metode Pembayaran Pada Midtrans 138
Gambar 5.13 Implementasi Antarmuka Halaman Metode Pembayaran Menggunakan ATM/Bank Transfer Pada Midtrans 139
Gambar 5.14 Implementasi Antarmuka Halaman Cara Pembayaran Melalui Permata Pada Midtrans 140
Gambar 5.15 Implementasi Antarmuka Halaman Metode Pembayaran Melalui Permata Bank 140
Gambar 5.16 Implementasi Antarmuka Halaman Pada Menu Tiket 141
Gambar 5.17 Implementasi Antarmuka Halaman Register 142
Gambar 5.18 Implementasi Antarmuka Halaman *Login* 142
Gambar 5.19 Implementasi Antarmuka Halaman *Logout* 143
Gambar 5.20 Implementasi Antarmuka Halaman *Dashboard* Admin 143
Gambar 5.21 Implementasi Antarmuka Halaman Menu Tiket 144
Gambar 5.22 Implementasi Antarmuka Halaman Menu Kegiatan 144
Gambar 5.23 Implementasi Antarmuka Halaman Menu Daftar Pengguna 145
Gambar 5.24 Implementasi Antarmuka Halaman Menu Daftar Pemesanan 145
Gambar 5.25 Implementasi Antarmuka Halaman Menu Laporan Keuangan 146
Gambar 5.26 Aturan Penomoran Pengujian *Black – Box* 146
Gambar 5.27 Hasil Uji *Compatibility* Menggunakan *Software Sortsite* 156
Gambar 5.28 Petunjuk Adanya Masalah Pada *Critical Issues* 157
Gambar 5.29 Petunjuk Adanya Masalah Pada *Mayor Issues* 157
Gambar 5.30 Petunjuk Adanya Masalah *Minor Issues* 157

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Electronic Commerce (*E-commerce*) merupakan sebuah bisnis online yang mengacu pada penggunaan sebuah teknologi secara elektronik untuk melakukan perdagangan yang dapat berjalan dengan melalui jaringan internet. Penelitian yang dilakukan sebelumnya oleh (Mukherjee & Roy, 2017) tentang *E-Commerce and Online Payment in the Modern Area* menyatakan bahwa *E-commerce* adalah salah satu jenis metode perdagangan saat ini yang digunakan pedagang untuk berinteraksi dengan pelanggan dengan cepat dan sistem pembayaran secara elektronik yang dirancang dengan baik dapat meberikan keamanan yang lebih baik daripada sistem pembayaran tradisional. *E-commerce* saat ini tidak hanya bergerak pada transaksi jual beli barang, namun juga merambah pada sektor pariwisata seperti pemesanan tiket wisata, akomodasi perjalanan wisata, dan pemesanan hotel. Dalam menghadapi tingginya persaingan dunia *e-commerce* pada era saat ini dibutuhkan sebuah metode pembayaran yang dapat memudahkan pengguna dalam melakukan transaksi pembayaran yaitu dengan *payment gateway*. *Payment gateway* adalah sebuah sistem penyedia layanan *e-commerce* yang bertindak sebagai jembatan antara situs *web* pedagang dan lembaga keuangan yang memproses transaksi secara *online*. *Payment gateway* memberikan layanan transaksi secara mudah, efisien, dan aman. Salah satu layanan *payment gateway* yang sering digunakan adalah midtrans. Midtrans adalah sistem pembayaran yang dapat memfasilitasi penjual dan pembeli untuk melakukan transaksi ((Midtrans (PT Midtrans), 2019). Pengimplementasian kedua aplikasi dengan *platform* ini diperlukan sebuah integrasi data menggunakan *web service. Web service* merupakan sebuah teknologi untuk merealisasikan data dari integrasi aplikasi yang ada. Terdapat sebuah teknologi yang menjadi arsitektur penyedia *web service* yaitu *Service Oriented Architecture* (SOA). SOA adalah arsitektur yang digunakan untuk membangun aplikasi yang memanfaatan teknologi *web service* (Aditama, 2017).

Kota Batu adalah salah satu kota wisata yang memiliki beragam konsep wisata salah satunya wisata edukasi. Pemerintah Kota Batu berupaya meningkatkan perekonomian masyarakat dan jumlah pariwisatawan dengan cara mempromosikan objek wisata yang ada di Batu secara *online* seperti, media sosial dan *website*. Akan tetapi, masih terdapat beberapa objek wisata di Kota Batu yang belum diketahui wisatawan, salah satunya Kampung Sapi *Adventure*. Wisata Kampung Sapi *Adventure* merupakan peternakan sapi yang dijadikan sebagai tempat objek wisata edukasi sekaligus mengembangkan konsep agrowisata. Menurut wawancara dengan Aribowo (Aribowo, 2019) selaku pengelola Wisata Kampung Sapi *Adventure*, pempromosian dan penjualan tiket wisata melalui media sosial Facebook, Instagram, dan Whatsapp. Namun upaya tersebut belum efektif, khususnya pada transaksi jual beli tiket dan reservasi. Dikarenakan transaksi jual beli tiket yang dibeli di hari yang sama atau disebut dengan tiket *go show* maupun reservasi melalui media sosial masih dilakukan

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

secara manual yaitu wisatawan mengirimkan foto bukti pembayaran transfer. Kemudian petugas wisata akan menginputkan data transaksi pembelian tiket ke dalam aplikasi pemrosesan *spreadsheet*. Proses manual ini dapat menyebabkan data tidak valid karena petugas kurang teliti dalam menginputkan data. Selain itu, transaksi tersebut membutuhkan waktu yang cukup lama. Agar transaksi dapat berjalan dengan efektif dan aman, maka pengelola Wisata Kampung Sapi *Adventure* menginginkan adanya jaminan transaksi yang baik.

Berdasarkan uraian permasalahan di atas, maka pengelola Wisata Kampung Sapi *Adventure* menginginkan pengembangan aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* dengan memanfaatkan pembayaran secara *online* melalui *payment gateway* midtrans untuk mempermudah pendataan wisatawan dalam hal transaksi jual beli tiket dan reservasi serta memudahkan petugas wisata dalam melakukan pencatatan transaksi penjualan tiket. Pengembangan aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* dikembangkan menggunakan konsep SOA dengan memanfaatkan layanan sistem pembayaran midtrans.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, dapat duraikan permasalahan rumusan masalah sebagai berikut :

1. Bagaimana analisis dan perancangan aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* dengan memanfaatkan layanan *payment gateway* midtrans?
2. Bagaimana implementasi aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* dengan memanfaatkan layanan *payment gateway* midtrans?
3. Bagaimana pengujian aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* dengan memanfaatkan layanan *payment gateway* midtrans?

1.3 Tujuan

Tujuan penelitian ini dibuat berdasarkan rumusan masalah sebagai berikut:

1. Menghasilkan analisis dan rancangan aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* dengan memanfaatkan layanan *payment gateway* midtrans.
2. Menghasilkan implementasi aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* dengan memanfaatkan layanan *payment gateway* midtrans.
3. Mendapatkan hasil pengujian aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* dengan memanfaatkan layanan *payment gateway* midtrans.

1.4 Manfaat

Manfaat yang didapat dari pengembangan aplikasi ini adalah sebagai berikut:

1. Memberikan informasi kepada calon wisatawan gambaran mengenai informasi Wisata Kampung Sapi *Adventure*.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

2. Mempermudah wisatawan dalam melakukan transaksi pembelian dan reservasi tiket.
3. Memberikan kemudahan bagi pengelola Wisata Kampung Sapi *Adventure* dalam membuat laporan keuangan.

1.5 Batasan Masalah

Pada penelitian ini memiliki beberapa batasan masalah agar dalam implementasinya lebih terarah:

1. Penelitian ini melakukan analisis, desain, implementasi, dan pengujian pada aplikasi *e-commerce* Wisata Kampung Sapi *Adventure*.
2. Aplikasi ini ditujukan untuk digunakan oleh pengelola dan wisatawan Kampung Sapi *Adventure*.
3. Aplikasi e-commerce Wisata Kampung Sapi *Adventure* diimplementasikan dengan *framework CodeIgniter* dan bahasa pemrograman PHP.
4. Integrasi *payment gateway* dikelola dengan memanfaatkan layanan API *sandbox* dari midtrans.

1.6 Sistematika Pembahasan

Sistematika pembahasan penelitian ini terdiri dari enam bab adalah sebagai berikut:

BAB 1 : PENDAHULUAN

Bab ini menjelaskan tentang latar belakang mengenai diangkatnya suatu permasalahan, rumusan masalah, tujuan dari penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan.

BAB 2 : TINJAUAN PUSTAKA

Bab ini menjelaskan tentang pembahasan teori – teori yang dijadikan untuk referensi penelitian.

BAB 3 : METODOLOGI PENELITIAN

Bab ini menjelaskan tentang tahapan - tahapan yang dibutuhkan selama melakukan analisis dan pengembangan aplikasi *e-commerce* Wisata Kampung Sapi *Adventure.*

BAB 4: ANALISIS KEBUTUHAN DAN PERANCANGAN SISTEM

Bab ini menjelaskan tentang analisis kebutuhan serta perancangan aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* sesuai dengan teori yang ada serta pembahasan tentang implementasi dari sistem.

BAB 5: IMPLEMENTASI DAN PENGUJIAN

Bab ini menjelaskan tentang penjelasan spesifikasi lingkungan sistem dan hasil pengujian terhadap sistem yang telah dibuat untuk mengetahui apakah sistem sudah sesuai dengan kebutuhan sistem.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB 6: KESIMPULAN DAN SARAN

Bab ini akan menjelaskan tentang kesimpulan dari penelitian yang telah dilakukan dan saran untuk digunakan penelitian selanjutnya.

BAB 2 TINJAUAN PUSTAKA

2.1 Wisata Kampung Sapi *Adventure*

Wisata Kampung Sapi *Adventure* merupakan salah satu wisata edukasi susu dan sapi perah yang terletak pada wilayah Kota Wisata Batu tepatnya di Jl Makam Desa Beji, Kec. Junrejo. Wisata Kampung sapi mampu menampung 350 pengunjung per harinya. Wisatawan Kampung Sapi *Adventure* terdiri dari tiga kategori pengunjung yaitu grup, umum, dan pencari inspirasi bisnis. Pada tiap-tiap kategori memiliki karakteritik antara lain wisatawan dengan kategori grup yang berasal dari kelompok bermain sampai SMP, wisatawan kategori umum tidak memiliki batasan dan tidak terpaut umur, dan wisatawan kategori pencari inspirasi bisnis berasal dari pelajar SMA sampai pensiunan. Wisata Kampung Sapi *Adventure* selain dijadikan objek wisata, wisata tersebut juga sebagai tempat pembelajaran bagi peternak sapi perah untuk mengetahui cara beternak sapi perah dan menghasilkan susu perah yang berkualitas. Pada Gambar 2.1 merupakan logo dari Wisata Kampung Sapi *Adventure.*

Gambar 2.1 Logo Wisata Kampung Sapi *Adventure*

Sumber: (Aribowo, 2019)

2.1.1 Visi

Menyejahterakan peternak khususnya peternak sapi perah.

2.1.2 Misi

Ikut andil dalam sektor agrowisata dalam mendukung daya tarik Kota Batu sebagai kota wisata.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

2.2 Kajian Pustaka

Pada penelitian ini didukung dengan beberapa referensi terkait yang telah dilakukan sebelumnya dengan topik dan bahasan yang relevan. Pada penelitian ini terdapat dua referensi jurnal ilmiah yang digunakan sebagai acuan dalam melaksanakan penelitian ini. Referensi pertama ditulis oleh (Marthasari, et al., 2010) yang berjudul "Implementasi *Web Service* Untuk Mendukung Interoperabilitas Pada Aplikasi E-Commerce". *E-commerce* merupakan aplikasi yang digunakan untuk melakukan transaksi secara online dan dapat dijalankan ketika aplikasi terhubung dengan internet. Namun, saat ini dengan beraneka ragam jenis hardware hingga bahasa pemrograman dapat memicu terjadinya permasalahan dalam proses pertukaran data antar aplikasi dengan *platform* yang berbeda. Dengan memanfaatkan teknologi *web service* permasalahan dalam proses pertukaran data antar aplikasi dengan *platform* yang berbeda dapat teratasi. Oleh karena itu, dalam pengimplementasian aplikasi *e-commerce* diperlukan *web service* untuk memudahkan distributor maupun agen dalam melakukan pekerjaannya.

Penelitian kedua diambil dari jurnal sebelumnya yang ditulis oleh (Pattnaik, et al., 2005) yang berjudul "*The Working Model Of An E-Payment System*". Pada penelitian ini penulis mengusulkan sistem pembayaran melalui internet menggunakan *payment gateway* untuk menangani transaksi pembayaran kartu kredit antara pelanggan, pedagang dan bank. Untuk menguji dan mengevaluasi sistem pembayaran secara *online*, penulis membangun aplikasi agen perjalanan *online* yang disebut E-Travel, yang mensimulasikan *E-Payment* pada kehidupan nyata. Layanan perjalanan *online* termasuk reservasi penerbangan, pemandu wisata, dan reservasi hotel yang disediakan di E-Travel. E-Travel memanfaatkan sistem pembayaran yang diusulkan yaitu dengan *payment gateway* untuk menangani pembayaran secara transfer antara pelanggan dan pedagang. Penulis menyatakan dari evaluasi yang telah dilakukan menunjukkan bahwa sistem pembayaran *gateway* dapat digunakan dengan mudah, aman, dan hemat biaya.

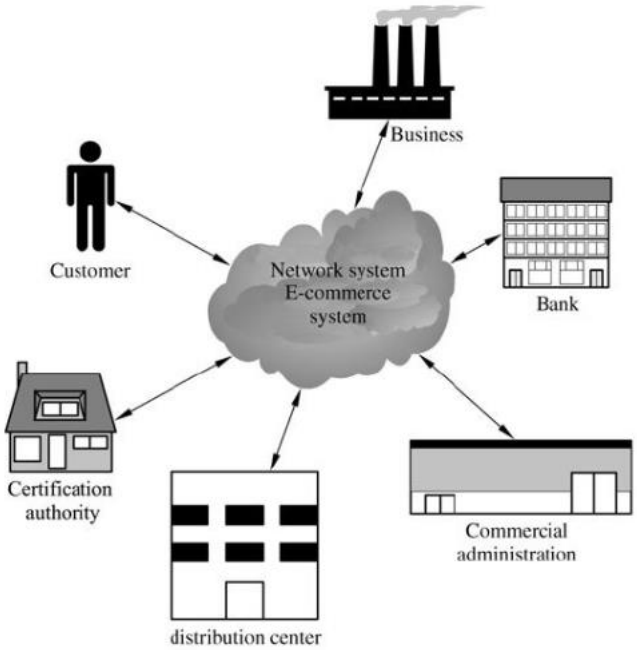
2.3 Dasar Teori

2.3.1 Pengertian *Electronic Commerce* (*E-commerce*)

Electronic Commerce (*E-commerce*) merupakan sebuah bisnis online yang mengacu pada penggunaan sebuah teknologi secara elektronik untuk melakukan perdagangan yang dapat berjalan dengan melalui jaringan internet. Dengan menggunakan internet penyebaran dan pertukaran infromasi dapat tersebar secara luas dengan waktu dan tenaga yang lebih sedikit daripada umumnya. Oleh karena itu, strategi bisnis secara online melalui sistem aplikasi *e-commerce* sangat efisien dan efektif. Pelaku bisnis dapat melakukan pertukaranan data dan informasi dengan perusahaan lain tanpa harus bertatap muka secara langsung. Agar penjual dan pelanggan merasa aman dan nyaman maka, pembayaran dapat dilakukan menggunakan kartu kredit, karena terdapat layanan untuk verifikasi

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

terkait keaslian kartu kredit yang digunakan oleh pembeli untuk meyakinkan kartu kreditnya sah, kemudian jika terbukti valid akan diproses oleh pihak bank yang mengeluarkan kartu kredit tersebut untuk mentransfer uang sesuai dengan pembelian ke rekening bank penjual (Sarwono & Prihartono, 2012). Saat ini terdapat beberapa bentuk *e-commerce* seperti toko *online*, internet banking, pemesanan tiket pesawat, reservasi penginapan hotel sampai reservasi ditempat wisata. *E-commerce* memiliki beberapa komponen utama seperti yang terdapat pada Gambar 2.2 (Qin, 2009).



Gambar 2.2 Komponen *E-commerce*

Sumber: (Qin, 2009)

Gambarr 2.2 adalah penjelasan komponen utama pada sebuah *e-commerce*.

1. Jaringan

Jaringan yang dimaksud yaitu internet, intranet, dan extranet. Internet adalah jaringan dasar dari *e-commerce* yang berfungsi untuk membawa informasi bisnis komersial. Adapun intranet yaitu jaringan yang digunakan bagi perusahaan untuk melakukan urusan internal. Berkenaan dengan Extranet adalah jaringan yang digunakan untuk menghubungkan antara perusahaan dan pengguna untuk melakukan kegiatan komersial.

2. Pengguna *E-commerce*

Pengguna *e-commerce* dapat dibedakan menjadi konsumen pribadi dan konsumen bisnis. Konsumen bisnis bertugas untuk mengelola staf, penghasilan, barang, produksi, pasokan, dan penjualan melalui internet, extranet, dan

Management Information System (MIS). Konsumen pribadi memiliki akses ke informasi dan membeli barang dengan menghubungkan internet dengan *browser*.

3. Otoritas Otentikasi

Otoritas Otentikasi (CA) yang diakui oleh hukum, bertanggung jawab untuk menerbitkan, mengelola sertifikat digital dan memfasilitasi pihak-pihak yang terlibat dalam penjualan *online* untuk mengidentifikasi satu sama lain.

4. Pusat Distribusi

Pusat distribusi bertanggung jawab atas pengiriman barang yang tidak dapat dikirim secara *online* ke konsumen dan melacak *traffic* barang.

5. *Online* Bank

Online bank memiliki fungsi untuk memberi layanan bagi penjual dan pembeli bisnis bank tradisional, seperti penyelesaian transaksi, dan layanan sepanjang waktu.

6. Administrasi kegiatan komersial

Administrasi kegiatan komersial terutama terdiri dari departemen industri, bea cukai, pajak dan perdagangan.

2.3.2 Proses Bisnis

Proses bisnis adalah kegiatan yang dilakukan oleh sebuah organisasi dengan alur yang runtut. Dalam menganalisis proses bisnis pada sebuah organisasi dapat dilakukan dengan dua pendekatan, yaitu membuat pemodelan proses bisnis *as – is* dan pemodelan proses bisnis *to – be*. Pemodelan proses bisnis *as – is* menggambarkan kondisi saat ini yang terjadi pada sebuah perusahaan. Sedangkan proses bisnis *to – be* dibuat berdasarkan proses bisnis *as – is* untuk memperbaiki beberapa aktivitas yang masih dilakukan secara manual.

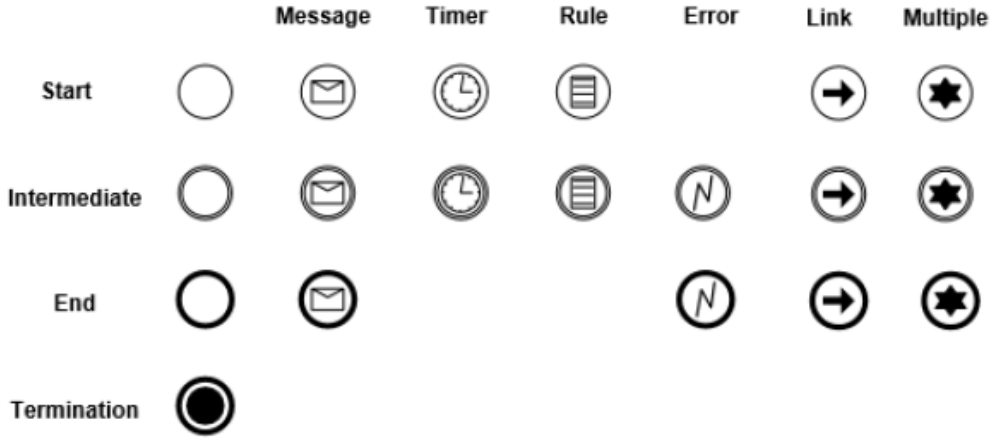
2.3.3 *Business Process Modeling Notation* (BPMN)

Business Process Modeling Notation (BPMN) adalah proses pemodelan dalam konteks *Service-Oriented Architecture* (SOA) dan komposisi layanan (Notation, 2019). BPMN dapat menggambarkan interaksi proses bisnis antar beberapa organisasi. BPMN berfokus pada beragam tingkatan abstraksi dari tingkatan bisnis hingga ke yang lebih teknis (teknologi perangkat lunak). Pada BPMN terdapat notasi grafis yang menggabungkan kemudahan dan kenyamanan penotasian dengan makna yang jelas. Berikut adalah notasi yang digunakan pada BPMN:

a. Flow Object

Flow object memiliki tiga komponen utama, yaitu *activity*, *event,* dan *gateway*. *Event* merupakan penghubung suatu kondisi yang terjadi pada organisasi dengan proses bisnisnya. Terdapat tiga tipe *event* pada pemodelan proses bisnis, yaitu *end event*, *start event,* dan *intermediate event*. *Start event*

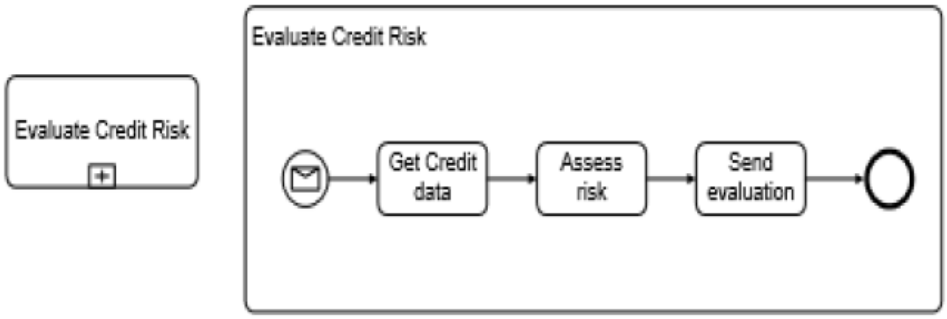
digunakan untuk menunjukkan letak dimana sebuah proses akan dimulai. *Intermediate event* digunakan untuk mengindikasi terjadinya sebuah *event* diantara awal dan akhir sebuah proses. *End event* digunakan ketika sebuah proses telah selesai. Gambar 2.3 adalah penjelasan simbol – simbol yang terdapat pada *type event*.



Gambar 2.3 *Type Event* pada BPMN

Sumber: (Weske, 2007)

Activity merupakan inti dalam suatu proses. *Activity* digunakan ketika pekerjaan dalam proses tidak mengalami kerusakan. Gambar 2.4 adalah penjelasan simbol – simbol yang terdapat pada *activity*.

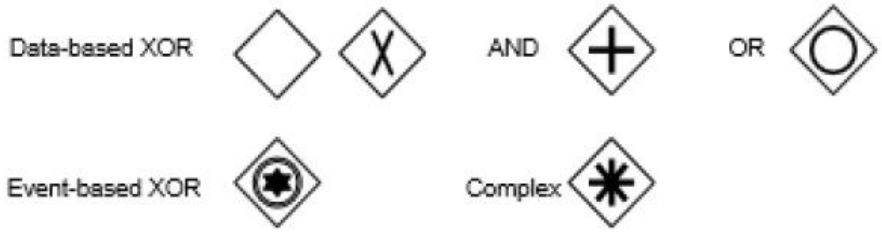


Gambar 2.4 *Activity* pada BPMN

Sumber: (Weske, 2007)

Gateway dapat berperan menjadi *join node* maupun *split node*. *Gateway* akan berperan menjadi *join node* ketika terdapat dua panah masuk dan satu panah keluar dan berlaku sebaliknya pada *split node*. Pada *event-based* xor *gateway* terdapat beberapa aktivitas yang diaktifkan dan siap untuk menerima pesan pada saat yang sama. Pada *data-based xor gateway,* keputusan dibuat oleh *gateway*. Pada AND *gateway* digunakan ketika suatu proses dapat

melakukan beberapa cabang operasi secara paralel. Pada OR *gateway* adalah terdapat dua panah masuk atau lebih dan terdapat lebih dari satu kemungkinan hasil. Gambar 2.5 adalah penjelasan simbol – simbol yang terdapat pada *gateway*.

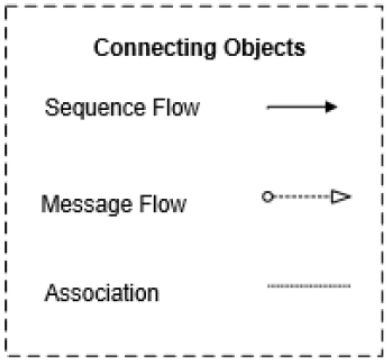


Gambar 2.5 *Type Gateway* pada BPMN

Sumber: (Weske, 2007)

b. *Connecting Objects*

Connecting Object dapat dibagai menjadi tiga tipe, yaitu *association*, *sequence flow*, dan *message flow*. *Sequence flow* digunakan untuk menunjukkan urutan kegiatan yang akan dilakukan dalam suatu proses. *Message flow* digunakan untuk menggambarkan alur pengiriman dan penerimaan pesan antar kegiatan yang terjadi dalam satu proses. *Association* digunakan ketika memerlukan *artefact* dan *flow object* untuk menambahkan informasi pada suatu kegiatan. Gambar 2.6 adalah penjelasan simbol – simbol yang terdapat pada *connection objects*.



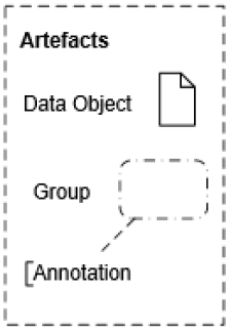
Gambar 2.6 *Connecting Object* pada BPMN

Sumber: (Weske, 2007)

c. *Artefact*

Artefact dapat dikategorikan menjadi tiga tipe, yaitu *group, data object,* dan *annotation*. *Data object* berfungsi memberikan informasi selama proses berlangsung tentang bagaimana menggunakan dan memperbarui dokumen, data, dan objek lain. *Group* adalah artefak yang menyediakan mekanisme visual untuk mengelompokkan elemen-elemen diagram secara tidak resmi. *Annotation* adalah informasi tambahan yang dibuat oleh pemodel untuk pembaca diagram

BPMN. Gambar 2.7 merupakan penjelasan simbol – simbol yang terdapat pada *artefacts*.

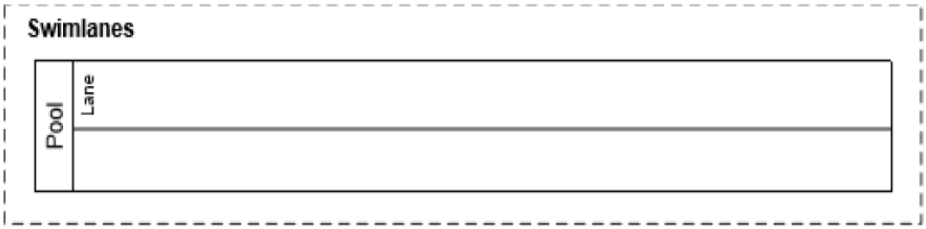


Gambar 2.7 *Artefacts* pada BPMN

Sumber: (Weske, 2007)

d. Swimlanes

Swimlane berfungsi untuk menetapkan proses atau bagian proses secara grafis ke entitas organisasi yang terlibat pada proses bisnis. *Swimlane* memuat *pool* dan *lane*. *Pool* mewakili peserta proses tertentu, atau peran entitas bisnis. *Lane* digunakan untuk mewakili entitas organisasi di dalam *pool*. Gambar 2.8 adalah penjelasan simbol – simbol yang terdapat pada *swimlanes*.



Gambar 2.8 *Swimlanes* pada BPMN

Sumber: (Weske, 2007)

2.3.4 *Framework CodeIgniter*

Menurut (Griffiths, 2010), *CodeIgniter* merupakan salah satu kerangka kerja bersifat *open source* yang dapat digunakan untuk membangun aplikasi berbasis web dan diimplementasikan dengan bahasa pemrograman PHP (*Hypertext Preprocessor*). *CodeIgniter* menyediakan dokumentasi *library* yang lengkap untuk membantu pengembang dalam membangun sebuah aplikasi. Konsep yang digunakan pada *framework CodeIgniter* adalah MVC (*Model-View-Controller*). MVC adalah arsitektur yang menghasilkan struktur kode yang lebih terstruktur dengan memisahkan representasi informasi dari proses pengolahan data, dan logika interaksi pengguna. Model pada konsep MVC bertanggung jawab untuk mengelola data. *View* bertanggung jawab untuk menampilkan data yang disediakan oleh model sesuai dengan permintaan pengguna. Sedangkan *controller* bertugas untuk mengatur model dan *view untuk* menangani permintaan user.

2.3.5 *Object Oriented Analysis and Design* (OOAD)

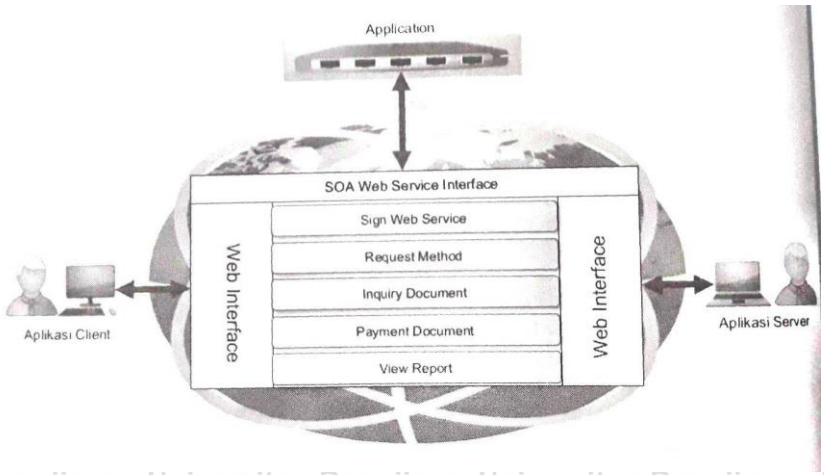
Object Oriented Analysis and Design (OOAD) merupakan salah satu metode untuk menyelesaikan suatu permasalahan dengan menerapkan konsep sekitar dunia nyata pada model yang digunakan (Kendall & Kendall, 2011).

Diagram UML (*Unified Modeling Language*) berperan pentingan dalam melakukan pemodelan sistem berorientasi objek dengan menggunakan pendekatan OOAD. Berikut ini adalah diagram yang ada pada diagram UML, antara lain: *use case diagram, activity diagram*, *class diagram, object diagram, state machine diagram, composite structure diagram, sequence diagram, communication diagram, interaction overview diagram, timing diagram, component diagram, deployment diagram,* dan *package diagram* (Jeffrey Whitten, 2007).

2.3.6 *Service Oriented Architecture* (SOA)

SOA adalah arsitektur yang digunakan oleh perusahan - perusahaan untuk memungkinkan dapat saling bertukar data melalui sebuah layanan yang tersedia. SOA memilki beberapa karakteristik yaitu: (Aditama, 2017)

1. *Loose Coupled* adalah *service* yang harus ada ketika kita menggunakan SOA, karena pemanggilan *service* oleh *service* lainnya dapat dilakukan pada saat *run time*.
2. SOA terbentuk dari dua *service*, yaitu *service implementation* dan *service interface. Service Interface* yaitu sebuah *service* yang berfungsi untuk menentukan masukan dan keluaran serta letak dari *service* tersebut., Sedangkan *service implementation* adalah *logic process* dari sebuah *service*. Gambar 2.9 adalah penggambaran arsitektur dari *service interface* dan *service implementation*.



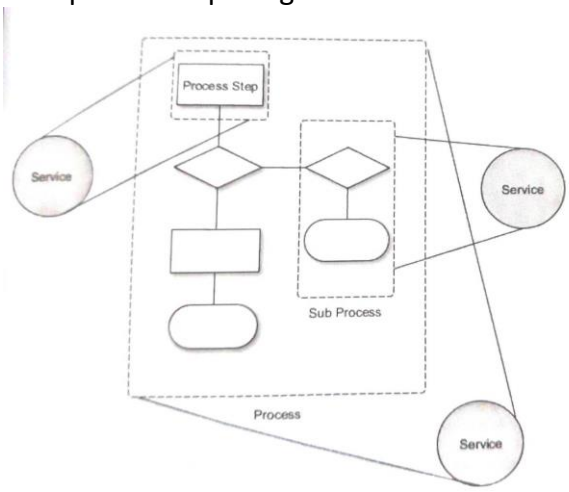
Gambar 2.9 Arsitektur dari *Service Interface* dan *Service Implementation*

Sumber: (Aditama, 2017)

Dari Gambar 2.9 terlihat bagaimana skema yang akan dibangun pada aplikasi pembayaran tiket secara *online*, adanya aplikasi yang saling berkomunikasi antar beda *platform*, dari rancangan tersebut perlu adanya

sebuah rancangan yang mengubah bisnis *logic* menjadi sebuah layanan yang dibungkus dalam sebuah wadah aplikasi atau disebut juga *web service.*

3. *Service* bersifat *business oriented*, adalah setiap *service* yang disediakan harus dapat melakukan aktivitas bisnis tertentu. Adapun bagaimana enkapsulasi logic dalam *service* dapat dilihat pada gambar 2.10.

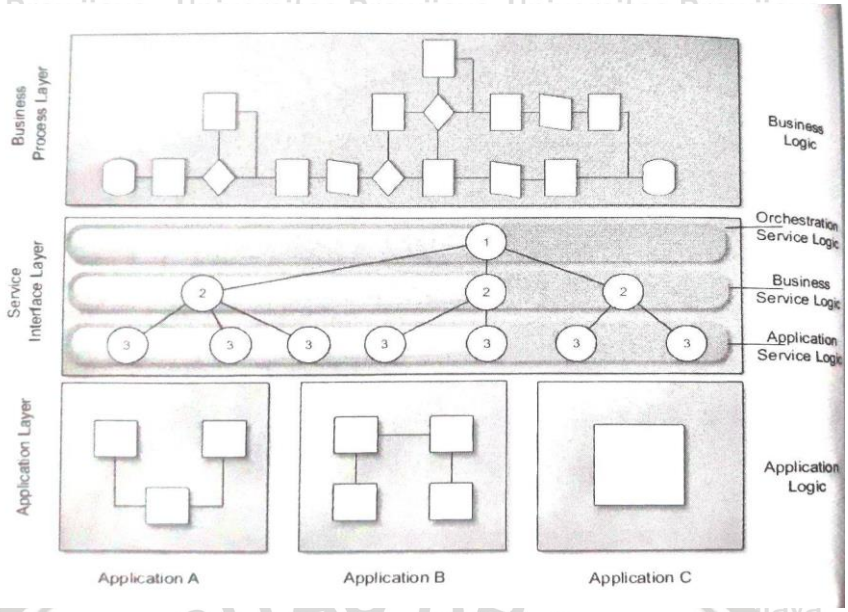


Gambar 2.10 Enkapsulasi *Logic* dalam *Service*

Sumber: (Aditama, 2017)

Dari Gambar 2.10 dapat dilihat bagaimana sebuah *service* mengengkapsulasi sebuah business logic menjadi *service* - *service* kecil.

Menurut (Riyanto, et al., 2013) dalam sebuah pengembangan perangkat lunak menggunakan pendekatan konsep SOA berbeda dengan pengembangan tanpa menggunakan konsep SOA, dimana sebuah perangkat lunak yang dikembangkan tidak menggunakan SOA mengakibatkan aplikasi berjalan pada dua layer utama yaitu *application layer* dan *business process layer*, sementara itu dalam menggunakan konsep SOA di implementasikan dalam sebuah layer di antara *application layer* dan *business process layer* yaitu *service interface layer*, fungsi layer ini adalah menggambarkan *application logic* dan *business process* yang ada di *business logic.*



Gambar 2.11 *Layering* pada SOA

Sumber: (Aditama, 2017)

Dari Gambar 2.11 terlihat bahwa teknologi aplikasi A, B, C ter-enkapsulasi oleh *service interface layer*, *application logic* dan *business logic* dibungkus dalam suatu layanan. Dan *service interface layer* itu sendiri memiliki tiga lapisan abstraksi, yaitu:

1. *Application Service Layer*

   Sebuah layanan yang letaknya berbatasan dengan *application layer*, yang bertugas sebagai penyedia fungsi – fungsi untuk pemrosesan data yang nantinya akan dibutuhkan untuk menjalankan fungsi – fungsi yang ada pada proses bisnis.
2. *Business Service Layer*

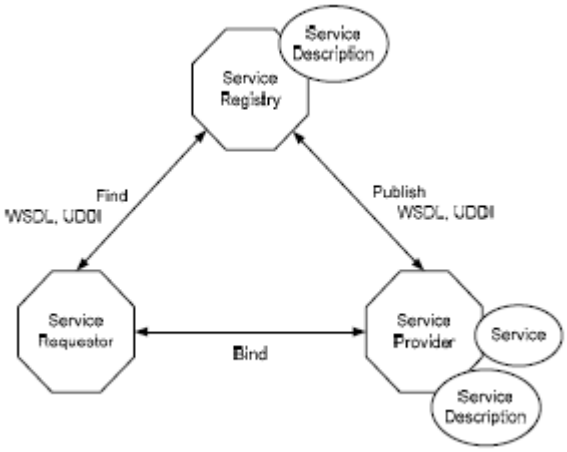
   Digunakan untuk mengimplementasikan *logic* dan model bisnis menjadi sebuah *service*.
3. *Orchestration Service Layer*

   Sebuah layanan yang memberikan abstraksi bagaimana *service – service* yang berjalan sesuai dengan aturan dan urutannya yang sesuai dengan *logic* dan bisnis proses yang ada pada perusahaan.

2.3.7 *Web Service*

Web service adalah rancangan teknologi yang mengubah bisnis *logic* menjadi sebuah layanan yang dapat merealisasikan pertukaran data dari integrasi aplikasi yang ada. *Web service* merupakan solusi yang tepat bagi pengembang sebuah sistem informasi. Karena dengan menggunakan *web service* aplikasi dapat berkomunikasi tanpa adanya batasan penggunaan jenis *hardware*, *software*, sistem operasi, bahasa pemrograman, *database*, maupun *framework*. *Web service* dalam layanan yang diberikan mengembalikan format penulisan yang dapat langsung dibaca pengguna berupa kode XML dan JSON. Operasi suatu

web *service* dapat digambarkan dengan sebuah arsitektur yang memiliki 3 komponen dengan peran berbeda – beda seperti pada Gambar 2.12.



Gambar 2.12 Operasi *Web Service*

Sumber: (Aditama, 2017)

Berikut ini adalah penjelasan komponen operasi *web service* yang terdapat pada Gambar 2.12.

1. Penyedia Layanan (*Service Provider*) berfungsi menyediakan semua layanan yang sudah di register.
2. Daftar Layanan (*Service Registry*) berfungsi menyimpan semua layanan serta jika *service requestor* membutuhan sebuah layanan, maka *service* ini bertugas untuk mencari dan menemukan layanan yang dibutuhan tersebut.
3. Peminta Layanan (*Service Requestor*) berfungsi untuk meminta layanan yang telah tersedia pada *service registry.*

2.3.8 JSON (*JavaScript Object Notation*)

JSON adalah format minimal yang dapat dibaca untuk menyusun data. JSON digunakan untuk mengirimkan data antara server dan aplikasi web. Terdapat dua bagian utama yang membentuk JSON yaitu *key* dan *value*. *Key* selalu berupa string yang disertai tanda kutip. *Value* dapat berupa string, angka, ekspresi *boolean*, *array*, atau objek. Gambar 2.13 adalah contoh penulisan dari JSON.

Gambar 2.13 Penulisan JSON

```
{
"employees":[
    {"firstName":"John", "lastName":"Doe"},
    {"firstName":"Anna", "lastName":"Smith"},
    {"firstName":"Peter", "lastName":"Jones"}
]
}
```

2.3.9 *Representation State Transfer* (REST)

REST (*Representational State Transfer*) merupakan salah satu gaya arsitektur untuk mengembangkan web service (Fielding, 2000). Arsitektur REST-*style* terdiri dari client dan server. Teknik untuk menerapkan web service dengan menggunakan dokumen XML dan JSON dengan menggunakan protocol HTTP (*Hypertext Transfer Protocol*). Pada arsitektur REST terdapat REST *server* dan REST *client.* REST *server* berfungsi untuk mengakses , sedangkan REST *client* berfungsi untuk menampilkan *resource* tersebut untuk penggunaan selanjutnya. Setiap *resource* diidentifikasi oleh URIs (*Universal Resource Identifiers*) atau global ID. *Resource* tersebut direpresentasikan dalam bentuk format teks, JSON atau XML. Pada umumnya menggunakan JSON dan XML. Pada arsitektur REST terdapat beberapa method HTTP yaitu:

1. *GET*, akses yang hanya digunakan untuk membaca *resource*
2. *PUT*, akses yang digunakan untuk membuat *resource* baru
3. *DELETE*, akses digunakan untuk menghapus *resource*
4. *POST*, akses yang digunakan untuk membuat *resource* baru
5. *OPTIONS*, akses yang digunakan untuk mendapatkan operasi yang di*support* pada *resource*

2.3.10 Midtrans

Midtrans merupakan salah satu platform untuk melakukan *payment gateway*. Midtrans merupakan sistem pembayaran yang dapat memfasilitasi penjual dan pembeli untuk melakukan transaksi ((Midtrans (PT Midtrans), 2019). Midtrans menyediakan *tools* terintegrasi ke *e-commerce* sesuai kebutuhan pembayaran secara *online* dengan kartu debit, kartu kredit, bahkan penarikan uang, dan pengiriman uang. Dengan midtrans dapat melakukan pembayaran belanja *online*, donasi, produk berlangganan, dan penarikan uang dengan mudah, cepat dan aman.

2.3.11 *Software Engineering*

Menurut (Pressman, 2010) *Software Engineering* merupakan prinsip yang digunakan dalam membangun sebuah rekayasa perangkat lunak dengan melakukan analisis kebutuhan suatu pelanggan untuk mendapatkan perangkat

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

lunak yang ekonomis, yaitu perangkat lunak yang terpercaya dan bekerja efisien pada komputer. Menurut (Pressman, 2010) secara umum (RPL) Rekayasa Perangkat Lunak merupakan teknologi yang memiliki 4(empat) lapisan yang dapat di lihat pada Gambar 2.14.

Gambar 2.14 Tingkatan Pada Rekayasa Perangkat Lunak

Sumber: (Pressman, 2010)

Tingkatan masing - masing lapisan pada RPL (Rekayasa Perangkat Lunak) dapat dijelaskan sebagai berikut:

a. *Tools,* merupakan alat yang digunakan sebagai metode dan pemrosesan informasi.
b. *Methods,* merupakan mekanisme untuk membangun suatu *software.* .
c. *Process,* mengintegrasikan teknologi antar satu sama lain yang digunakan dalam perancangan suatu program, agar pembuatan program dapat diselesaikan tepat waktu.
d. *A Quality Focus,* sebagai dasar untuk dilakukannya RPL (Rekayasa Perangkat Lunak)

2.3.12 *Unifed Model Language* (UML)

Unifed Model Language (UML) merupakan sebuah bahasa yang menjadi standar untuk pemodelan sistem berorientasi objek. UML berupa standar untuk pemodelan objek bukan metode untuk mengembangkan sistem (Jeffrey Whitten, 2007) UML bertujuan untuk : (Booch, et al., 2005).

1. *Visualizing* adalah memvisualisasikan grafis model yang tepat.
2. *Specifying* adalah membangun model yang tepat dan tidak ambigu.
3. *Constructing* adalah membangun model yang dapat langsung terhubung ke berbagai Bahasa pemrograman.
4. *Documenting* adalah mendokumentasikan mulai arsitektur sistem sampai informasi secara detail dan lengkap yang telah dikumpulkan oleh tim sehingga memungkinkan untuk berbagi informasi.

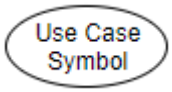
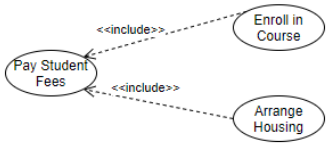
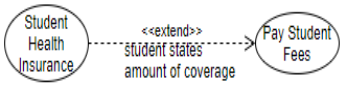
Pada penelitian ini menggunakan *use case diagram, activity diagram, class diagram, sequence diagram*. Berikut ini adalah penjelasan masing-masing diagram yang digunakan pada penelitian ini, yaitu

a. Use Case Diagram

Use Case Diagram adalah diagram yang menunjukkan relasi antar sistem dan user serta mendefinisikan pengguna dan aktivitas yang dapat dilakukan

pengguna pada sistem tersebut (Jeffrey Whitten, 2007). Melalui *use case diagram*, dapat diketahui fungsi-fungsi apa saja yang ada pada sistem. Penamaan pada *use case* harus didefinisikan dengan jelas dan sesederhana mungkin agar mudah dipahami. Tabel 2.1 merupakan penjelasan simbol-simbol yang digunakan pada *use case diagram*.

Tabel 2.1 Simbol-simbol *Use Case Diagram*

No	Simbol	Nama	Keterangan
1		*Use Case*	*Use case* mendefiniskan aktivitas yang dapat dilakukan oleh pengguna pada sistem tersebut
2		*Actor*	*Actor* mendeskripsikan peran pengguna ketika berinteraksi dengan *use case.*
3		*Communicates Relationship*	*Communicates Relationship* digunakan untuk menghubungkan aktor dengan *use case*
4		*Include Relationship*	*Include Relationship* atau disebut juga *uses relationship* menggambarkan suatu *use case* dapat didefiniskan melalui use case lain.
5		*Extend Relationship*	*Extend Relationship* bertujuan untuk menggambarkan sebuah *use case yang* memiliki aktivitas yang dapat digunakan oleh *use case* lain dengan kondisi pengecualian.

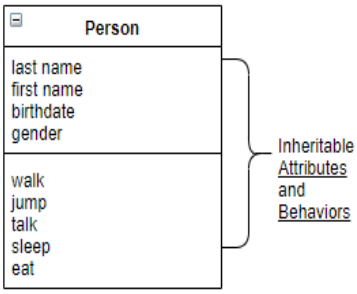
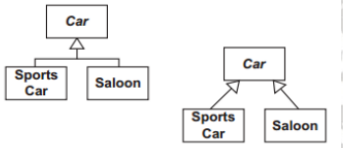
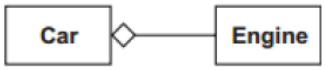
Sumber: (Jeffrey Whitten, 2007)

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

b. Class Diagram

Class Diagram digunakan untuk menggambarkan struktur objek berupa *class* – *class* yang bersifat statis yang digunakan untuk membangun sistem (Jeffrey Whitten, 2007). Tabel 2.2 merupakan penjelasan simbol – simbol yang digunakan untuk membuat *class diagram*.

Tabel 2.2 Simbol-simbol *Class Diagram*

No	Simbol	Nama	Keterangan
1		*Attribute*	*Attribute* adalah sekumpulan yang mengitari *class*, dengan nilai datanya bisa ditentukan di object. Sedangkan *behavior* merupakan fungsi - fungsi yang dimiliki oleh setiap objek.
2		*Inheritance*	*Inheritance*, menggambarkan relasi *is –a* (relasi adalah) hubungan yang menunjukkan pewarian data dan method dari suatu kelas kepada kelas lain.
3		*Association*	*Association*, menunjukkan hubungan objek dari satu kelas dengan kelas lain.
4		*Agregation*	*Agregation*, menggambarkan relasi *has – a* (relasi mempunyai) yang artinya model agregasi memiliki hubungan dan mewakili hubungan kepemilikan antara dua objek.

Tabel 2.2 Simbol-simbol *Class Diagram* (Lanjutan)

No	Simbol	Nama	Keterangan
5		*Composition*	*Composition*, merupakan bentuk khusus dari agregasi (disebut juga *death relationship).* Relasi ini menggambarkan sebuah kondisi jika suatu objek adalah bagian dari (*part – of*) objek lain.

Sumber: (Jeffrey Whitten, 2007)

c. Activity Diagram

Activity Diagram merupakan diagram yang menunjukkan kegiatan yang dilakukan secara berurutan pada *use case* atau proses bisnis (Jeffrey Whitten, 2007). Oleh karena itu, *activity diagram* dapat digunakan untuk menentukan model proses bisnis. Fungsi dari *activity diagram* antara lain :

1. Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses.
2. Memperlihatkan urutan aktifitas proses pada sistem.
3. *Activity diagram* dibuat berdasarkan sebuah atau beberapa *use case* pada *use case diagram*.

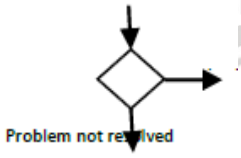
Tabel 2.3 merupakan penjelasan simbol-simbol yang digunakan pada *activity diagram*.

Tabel 2.3 Simbol-simbol *Activity Diagram*

No	*Simbol*	*Nama*	*Keterangan*
1		*Initial node*	*Initial node* berupa lingkaran penuh yang menggambarkan titik mulai suatu proses.
2		*Actions*	*Actions* adalah notasi yang menggambarkan langkah-langkah yang terjadi.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 2.3 Simbol-simbol *Activity Diagram* (Lanjutan)

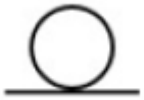
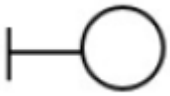
No	*Simbol*	*Nama*	*Keterangan*
3		*Flow* (alur)	*Flow* (alur) merupakan panah dalam diagram yang mengindikasikan alur antar–*actions.*
4		*Decision*	*Decision* memiliki bentuk seperti wajik dengan satu alur masuk dan dua atau lebih alur keluar, alur keluar ditentukan dengan kondisi tertentu.
5		*Activity final*	*Activity final* berbentuk lingkaran penuh dengan satu lingkaran di luarnya untuk menggambarkan titik akhir proses.

Sumber: (Jeffrey Whitten, 2007)

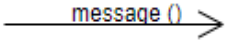
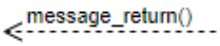
d. Sequence Diagram

Sequence diagram merupakan diagram yang digunakan untuk memodelkan interaksi antara aktor dan objek dalam suatu sistem (Pressman, 2010). *Sequence diagram* menggambarkan bagaimana interaksi antar objek dalam mengirim maupun menerima pesan pada sekuensi sebuah operasi. Pada Tabel 2.4 menjelaskan simbol-simbol yang digunakan untuk membuat *sequence diagram*.

Tabel 2.4 Simbol-simbol *Sequence Diagram*

No	Simbol	Nama	Keterangan
1		*Entity*	*Entity*, menggambarkan sebuah atribut yang memiliki data.
2		*Boundary*	*Boundary*, menggambarkan relasi antara aktor dengan sistem.

Tabel 2.5 Simbol-simbol *Sequence Diagram*

No	Simbol	Nama	Keterangan
3		*Control*	*Control*, untuk mengatur aktivitas-aktivitas yang dilakukan oleh sebuah kegiatan.
4		*Message*	*Message*, pengiriman pesan.
5		*Return Values*	*Return Values*, ditampilkan dengan garis panah terputus yang menggambarkan hasil dari pengiriman pesan. Digambarkan arah dari kanan ke kiri.
6		*Lifelines*	Garis kehidupan (*Lifelines*), Garis vertical putus-putus yang memanjang kebawah dari simbol *actor* dan sistem yang mengindikasikan urutan kehidupan.
7		*Bar*	*Bar* aktivasi, *Bar* didalam garis kehidupan yang menunjukkan periode waktu ketika peserta aktif dalam interaksi.

Sumber: (Pressman, 2010)

2.3.13 *System Development Live Cycle* (SDLC)

Pengertian definisi *System Development Life Cycle* (SDLC) menurut (A.S. & M.Shalahuddin, 2016) menyatakan bahwa :

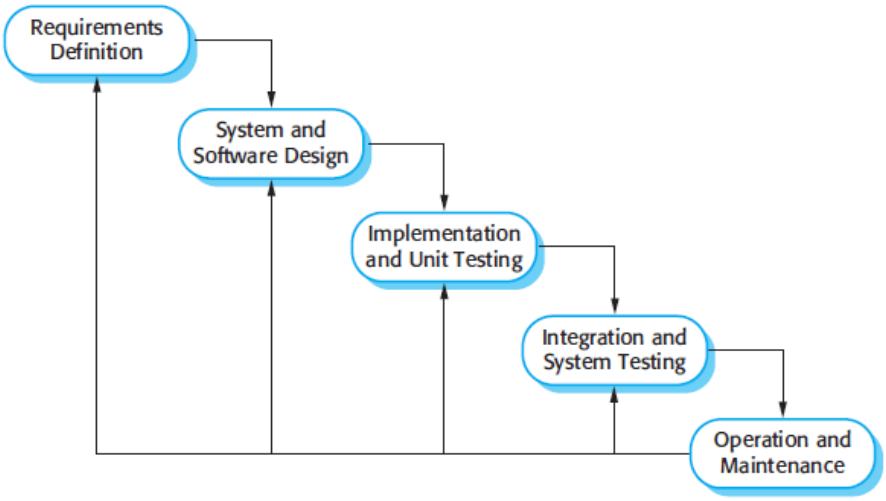
System Development Life Cycle (SDLC) adalah salah satu metode pengembangan sistem informasi yang popular pada saat sistem informasi pertama kali dikembangkan. Metode SDLC adalah tahap-tahap pengembangan sistem informasi yang pertama kali dikembangkan yang dilakukan oleh analisis sistem

dan programmer untuk membangun sebuah sistem informasi. Metode SDLC ini seringkali dinamakan sebagai proses pemecahan masalah, yang langkah-langkahnya adalah:

1. Analisis, Tahap mempelajari sistem informasi yang sedang berjalan sangat berguna untuk mengetahui sebab dan akibat yang ditimbulkan oleh masalah, sehingga akan menghasilkan pelaporan yang mengungkapkan adanya permasalahan.
2. Perancangan, Memahami bagaimana menterjemahkan keinginan pemakai sistem informasi tersebut kedalam bahasa komputer, untuk memulai merancang suatu sistem informasi baru yang meliputi : *input*, *file-file database* dan *output*, bahasa yang digunakan, metode dan prosedur serta pengendalian.
3. Penerapan, Hasil penyusunan sistem informasi adalah sebuah *software* komputer yang siap digunakan untuk kebutuhan *user* untuk dioperasikan.
4. Pemeliharaan, Pemeliharaan yang dilakukan analis adalah dengan melakukan perbaikan dan pemeliharaan pada kesalahan atau kegagalan yang timbul dalam penggunaan sistem informasi.

2.3.14 *Waterfall Model*

Menurut Sommerville (Sommerville, 2011) *Waterfall model* memiliki beberapa tahapan, sebagaimana terdapat pada Gambar 2.15.



Gambar 2.15 *Waterfall Model*

Sumber: (Sommerville, 2011)

Berikut ini adalah penjelasan tahapan *Waterfall model* yang sesuai pada Gambar 2.15, yaitu

1. *Requirement Analysis and Definition*
   Pada tahap dilakukan pendefinisian kebutuhan sistem yang didapat dari pengguna siistem melalui wawancara yang nantinya didapatkan tujuan serta layanan sistem.

2. *System and Software Design*
   Membahas mengenai rancangan sistem dengan membuat sebuah arsitektur sistem sesuai dengan kebutuhan yang telah didefiniskan sebelumnya.
3. *Implementation and Unit Testing*
   Selama tahap ini, perancangan perangkat lunak diwujudkan sebagai satu set program atau unit program.
4. *Integration and System Testing*
   Program yang telah dikembangkan akan dilakukan pengujian pada masing – masing unit untuk memastikan bahwa program telah dikembangkan sesuai dengan kebutuhan yang telah didefinisikan.
5. *Operation and Maintenance*
   Program yang telah diimplementasikan diperlukan proses *maintenance*. Proses *maintenance* ini merupakan fase yang lama karena, pada proses ini akan dilakukan pengindentifikasian kesalahan yang ada pada sistem untuk dilakukan pembenaran. Proses ini bertujuan untuk meningkatkan meningkatkan kualitas layanan sistem.

2.3.15 Pengujian *Black-box*

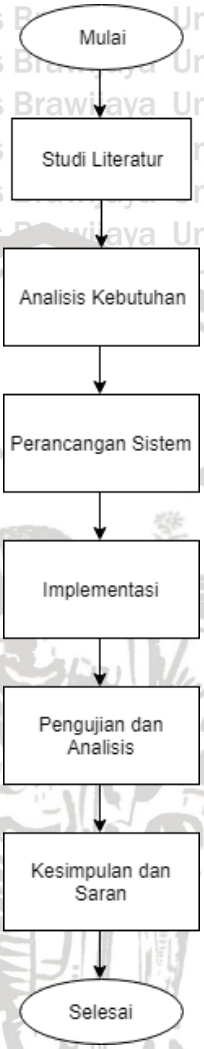
Pengujian *black-box* berfokus pada kebutuhan fungsional dari sebuah perangkat lunak. Teknik pengujian *black-box* dapat digunakan untuk menganalisis terjadinya kesalahan dalam kategori berikut:

1. Mengidentifikasi adanya ketidaksesuain fungsi dalam spesifikasi perangkat lunak baik fungsi tidak benar maupun fungsi yang hilang.
2. Mengidentifikasi adanya kesalahan pada *interface*.
3. Mengidentifikasi terjadinya kesalahan dalam struuktur data.
4. Mengidentifikasi ketidaksesuain kinerja atau perilaku.
5. Melakukan inisialisasi dan pemutusan kesalahan.

Validation testing dilaksanakan ketika sebuah *software* telah selesai dikembangkan sesuai dengan kebutuhan fungsional. Kegiatan pengujian validasi berfokus pada fungsi – fungsi yang dapat dilakukan oleh pengguna untuk membuktikkan bahwa sistem memenuhi kebutuhan fungsional yang telah didefinisikan.

BAB 3 METODOLOGI PENELITIAN

3.1 Struktur Tahapan Penelitian



Gambar 3.1 Diagram Alir Metode Penelitian Aplikasi *E-commerce* Wisata Kampung Sapi *Adventure*

Sumber: (Penulis, 2019)

Pembuatan aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* pada laporan ini menggunakan tahapan seperti pada Gambar 3.1. Penelitian menggunakan metode *waterfall*. Tahapan atau metode pelaksanaan dalam pembuatan aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* terdiri dari studi literatur, analisis kebutuhan, perancangan sistem, implementasi, pengujian, serta kesimpulan dan saran.

3.2 Studi Literatur

Studi literatur merupakan tahap mempelajari dan mencari referensi dari buku, *e-book* ataupun jurnal untuk memperoleh penjelasan teori sesuai dengan

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

penelitian yang dilakukan, sehingga teori tersebut diharapkan dapat mempermudah dan membantu dalam melaksanakan penelitian.

3.3 Analisis Kebutuhan

Tahapan analisis kebutuhan dilakukan oleh penulis melalui wawancara dengan pengelola Wisata Kampung Sapi *Adventure*. Wawancara dilakukan demi menganalisa kebutuhan yang diperlukan sistem, selain itu menggabungkan gagasan-gagasan dari penulis dan pemilik. Sehingga kedua belah pihak memiliki pemahaman yang sama dalam mengembangkan sistem. Pada akhirnya, setelah melakukan analisis kebutuhan dapat diketahui fungsional, data maupun dokumen yang diperlukan.

3.4 Perancangan Sistem

Perancangan sistem perlu dilakukan untuk mengetahui tahapan - tahapan yang harus dilakukan berdasarkan analisis kebutuhan yang telah didefiniskan dengan benar, agar terdapat kesesuaian hasil rancangan dan implementasi sistem . Dapat digunakan *Unified Modelling Language (UML)* sebagai alat bantu dalam melakukan perancangan aplikasi *e-commerce* Wisata Kampung Sapi Adventure. UML yang digunakan terdiri dari *class diagram*, *use case diagram*, *sequence diagram*, dan *entity relationship diagram* (ERD) yang berfungsi untuk menggambarkan perancangan *database* sekaligus entitas yang digunakan.

3.5 Implementasi Sistem

Tahapan pembuatan sistem dengan menggunakan pengkodean baris kode (*source code*). Implementasi dilakukan sesuai dengan perancangan atau *design* sistem yang telah dilakukan. Bahasa pemrograman yang digunakan antara lain PHP, HTML, Java*script*, dan CSS.

3.5.1 Implementasi *E-commerce*

Implementasi pembuatan *e-commerce* Wisata Kampung Sapi *Adventure* dibangun dengan konsep pengintegrasian salah satu layanan *payment gateway* menggunakan bahasa pemrograman PHP dan menggunakan *framework CodeIgniter* untuk mempermudah wisatawan dalam melakukan reservasi maupun transaksi dan mempermudah pengelola wisata Kampung Sapi *Adventure* dalam membuat laporan transaksi.

3.5.2 Implementasi *Payment Gateway*

Implementasi penggunaan *payment gateway* pada penelitian ini dengan menggunakan API dari salah satu layanan *payment gateway* midtrans. Pengintegrasian *e-commerce* Wisata Kampung Sapi *Adventure* dengan midtrans dapat memudahkan wisatawan dalam melakukan transaksi secara *online* dengan mudah, cepat dan aman.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

3.6 Pengujian

Pengujian sistem menggunakan metode *blackbox*. Pengujian dilakukan setelah aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* selesai dibuat. Sebuah perangkat lunak dilakukan pengujian, untuk menemukan *error* atau kesalahan dalam pengembangan sistem. Selain itu untuk menghindari kesalahan sebanyak mungkin sebelum digunakan pengguna.

3.7 Kesimpulan dan Saran

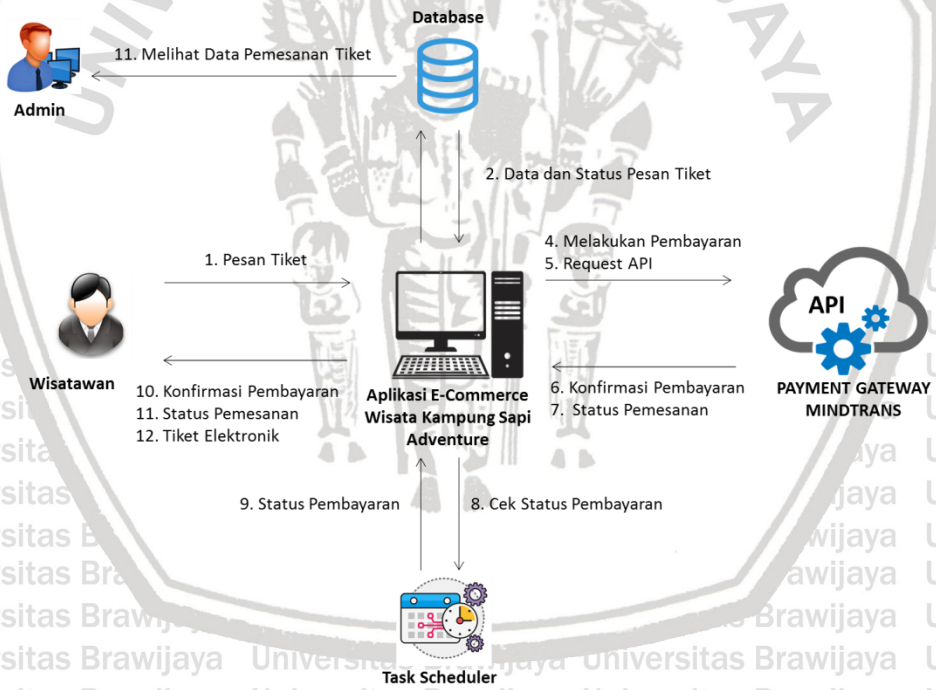
Tahapan akhir dalam pengembangan aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* ini adalah pengambilan kesimpulan. Tahap ini dilakukan untuk menyimpulkan dari permasalahan yang mendasari perancangan dan pengembangan aplikasi *e-commerce* Wisata Kampung Sapi *Adventure*.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB 4 ANALISIS KEBUTUHAN DAN PERANCANGAN SISTEM

Pada bab ini akan membahas terkait arstiketur, pemodelan proses bisnis dan analisis kebutuhan pada aplikasi *e-commerce* Wisata Kampung Sapi *Adventure*. Pemodelan proses bisnis dalam pembangunan aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* terdiri dari proses bisnis yang berjalan sebelumnya atau *as-is* dan proses bisnis yang diusulkan oleh peneliti atau *to-be*. Pemodelan proses bisnis dapat dimodelkan dengan menggunakan *Business Process Modeling Notation* (BPMN). Analisis kebutuhan aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* didapat dari hasil wawancara yang telah dilakukan sebelumnya dengan bapak Aribowo selaku pengelola Wisata Kampung Sapi *Adventure*. Analisis kebutuhan aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* dilakukan mulai dari mengidentifikasi aktor, mengidentifikasi kebutuhan fungsional dan non fungsional sistem, menggambarkan kebutuhan sistem ke dalam *use case diagram*, hingga membuat spesifikasi *use case.*

4.1 Arsitektur Aplikasi *E – Commerce* Wisata Kampung Sapi *Adventure*



Gambar 4.1 Arsitektur Aplikasi *E – Commerce* Wisata Kampung Sapi *Adventure*

Sumber: (Penulis, 2019)

Sistem yang akan dirancang pada penelitian ini adalah aplikasi *e – commerce* Wisata Kampung Sapi *Adventure* didapat dari hasil wawancara yang telah Wisata Kampung Sapi *Adventure*. Aplikasi *e-commerce* ini dirancang untuk untuk mempermudah pendataan wisatawan dalam hal transaksi jual beli tiket dan reservasi serta memudahkan petugas wisata dalam melakukan pencatatan

transaksi penjualan tiket. Aplikasi *e-commerce* yang akan dibangun adalah aplikasi berbasis web dengan memanfaatkan layanan *payment gateway* midtrans. Pengguna aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* adalah wisatawan dan pengelola Wisata Kampung Sapi *Adventure*.

Pada Gambar 4.1 merupakan rancangan arsitektur secara umum aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* yang akan dikembangkan. Wisatawan melakukan pemesanan tiket pada aplikasi *e-commerce* Wisata Kampung Sapi *Adventure.* Data pemesanan dan status pesan tiket akan tersimpan pada *database.* Kemudian wisatawan akan melakukan pembayaran dengan melakukan *request* pada API midtrans, selanjutnya wisatawan menerima konfirmasi dan status pembayaran dari API midtrans untuk melakukan pembayaran langsung atau menunda pembayaran. Pembayaran dengan memanfaatakan layanan API *sandbox* dari midtrans dibangun menggunakan konsep SOA. Ketika wisatawan melakukan pembayaran pada midtrans maka, aplikasi *e-commerce* Wisata Kampung Sapi Adventure akan melakukan pengecekan status pembayaran menggunakan *task scheduler* setiap 1 menit sekali. Setelah dilakukan pengecekan status pembayaran maka data pemesan tiket pada *database* terkait status pembayaran tiket akan mengalami pembaruan secara otomatis. Jika status pembayaran sukses maka aplikasi *e – commerce* Wisata Kampung Sapi Adventure akan memberikan elektronik tiket kepada wisatawan. Kemudian admin dapat melihat data pemesanan tiket yang telah tersimpan pada *database*.

4.2 Pemodelan Proses Bisnis

Pada penelitian ini didapat dua pemodelan proses bisnis yaitu pemodelan proses bisnis saat ini atau *as-is* dan pemodelan proses bisnis usulan atau *to-be*. Pemodelan proses bisnis dilakukan untuk mengetahui alur proses bisnis yang terjadi saat ini pada Wisata Kampung Sapi *Adventure*. Pemodelan proses bisnis yang telah dilakukan akan menghasilkan proses bisnis saat ini yang nantinya dapat dijadikan acuan peneliti dalam membuat proses bisnis usulan. Proses bisnis usulan diharapkan dapat memperbaiki proses bisnis yang saat ini berjalan, agar dapat memberikan pelayanan yang lebih baik bagi wisatawan dalam melakukan pemesanan tiket dan mempermudah petugas tiket dalam melakukan pendataan transaksi penjualan pemesanan tiket pada Wisata Kampung Sapi *Adventure*.

4.2.1 Pemodelan Proses Bisnis *As-Is*

Pemodelan proses bisnis yang berjalan saat ini atau *as-is* yang telah dilakukan sebelumnya melalui wawancara dengan bapak Aribowo selaku pengelola Wisata Kampung Sapi *Adventure* dapat dimodelkan dengan *Business Process Modelling Notation* (BPMN). Proses bisnis yang terjadi saat ini pada Wisata Kampung Sapi *Adventure* didapat pada proses bisnis pemesanan tiket yang dilakukan oleh wisatawan dan penginputan data transaksi yang dilakukan secara manual ke dalam *Spreadsheet* oleh petugas tiket. Wisatawan yang ingin

melakukan pemesanan tiket Wisata Kampung Sapi *Adventure* dapat dilakukan dengan menghubungi pihak Wisata Kampung Sapi *Adventure* melalui whatsapp maupun datang langsung ke loket tiket Wisata Kampung Sapi *Adventure.*

4.2.1.1 Proses Bisnis Saat Ini Pada Pemesanan Tiket Wisata Kampung Sapi *Adventure* Melalui Via Whatsapp

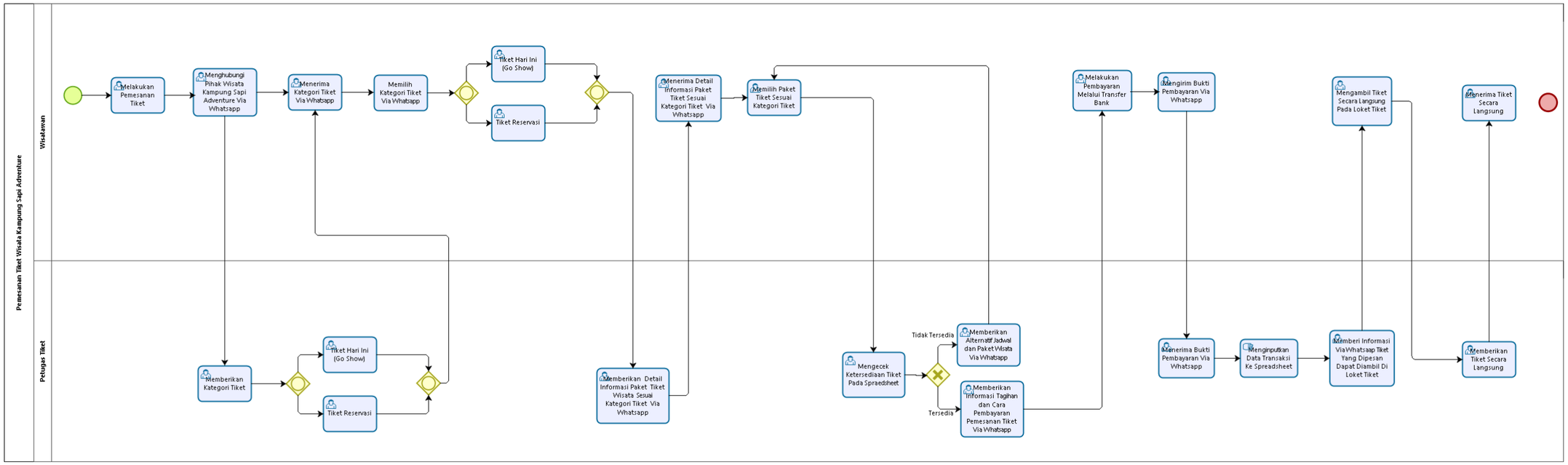
Proses bisnis saat ini pada pemesanan tiket Wisata Kampung Sapi *Adventure* melalui via whatsapp dapat dilihat pada Gambar 4.2. Berikut adalah uraian proses bisnis saat ini pada pemesanan tiket Wisata Kampung Sapi *Adventure* melalui via whatsapp yang terdapat pada Gambar 4.2:

1. Wisatawan akan melakukan pemesanan tiket
2. Untuk melakukan pemesanan tiket, maka wisatawan menghubungi pihak Wisata Kampung Sapi *Adventure* via whatsapp
3. Kemudian petugas tiket Sapi *Adventure* akan memberikan pilihan kategori tiket via whatsapp
4. Pilihan kategori tiket yang diberikan oleh petugas tiket terdiri dari kategori tiket hari ini (*go show*) dan tiket reservasi
5. Selanjutnya wisatawan menerima kategori tiket via whatsapp
6. Wisatawan memilih kategori tiket sesuai dengan kategori tiket yang diberikan petugas tiket
7. Setelah wisatawan memilih kategori tiket, maka petugas tiket akan memberikan detail informasi paket tiket wisata sesuai kategori tiket yang dipilih wisatawan via whatsapp
8. Langkah selanjutnya wisatawan menerima detail informasi paket tiket sesuai kategori tiket via whatsapp
9. Wisatawan yang telah menerima detail infromasi paket tiket dari petugas tiket dapat memilih paket tiket sesuai kategori tiket yang dipilih
10. Kemudian petugas tiket mengecek ketersediaan tiket pada *Spreadsheet.*
11. Jika paket tidak tersedia, maka petugas tiket akan memberikan alternatif jadwal dan paket wisata via whatsapp. Wisatawan kemudian diminta petugas untuk memilih paket kembali. Sedangkan jika paket tersedia, maka petugas tiket akan memberikan informasi tagihan dan cara pembayaran pemesanan tiket kepada wisatawan

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

12. Wisatawan yang telah menerima informasi tagihan dan cara pembayaran dari petugas tiket dapat melakukan pembayaran melalui transfer bank
13. Wisatawan yang telah melakukan pembayaran melalui transfer bank diminta untuk mengirimkan bukti pembayaran oleh petugas tiket via whatsapp
14. Petugas tiket menerima bukti pembayaran via whatsapp
15. Setelah itu petugas tiket menginputkan data transaksi ke *Spreadsheet*
16. Petugas Tiket memberikan informasi via whatsapp pengambilan tiket yang telah dipesan dapat diambil pada loket tiket
17. Wisatawan dapat melakukan pengambilan tiket dengan datang langsung ke loket tiket Wisata Kampung Sapi *Adventure* dan memperlihatkan bukti pembayaran
18. Petugas tiket memberikan tiket secara langsung
19. Wisatawan merima tiket dari petugas tiket secara langsung
20. Wisatawan berhasil melakukan pemesanan tiket

Pada Gambar 4.2 merupakan pemodelan proses bisnis saat ini pada Wisata Kampung Sapi *Adventure* melalui via *whatsapp* dengan menggunakan *Business Process Modelling Notation* (BPMN).



Gambar 4.2 Proses Bisnis Saat Ini Pada Pemesanan Tiket Wisata Kampung Sapi *Adventure* Melalui Via Whatsapp

Sumber: (Penulis, 2019)

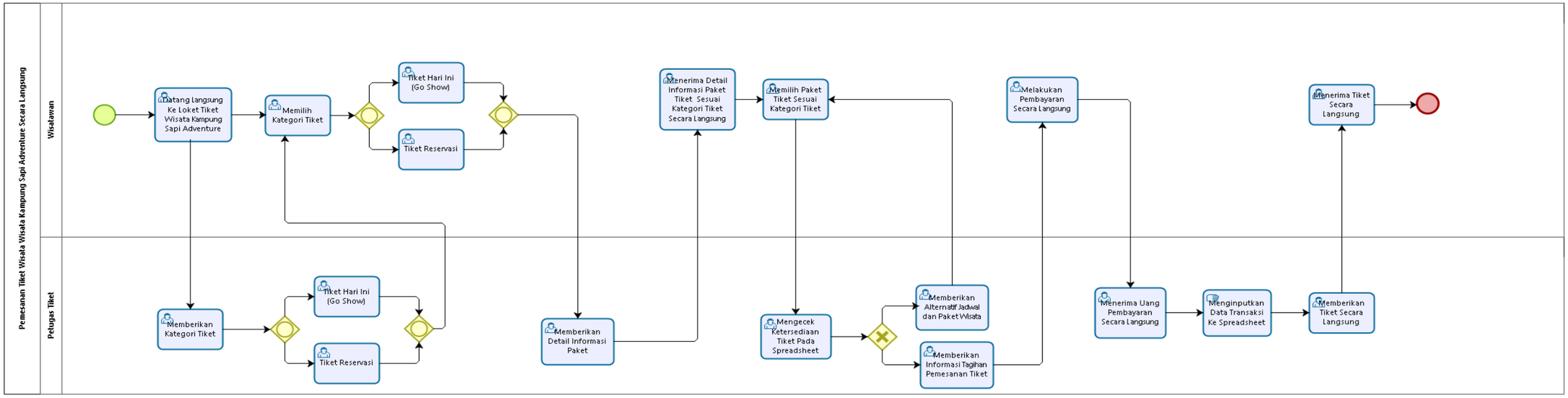
4.2.1.2 Proses Bisnis Saait Ini Pada Pemesanan Tiket Wisata Kampung Sapi *Adventure* Secara Langsung Ke Loket Tiket

Proses bisnis saat ini pada pemesanan tiket Wisata Kampung Sapi *Adventure* secara langsung ke loket tiket dapat dilihat pada Gambar 4.3. Berikut adalah uraian proses bisnis saat ini pada pemesanan tiket Wisata Kampung Sapi *Adventure* secara langsung ke loket tiket yang terdapat pada Gambar 4.3.

1. Wisatawan yang akan melakukan pemesanan tiket Wisata Kampung Sapi *Adventure* dapat dilakukan dengan datang langsung ke loket tiket Wisata Kampung Sapi *Adventure*
2. Petugas tiket akan memberikan pilihan kategori tiket secara langsung
3. Setelah itu wisatawan menerima kategori tiket secara langsung
4. Wisatawan dapat memilih kategori tiket secara langsung
5. Kategori tiket yang tersedia yaitu tiket untuk hari ini (*go show*) dan tiket reservasi
6. Kemudian petugas tiket akan memberikan detail informasi paket sesuai dengan kategori tiket yang dipilih oleh wisatawan secara langsung
7. Wisatawan menerima detail infomasi paket tiket sesuai kategori tiket secara langsung
8. Langkah selanjutnya wisatawan dapat memilih paket tiket sesuai kategori tiket yang dipilih
9. Petugas tiket akan mengecek ketersediaan tiket pada spreadsheet
10. Jika tiket tidak tersedia maka petugas tiket akan memberikan alternative jadwal dan paket wisata. Wisatawan kemudian diminta oleh petugas tiket untuk memilih paket kembali. Sedangkan tiket tersedia maka petugas tiket akan memberikan informasi tagihan pemesanan tiket kepada wisatawan
11. Wisatawan menerima tagihan pemesanan tiket dan melakukan pembayaran pemesanan tiket secara langsung
12. Petugas tiket akan menerima uang pembayaran pemesanan tiket secara langsung
13. Jika wisatawan telah melakukan pembayaran pemesanan tiket secara langsung maka petugas tiket akan menginputkan data transaksi ke *Spreadsheet*
14. Langkah selanjutnya petugas tiket akan memberikan tiket secara langsung
15. Wisatawan merima tiket dari petugas tiket secara langsung

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Pada Gambar 4.3 merupakan pemodelan proses bisnis saat ini pada Wisata Kampung Sapi *Adventure* secara langsung ke loket tiket dengan menggunakan *Business Process Modelling Notation* (BPMN).



Gambar 4.3 Proses Bisnis Saait Ini Pada Pemesanan Tiket Wisata Kampung Sapi Adventure Secara Langsung Ke Loket Tiket

Sumber: (Penulis, 2019)

4.2.2 Proses Bisnis *To-Be*

Proses bisnis usulan yang diusulkan peneliti dapat dimodelkan dengan menggunakan *Business Process Modelling Notation* (BPMN). Proses bisnis usulan dibuat berdasarkan pemodelan proses bisnis yang berjalan saat ini pada Wisata Kampung Sapi *Adventure*. Berdasarkan pemodelan yang telah dilakukan pada proses bisnis yang berjalan saat ini didapat beberapa aktivitas yang masih dilakukan secara manual yaitu pemesanan tiket dan penginputan data transaksi pemesan tiket pada Wisata Kampung Sapi *Adventure*. Oleh karena itu, peneliti mengusulkan proses bisnis usulan atau *to-be*. Pada proses bisnis usulan peneliti akan memperbaiki beberapa aktivitas yang masih dilakukan secara manual tersebut.

4.2.2.1 *Proses Bisnis Usulan Pemesanan Tiket Wisata Kampung Sapi Adventure*

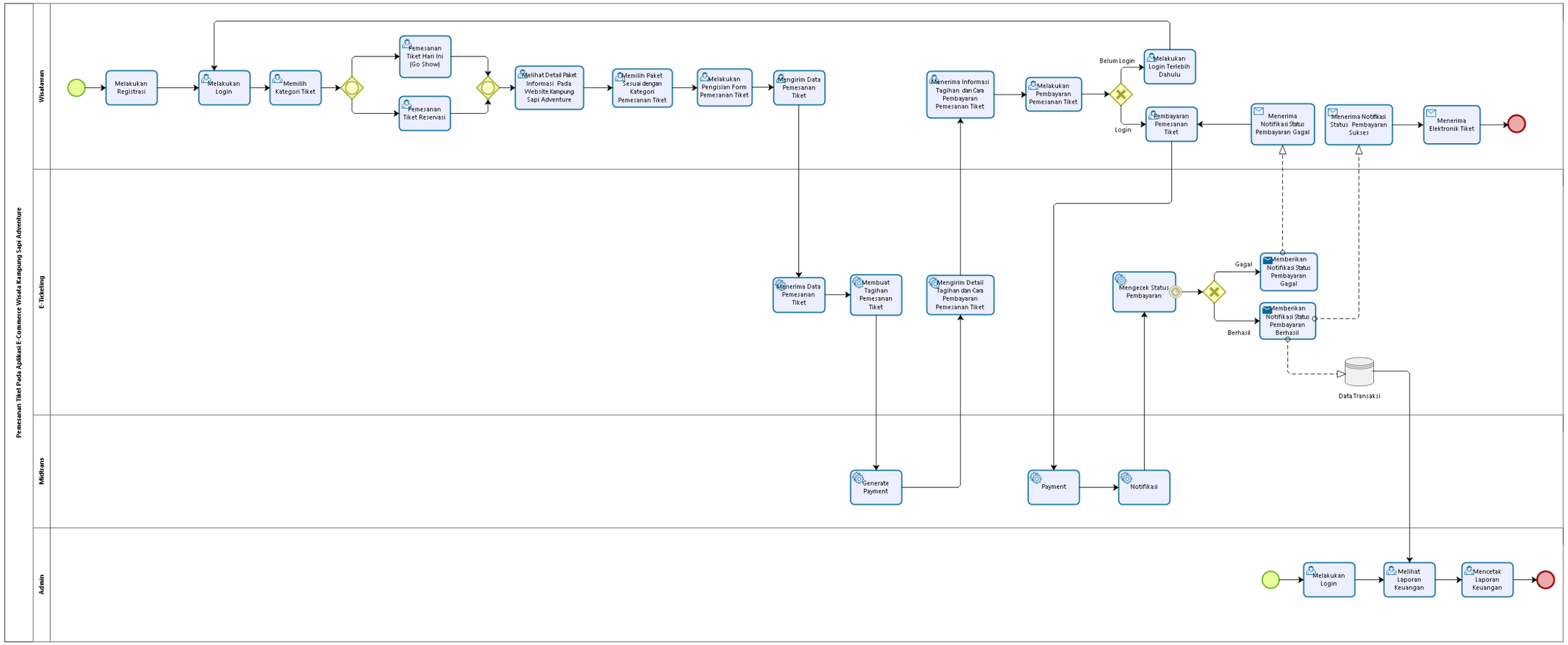
Proses bisnis usulan pada pemesanan tiket Wisata Kampung Sapi *Adventure* dapat dilihat pada Gambar 4.4. Berikut adalah uraian proses bisnis usulan pada pemesanan tiket Wisata Kampung Sapi *Adventure* yang terdapat pada Gambar 4.4.

1. Wisatawan dapat melakukan register terlebih dahulu sebelum melakukan pemesanan tiket
2. Wisatawan yang telah melakukan register dapat melakukan *login*
3. Wisatawan dapat memilih kategori tiket
4. Kategori tiket terdiri dari tiket hari ini (*go show*) dan tiket reservasi
5. Wisatawan yang telah memilih kategori tiket, kemudian wisatawan dapat melihat detail paket informasi pada aplikasi *e-commerce* Wisata Kampung Sapi *Adventure*
6. Setelah itu wisatawan dapat memilih paket sesuai kategori tiket yang dipilih
7. Wisatawan yang telah memilih paket diharuskan melakukan pengisian form pemesanan tiket
8. Data form pemesanan tiket dikirim pada sistem e-ticketing
9. Sistem *e-ticketing* menerima data form pemesanan tiket
10. Kemudian sistem *e-ticketing* membuat tagihan pemesanan tiket
11. Tagihan pemesan tiket yang dibuat pada sistem *e-ticketing* akan dilakukan *generate payment* pada midtrans
12. Sistem *e-ticketing* akan mengirimkan detail tagihan cara pembayaran pemesanan tiket

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

13. Selanjutnya wisatawan meneriman informasi tagihan dan cara pembayaran pemesanan tiket
14. Jika wisatawan yang akan melakukan pembayaran tiket namun belum login, maka wisatawan akan diharuskan melakukan login terlebih dahulu. Jika wisatawan sudah login, maka wisatawan dapat melakukan pembayaran
15. Midtrans akan menerima pembayaran dari wisatawan
16. Midtrans akan memberikan notifikasi pembayaran ke sistem *e-ticketing* untuk melakukan pengecekan status pembayaran yang dilakukan wisatawan
17. Pengecekan status pembayaran dilakukan dalam waktu 1 menit sekali pada sistem *e-ticketing*. Jika status pembayaran gagal maka sistem *e-ticketing* akan memberikan notifikasi kepada wisatawan bahwa status pembayaran gagal, kemudian wisatawan akan menerima notifikasi tersebut dan wisatawan diharuskan untuk melakukan pembayaran pemesanan tiket kembali. Sedangkan jika status pembayaran pemesanan tiket berhasil, maka sistem *e-ticketing* akan memberikan notifikasi status pembayaran berhasil.
18. Admin dapat melakukan login
19. Admin yang telah login dapat melihat laporan keuangan dan mencetak laporan keuangan

Pada Gambar 4.4 merupakan pemodelan proses bisnis usulan pada Wisata Kampung Sapi *Adventure* dengan menggunakan *Business Process Modelling Notation* (BPMN).



Gambar 4.4 Proses Bisnis Usulan Pemesanan Tiket Kampung Sapi *Adventure*

Sumber: (Penulis, 2019)

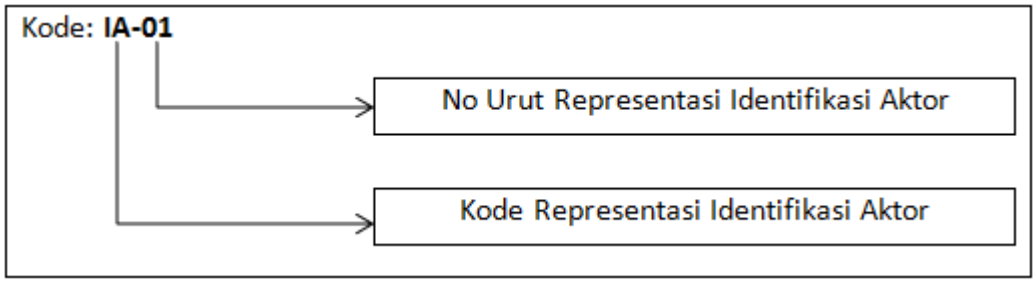
Pada pemodelan proses bisnis *as-is* menjadi *to-be* terdapat beberapa perubahan aktivitas yang berdampak pada aktor di dalam proses bisnis pemesanan tiket Wisata Kampung Sapi *Adventure.* Perubahan aktivitas yang diusulkan peneliti melalui proses bisnis usulan ini dikarenakan untuk meningkatkan kebutuhan wisatawan dalam melakukan pemesanan tiket dan petugas tiket dalam membuat laporan keuangan penjualan tiket.

4.3 Analisis Kebutuhan

Pada bagian ini menjelaskan kebutuhan yang diperlukan oleh sistem, seperti identifikasi aktor, kebutuhan fungsional, dan kebutuhan non-fungsional.

4.3.1 Identifikasi Aktor

Pada bagian ini menjelaskan beberapa jenis aktor yang terlibat pada sistem. Tiap aktor akan memiliki fungsionalnya masing-masing dan diberikan aturan penomoran. Aturan penomoran akan dijelaskan pada Gambar 4.5. Identifikasi aktor dilakukan dengan menggunakan aturan penomoran yang dapat dijelaskan pada Tabel 4.1.



Gambar 4.5 Aturan Penomoran Identifikasi Aktor

Sumber: (Penulis, 2019)

Pada Gambar 4.5 terdapat kode IA-01 yang merupakan aturan penomoran pada identifikasi aktor.

Tabel 4.1 Identifikasi Aktor

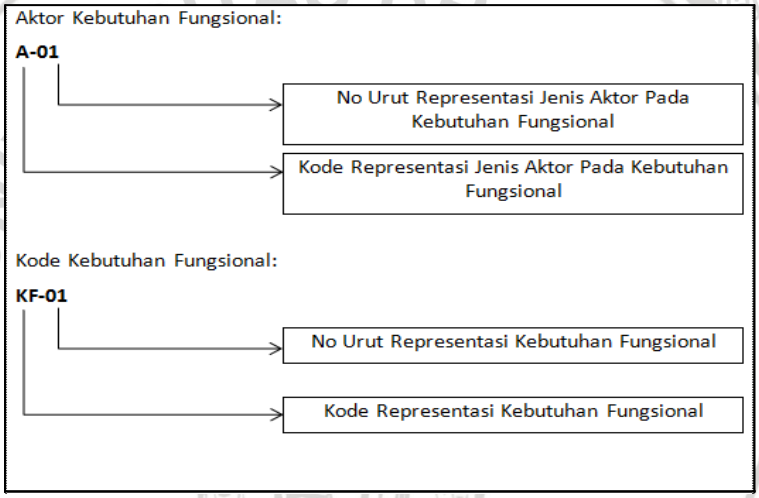
Kode Aktor	Aktor	Deskripsi
IA-01	Wisatawan	Wisatawan merupakan seseorang yang menggunakan aplikasi *e-commerce* Wisata Kampung Sapi Adventure untuk melakukan pemesanan tiket.
IA-02	Admin	Admin merupakan seseorang yang memilki tanggung jawab untuk mengelola data pada aplikasi *e-commerce* Wisata Kampung Sapi *Adventure.*

Sumber: (Penulis, 2019)

Pada Tabel 4.1 merupakan penjelasan identifikasi aktor pada aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* dengan menggunakan aturan penomoran. Analisis kebutuhan identifikasi aktor pada Tabel 4.1 didapat dua jenis aktor yang terlibat pada aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* yaitu aktor wisatawan dan aktor admin.

4.3.2 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang berisi proses-proses atau layanan apa saja yang harus disediakan oleh sistem. Kebutuhan fungsional sistem akan dibagi berdasarkan jenis aktor yang terlibat. Pada masing – masing penjelasan kebutuhan fungsional diberikan aturan penomoran yang terdapat pada Gambar 4.6. Kebutuhan fungsional pada aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* dapat dibagi menjadi dua yaitu kebutuhan fungsional aktor wisatawan yang ditunjukkan pada Tabel 4.2. Serta kebutuhan fungsional aktor admin ditunjukkan pada Tabel 4.3.



Gambar 4.6 Aturan Penomoran Pada Kebutuhan Fungsional

Sumber: (Penulis, 2019)

Pada Gambar 4.6 terdapat kode A-01 yang merupakan aturan penomoran aktor kebutuhan fungsional. Sedangkan pada kode KF-01 merupakan aturan penomoran kebutuhan fungsional

Tabel 4.2 Kebutuhan Fungsional Wisatawan

No.	[A01] Wisatawan		
	Kode	Nama Fungsi	Deskripsi
1.	KF-01	Melakukan Register	Sistem dapat digunakan melakukan register.
2.	KF-02	Melakukan *Login*	Sistem dapat melakukan *login*.

Tabel 4.2 Kebutuhan Fungsional Wisatawan (Lanjutan)

No.	[A01] Wisatawan		
	Kode	Nama Fungsi	Deskripsi
3.	KF-03	Melakukan *Logout*	Sistem dapat digunakan untuk melakukan *logout.*
4.	KF-04	Melihat Kegiatan	Sistem dapat digunakan untuk melihat kegiatan pada Wisata Kampung Sapi *Adventure*.
5.	KF-05	Melakukan Pemesanan Tiket	Sistem dapat digunakan untuk melakukan pemesanan tiket dengan memilih kategori pemesanan tiket hari ini (*go show*) maupun reservasi, melihat fasilitas dan kegiatan paket sesuai kategori yang dipilih, mengecek tanggal, mengetahui ketersediaan tiket, melakukan pembelian tiket, melakukan pembayaran melalui midtrans, dan memberikan elektronik tiket.

Sumber: (Penulis, 2019)

Pada Tabel 4.2 analisis kebutuhan fungsional didapat 5 kebutuhan fungsional pada aktor wisatawan.

Tabel 4.3 Kebutuhan Fungsional Admin

No.	[A02] Admin		
	Kode	Nama Fungsi	Deskripsi
1.	KF-01	Melakukan *Login*	Sistem dapat digunakan untuk melakukan *login*.
2.	KF-02	Melakukan *Logout*	Sistem dapat digunakan untuk melakukan *logout.*
3.	KF-03	Melihat Halaman *Dashboard*	Sistem dapat digunakan untuk melihat halaman *dashboard* yang berisi grafik pendapatan penjualan tiket.
4.	KF-04	Menggelola Datta Tiket	Sistem dapat digunakan mengelola data tiket.
5.	KF-05	Menggelola Data Kegiatan	Sistem dapat digunakan mengelola data kegiatan.
6.	KF-06	Melihat Daftar Pemesaanan Tiket	Sistem dapat digunakan untuk melihat daftar pemesanan tiket.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 4.3 Kebutuhan Fungsional Admin (Lanjutan)

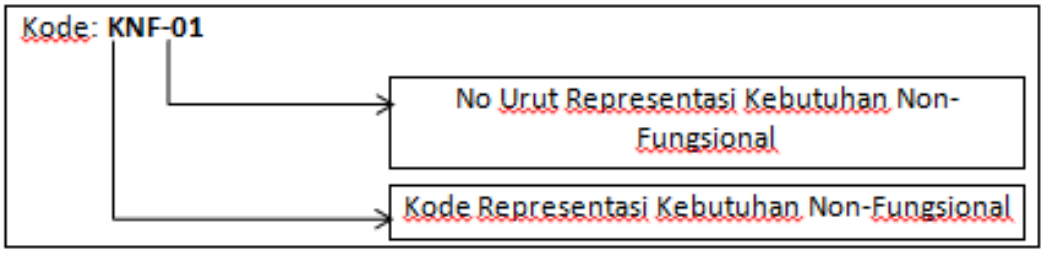
No.	[A02] Admin		
	Kode	Nama Fungsi	Deskripsi
7.	KF-07	Meengelola Data Daftar Pengguna	Sistem dapat digunakan untuk mengelola data daftar pengguna.
8.	KF-08	Melihat Laporan Keuangan	Sistem dapat digunakan untuk melihat laporan keuangan.
9.	KF-09	Mencetak Laporan Keuangan	Sistem dapat digunakan untuk mencetak laporan keuangan.

Sumber: (Penulis, 2019)

Pada Tabel 4.3 analisis kebutuhan fungsional didapat 9 kebutuhan fungsional pada aktor admin.

4.3.3 Kebutuhan Non-Fungsional

Kebutuhan non fungsional merupakan kebutuhan yang mengacu pada kinerja sebuah sistem. Kebutuhan non-fungsional sistem pada aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* ditunjukkan pada Tabel 4.4. Pada masing-masing kebutuhan non-fungsional dilakukan penomoran sesuai dengan aturan penomoran yang terdapat pada Gambar 4.7.



Gambar 4.7 Aturan Penomoran Kebutuhan Non-Fungsional

Sumber: (Penulis, 2019)

Pada Gambar 4.7 terdapat kode KNF-01 yang merupakan aturan penomoran aktor kebutuhan non-fungsional.

Tabel 4.4 Kebutuhan Non-Fungsional

No.	Kode	Nama Fungsi	Deskripsi
1.	KNF-01	*Compatibility*	Sistem dapat diakses pada browser yang berberda – beda.

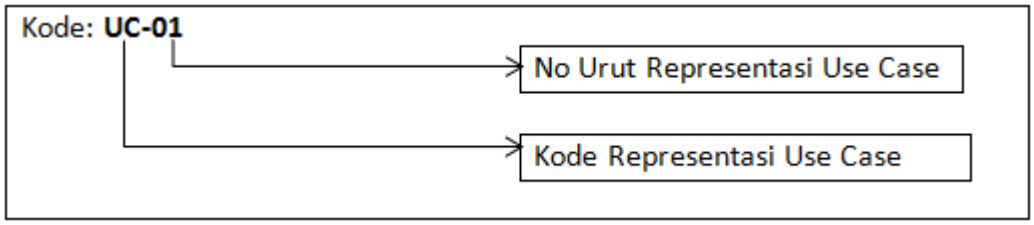
Sumber: (Penulis, 2019)

Berdasarkan Tabel 4.4 kebutuhan non-fungsional aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* memiliki 1 kebutuhan non-fungsional yakni *compatibility*.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

4.3.4 Pemodelan Use Case

Pada *use case diagram* ini memberikan gambaran mengenai siapa saja pengguna yang dapat menggunakan sistem dan apa saja yang dapat pengguna lakukan didalam sistem tersebut. Pada masing-masing *use case* dilakukan penomoran sesuai dengan aturan penomoran yang terdapat pada Gambar 4.8.

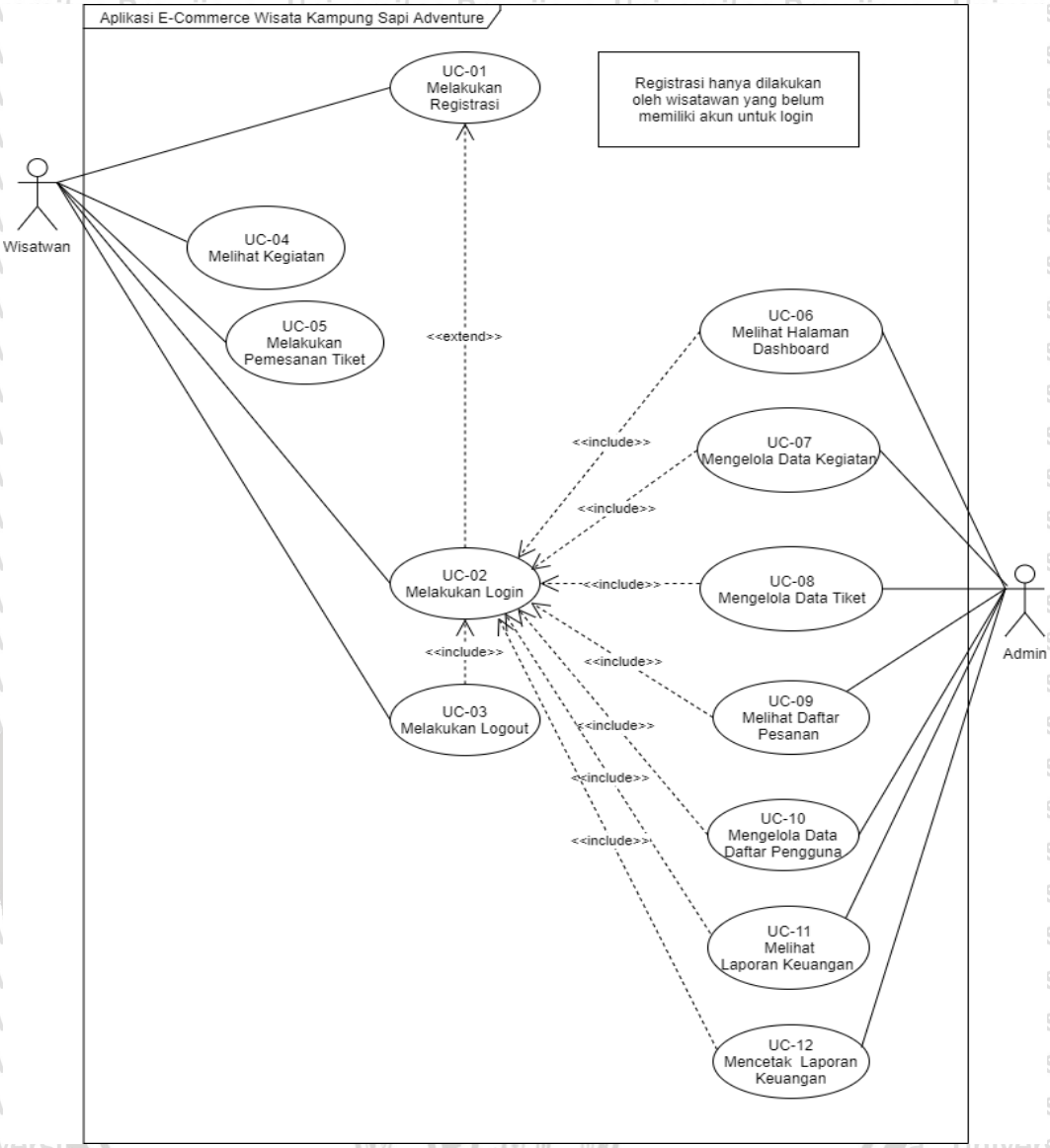


Gambar 4.8 Aturan Penomoran Pada *Use Case Diagram*

Sumber: (Penulis, 2019)

Pada Gambar 4.8 terdapat kode UC-01 yang merupakan aturan penomoran *use case diagram*.

Berikut *use case diagram* aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* yang dijelaskan pada Gambar 4.9. Aktor dari aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* adalah wisatawan dan admin. Aktivitas yang dapat dilakukan oleh aktor wisatawan, yaitu melakukan register, melakukan *login,* melakukan *logout*, melihat kegiatan, dan melakukan pemesanan tiket. Sedangkan aktivitas yang dilakukan oleh aktor admin, yaitu melakukan *login*, melakukan *logout,* melihat halaman *dashboard admin,* mengelola data kegiatan, melihat laporan keuangan, mencetak laporan keuangan, melihat daftar pemesanan tiket, mengelola data tiket, dan mengelola data daftar pengguna.



Gambar 4.9 *Use Case Diagram* Aplikasi *E-commerce* Wisata Kampung Sapi *Adventure*

Sumber: (Penulis, 2019)

Pada Tabel 4.5 dibawah ini berisi no, kode *use case*, aktor, dan nama dari *use case diagram* yang telah dibuat berdasarkan pada Gambar 4.8.

Tabel 4.5 *Use Case Diagram* Aplikasi *E-commerce* Wisata Kampung Sapi *Adventure*

No	Kode	Aktor	Nama *Use Case*
1.	UC01	Wisatawan	Melakukan Register
2.	UC02	Wisatawan dan Admin	Melakukan *Login*
3.	UC-03	Wisatawan dan Admin	Melakukan *Logout*
4.	UC-04	Wisatawan	Melihat Kegiatan

Tabel 4.5 *Use Case Diagram* Aplikasi *E-commerce* Wisata Kampung Sapi *Adventure* (Lanjutan)

No	Kode	Aktor	Nama *Use Case*
5.	UC-05	Wisatawan	Melakukan Pemesanan Tiket
6.	UC-06	Admin	Melihat Halaman *Dashboard* Admin
7.	UC-07	Admin	Mengelola Data Kegiatan
8.	UC-08	Admin	Mengelola Data Tiket
9.	UC-09	Admin	Melihat Daftar Pemesanan Tiket
10.	UC-10	Admin	Mengelola Data Daftar Pengguna
11.	UC-11	Admin	Melihat Laporan Keuangan
12.	UC-12	Admin	Mencetak Laporan Keuangan

Sumber: (Penulis, 2019)

4.4 Spesifikasi *Use Case*

Pada pembuatan spesifikasi *use case* ini dibuat berdasarkan *use case diagram*. Berikut ini adalah spesifikasi *use case* aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* berdasarkan aktor wisatawan yang dapat dilihat pada Tabel 4.6 sampai Tabel 4.10 .

Tabel 4.6 Spesifikasi *Use Case* Melakukan Register

Spesifikasi *Use Case* Melakukan Register	
Kode *Use Case*	UC-01
Actor	Wisatawan
Description	Skenario untuk melakukan register
Usecase Goal	Aktor wisatawan dapat melakukan register
Preconditions	Aktor wisatawan berada pada halaman utama aplikasi *e-commerce* Wisata Kampung Sapi *Adventure*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 4.6 Spesifikasi *Use Case* Melakukan Register (Lanjutan)

Spesifikasi *Use Case* Melakukan Register	
Kode *Use Case*	UC-01
Main Flow	1. Untuk melakukan register langkah pertama yang dilakukan aktor wisatawan yaitu memilih menu register 2. Sistem akan menampilkan halaman register 3. Aktor wisatawan melakukan register dengan mengisi data diri *username*, email, dan *password* 4. Aktor melakukan verifikasi email 5. Sistem memverifikasi email 6. Sistem menampilkan halaman login
Alternative Flow	Mengatasi kegagalan saat melakukan verifikasi email yang terdapat pada *Main Flow* langkah ke 4: 1. Jika register gagal maka sistem akan menampilkan informasi bahwa register gagal dan aktor wisatawan diminta untuk melakukan register kembali.
Postconditions	Aktor wisatawan berhasil melakukan register

Sumber: (Penulis, 2019)

Pada Tabel 4.6 menjelaskan spesifikasi *use case* register. Pada spesifikasi *use case* register menggambarkan alur penggunaa untuk melakukan register dengan menginputkan *username*, email dan *password.*

Tabel 4.7 Spesifikasi *Use Case* Melakukan Login

Spesifikasi *Use Case* Melakukan *Login*	
Kode *Use Case*	UC-02
Actor	Wisatawan
Description	Skenario untuk melakukan *login*
Usecase Goal	Aktor wisatawan dapat melakukan *login*
Preconditions	Aktor wisatawan berada pada halaman utama aplikasi *e-commerce* Wisata Kampung Sapi *Adventure*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 4.7 Spesifikasi *Use Case* Melakukan Login (Lanjutan)

Spesifikasi *Use Case* Melakukan *Login*	
Kode *Use Case*	UC-02
Main Flow	1. Untuk melakukan *login* langkah pertama yang dilakukan aktor wisatawan yaitu memilih menu *login* 2. Sistem menampilkan halaman *login* 3. Aktor wisatawan menginputkan *username* dan *password* yang telah terverifikasi 4. Sistem mengidentifikasi akun pengguna berdasarkan *username* dan *password* yang diinputkan pengguna 5. Aktor wiasatawan berhasil masuk ke dalam sistem
Alternative Flow	Mengatasi kegagalan saat melakukan *login* yang terdapat pada *Main Flow* langkah ke 4: 1. Jika *username* dan *password* yang diinputkan tidak sesuai dengan akun yang telah diverifikasi maka sistem akan menampilkan pesan bahwa *login* gagal.
Postconditions	Aktor wisatawan berhasil melakukan *login*

Sumber: (Penulis, 2019)

Pada Tabel 4.7 menjelaskan spesifikasii *use case login.* Pada spesifikasi *use case login* menggambarkan alur pengguna untuk masuk ke dalam sistem dan menggunakan sistem.

Tabel 4.8 Spesifikasi *Use Case* Melakukan *Logout*

Spesifikasi *Use Case* Melakukan *Logout*	
Kode *Use Case*	UC-03
Actor	Wisatawan
Description	Skenario untuk melakukan *logout*
Usecase Goal	Aktor wisatawan dapat melakukan *logout*
Preconditions	Aktor wisatawan sudah masuk ke dalam sistem aplikasi *e-commerce* Wisata Kampung Sapi *Adventure*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 4.8 Spesifikasi *Use Case* Melakukan *Logout* (Lanjutan)

Spesifikasi *Use Case* Melakukan *Logout*	
Kode *Use Case*	UC-03
Main Flow	1. Untuk melakukan *logout* langkah pertama yang dilakukan aktor wisatawan yaitu memilih menu *logout* 2. Sistem menghapus *authentication* pengguna 3. Sistem menampilkann *dashboard* aplikasi e-commerce Wisata Kampung Sapi *Adventure*
Alternative Flow	-
Postconditions	Aktor wisatawan berhasil melakukan *logout*

Sumber: (Penulis, 2019)

Pada Tabel 4.8 merupakan spesifikasi *use case logout.* Pada spesifikasi *use case logout* menggambarkan alur pengguna untuk keluar dari sistem.

Tabel 4.9 Spesifikasi *Use Case* Melihat Kegiatan

Spesifikasi *Use Case* Melihat Kegiatan	
Kode *Use Case*	UC-04
Actor	Wisatawan
Description	Skenario untuk melihat kegiatan.
Usecase Goal	Wisatawan dapat melihat kegiatan.
Preconditions	Aktor wisatawan berada pada halaman utama aplikasi *e-commerce* Wisata Kampung Sapi *Adventure*
Main Flow	1. Untuk melihat kegiatan langkah pertama yang dilakukan aktor wisatawan yaitu memilih menu kegiatan 2. Sistem menampilkan halaman kegiatan.
Alternative Flow	-
Postconditions	Aktor wisatawan berhasil melihat kegiatan.

Sumber: (Penulis, 2019)

Pada Tabel 4.9 merupakan spesifikasi *use case* melihat kegiatan. Pada spesifikasi *use case* melihat kegiatan menggambarkan alur pengguna untuk melihat kegiatan yang dilakukan pada Wisata Kampung Sapi *Adventure*.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 4.10 Spesifikasi *Use Case* Melakukan Pemesanan Tiket

Spesifikasi *Use Case* Melakukan Pemesanan Tiket	
Kode *Use Case*	UC-05
Actor	Wisatawan
Description	Skenario untuk melakukan pemesanan tiket.
Usecase Goal	Aktor wisatawan dapat melakukan pemesanan tiket.
Preconditions	Aktor wisatawan berada pada halaman utama aplikasi *e-commerce* Wisata Kampung Sapi *Adventure*
Main Flow	1. Untuk melakukan pemesanan tiket langkah pertama yang dilakukan aktor wisatawan yaitu memilih menu pesan tiket 2. Sistemm menampilkan kategori tiket 3. Aktor wisatawan memilih kategori tiket untuk hari ini (*go show*) 4. Sistem menampilkan paket sesuai dengann kategori tiket yang dipilih aktor wisatawan 5. Aktor wisatawan dapat mengetahui fasilitas dan kegiatan yang didapat ketika memesan paket yang dipilih 6. Aktor wisatawan mengetahui ketersediaan tiket 7. Aktor wisatawan dapat melakukan pembelian paket sesuai kategori tiket yang dipilih dengan menekan tombol beli 8. Sistem menampilkan data pembelian tiket 9. Aktor wisatawan menekan tombol lanjutkan pembayaran 10. Sistem mengecek *session* aktor wisatawan 11. Sistem menampilkan data pembayaran pemesanan tiket

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 4.10 Spesifikasi *Use Case* Melakukan Pemesanan Tiket (Lanjutan)

Spesifikasi *Use Case* Melakukan Pemesanan Tiket	
Kode *Use Case*	UC-05
Main Flow	12. Aktor memilih metode pembayaran 13. Sistem menampilkan cara pembayaran 14. Aktor menginputkan email untuk mengetahui cara status pembayaran 15. Sistem menampilkan jumlah yang dibayar, order ID, batas waktu pembayaran, dan nomor rekening bank yang dituju 16. Aktor menyelesaikan pembayaran dengan melakukan pembayaran pada simulator pembayaran pada midtrans 17. Sistem melakukan cek status transaksi 18. Sistem mengirim status pembayaran melalui email 19. Aktor mendapatkan elektronik tiket
Alternative Flow	*Main flow* langkah ke 3 mengetahui ketersediaan tiket : 1. Jika aktor memilih kategori tiket reservasi, maka aktor diharuskan untuk melakukan cek tanggal terlebih dahulu untuk mengetahui ketersediaan tiket *Main flow* langkah ke 10 melakukan pembayaran tiket: 1. Jika aktor belum melakukan *login*, maka aktor diharuskan melakukan *login* terlebih dahulu *Main flow* langkah ke 19 melakukan pembayaran tiket: 1. Jika status pembayaran gagal, maka aktor tidak mendapatkan elektronik tiket
Postconditions	Aktor wisatawan berhasil melakukan pemesanan tiket.

Sumber: (Penulis, 2019)

Pada Tabel 4.10 merupakan spesifikasi *use case* melakukan pemesanan tiket. Pada spesifikasi *use case* melakukan pemesanan tiket menggambarkan alur untuk melakukan pemesanan tiket dengan memilih kategori pemesanan tiket hari ini (*go Show*) maupun reservasi. Wisatawan dapat mengetahui fasilitas dan

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

kegiatan yang didapat ketika memesan paket yang dipilih. Wisatawan dapat mengetahui ketersediaan tiket. Wisatawan dapat melakukan pembelian paket sesuai dengan kategori tiket yang dipilih. Wisatawan dapat melakukan pembayaran tiket melalui midtrans.

Berikut ini adalah spesifikasi *use case* aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* berdasarkan aktor admin yang dapat dilihat pada Tabel 4.11 sampai Tabel 4.19.

Tabel 4.11 Spesifikasi *Use Case Login*

Spesifikasi *Use Case* Melakukan *Login*	
Kode *Use Case*	UC-02
Actor	Admin
Description	Skenario untuk melakukan *login*
Usecase Goal	Aktor admin dapat melakukan *login*
Kode *Use Case*	UC-02
Preconditions	Aktor admin berada pada halaman utama aplikasi *e-commerce* Wisata Kampung Sapi *Adventure*
Main Flow	1. Untuk melakukan *login* langkah pertama yang dilakukan aktor admin yaitu memilih menu *login* 2. Sistem menampilkann halaman *login* 3. Aktor admin menginputkan *username* dan *password* yang telah tersimpan di *database* 4. Sistem mengidentifikasi akun berdasarkan *username* dan *password* yangg diinputkann admin 5. Admin berhasil masuk ke dalam sistem
Alternative Flow	Mengatasi kegagalan saat melakukan *login* yang terdapat pada *Main Flow* langkah ke 4: 1. Jika *username* dan *password* yang diinputkan tidak sesuai dengan akun yang telah diverifikasi maka sistem akan menampilkan pesan bahwa *login* gagal
Postconditions	Aktor admin berhasil melakukan *login*

Sumber: (Penulis, 2019)

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Pada Tabel 4.11 menjelaskan spesifikasi *use case login.* Pada spesifikasi *use case login* menggambarkan alur admin untuk masuk ke dalam sistem dan menggunakan sistem.

Tabel 4.12 Spesifikasi *Use Case* Melakukan *Logout*

Spesifikasi *Use Case* Melakukan *Logout*	
Kode *Use Case*	UC-03
Actor	Admin
Description	Skenario untuk melakukan *logout*
Usecase Goal	Aktor admin dapat melakukan *logout*
Preconditions	Aktor admin berada pada halaman admin
Main Flow	1. Untuk melakukan *logout* langkah pertama yang dilakukan aktor admin yaitu memilih menu *logout* 2. Sistem menghapus *authentication* admin 3. Sistem menampilkann halaman utama aplikasi *e-commerce* Wisata Kampung Sapi *Adventure*
Alternative Flow	-
Postconditions	Aktor admin berhasil melakukan *logout*

Sumber: (Penulis, 2019)

Pada Tabel 4.12 merupakan spesifikasi *use case logout.* Pada spesifikasi *use case logout* pada admin menggambarkan alur admin untuk keluar dari halaman admin.

Tabel 4.13 Spesifikasi *Use Case* Melihat Halaman *Dashboard* Admin

Spesifikasi *Use Case* Melihat Halaman *Dashboard* Admin	
Kode *Use Case*	UC-05
Actor	Admin
Description	Skenario untuk melihat halaman *dashboard*
Usecase Goal	Aktor admin dapat melihat halaman *dashboard*
Preconditions	Aktor admin berada pada halaman admin

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 4.13 Spesifikasi *Use Case* Melihat Halaman *Dashboard* Admin (Lanjutan)

Spesifikasi *Use Case* Melihat Halaman *Dashboard* Admin	
Kode *Use Case*	UC-05
Main Flow	1. Untuk melihat halam *dashboard* yang dilakukan aktor admin yaitu memilihh menuu *dashboard* 2. Sistem menampilkann halaman *dashboard* berupa line chart pendapatan penjualan tiket
Alternative Flow	-
Postconditions	Aktor admin berhasil melihat halaman dashboard

Sumber: (Penulis, 2019)

Pada Tabel 4.13 merupakan spesifikasi *use case* melihat halaman *dashboard.* Pada spesifikasi *use case* melihat halaman *dashboard* menggambarkan alur admin untuk melihat halaman *dashboard* admin.

Tabel 4.14 Spesifikasi *Use Case* Mengelola Data Kegiatan

Spesifikasi *Use Case* Mengelola Data Kegiatan	
Kode *Use Case*	UC-07
Actor	Admin
Description	Skenario untuk mengelola data kegiatan
Usecase Goal	Aktor admin dapat *create, read*, *update*, dan *delete* pada data kegiatan
Preconditions	Aktor admin berada pada halaman admin
Main Flow	1. Untuk mengelola kegiatan langkah pertama yang dilakukan aktor admin yaitu memilih menu kegiatan 2. Sistem menampilkann halaman kegiatan 3. Aktor admin dapat melakukan *create, read*, *update*, dan *delete* pada kegiatan 4. Sistem menampilkan aksi dari aktor admin

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 4.14 Spesifikasi *Use Case* Mengelola Data Kegiatan (Lanjutan)

Spesifikasi *Use Case* Mengelola Data Kegiatan	
Kode *Use Case*	UC-07
Alternative Flow	-
Postconditions	Aktor admin berhasil melakukan *create, read*, *update*, dan *delete* kegiatan

Sumber: (Penulis, 2019)

Pada Tabel 4.14 merupakan spesifikasi *use case* mengelola data kegiatan. Pada spesifikasi *use case* mengelola data kegiatan menggambarkan alur admin untuk melakukan *create, read*, *update*, dan *delete* pada data kegiatan.

Tabel 4.15 Spesifikasi *Use Case* Mengelola Data Tiket

Spesifikasi *Use Case* Mengelola Data Tiket	
Kode *Use Case*	UC-08
Actor	Admin
Description	Skenario untuk mengelola data tiket
Usecase Goal	Aktor admin dapat melakukan *create, read*, *update*, dan *delete* pada tiket
Preconditions	Aktor admin berada pada halaman admin
Main Flow	1. Untuk mengubah data tiket langkah pertama yang dilakukan aktor admin yaitu memilih menu tiket 2. Sistem menampilkann halaman tiket 3. Aktorr dapat melakukan *create, read*, *update*, dan *delete* pada tiket 4. Sistem menampilkan aksi dari aktor admin
Alternative Flow	-
Postconditions	Aktor admin berhasil melakukan *create, read*, *update*, dan *delete* kegiatan

Sumber: (Penulis, 2019)

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Pada Tabel 4.15 merupakan spesifikasi *use case* mengelola data tiket. Pada spesifikasi *use case* mengelola data tiket menggambarkan alur admin untuk melakukan *create, read*, *update*, dan *delete* pada data tiket

Tabel 4.16 Spesifikasi *Use Case* Melihat Daftar Pemesanan

Spesifikasi *Use Case* Melihat Daftar Pemesanan	
Kode *Use Case*	UC-09
Actor	Admin
Description	Skenario untuk melihat daftar pemesanan
Usecase Goal	Aktor admin dapat melihat daftar pemesanan
Preconditions	Aktor admin berada pada halaman admin
Main Flow	1. Untuk melihat daftar pemesanan langkah pertama yang dilakukan aktor admin yaitu aktor admin memilih menu daftar pemesanan 2. Sistem menampilkan halaman daftar pemesanan 3. Aktor admin melihat daftar pemesanan
Kode *Use Case*	UC-09
Alternative Flow	-
Postconditions	Aktor admin berhasil melihat daftar pemesanan

Sumber: (Penulis, 2019)

Pada Tabel 4.16 merupakan spesifikasi *use case* melihat daftar pemesanan. Pada spesifikasi *use case* pemesanan menggambarkan alur admin untuk melihat daftar pemesanan.

Tabel 4.17 Spesifikasi *Use Case* Mengelola Data Daftar Pengguna

Spesifikasi *Use Case* Mengelola Data Daftar Pengguna	
Kode *Use Case*	UC-10
Actor	Admin
Description	Skenario untuk mengelola data daftar pengguna
Usecase Goal	Aktor admin dapat melihat dan menghapus pada daftar pengguna
Preconditions	Aktor admin berada pada halaman admin

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 4.17 Spesifikasi *Use Case* Mengelola Data Daftar Pengguna (Lanjutan)

Spesifikasi *Use Case* Mengelola Data Daftar Pengguna	
Main Flow	1. Untuk melihat daftar pengguna langkah pertama yang dilakukan aktor admin yaitu aktor admin memilih menu daftar pengguna 2. Sistem menampilkan halaman daftar pengguna 3. Aktor admin dapat melihat daftar pengguna yang telah melakukan register 4. Aktor admin menghapus daftar pengguna dengan menekan tombol hapus 5. Sistem menghapus daftar pengguna
Alternative Flow	-
Postconditions	Aktor admin berhasil melihat dan menghapus daftar pengguna

Sumber: (Penulis, 2019)

Pada Tabel 4.17 merupakan spesifikasi *use case* mengelola data daftar pengguna. Pada spesifikasi *use case* melihat pengguna menggambarkan alur admin untuk melihat dan menghapus data daftar pengguna.

Tabel 4.18 Spesifikasi *Use Case* Melihat Laporan Keuangan

Spesifikasi *Use Case* Melihat Laporan Keuangan	
Kode *Use Case*	UC-11
Actor	Admin
Description	Skenario untuk melihat laporan keuangan
Usecase Goal	Aktor admin dapat melihat laporan keuangan
Preconditions	Aktor admin berada pada halaman admin
Main Flow	1. Untuk melihat laporan keuangan langkah pertama yang dilakukan aktor admin yaitu aktor admin memilih menu laporan keuangan 2. Sistem menampilkan halaman laporan keuangan

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 4.18 Spesifikasi *Use Case* Melihat Laporan Keuangan (Lanjutan)

Spesifikasi *Use Case* Melihat Laporan Keuangan	
Kode *Use Case*	UC-11
Main Flow	3. Aktor admin dapat melihat laporan keuangan
Alternative Flow	-
Postconditions	Aktor admin berhasil melihat laporan keuangan

Sumber: (Penulis, 2019)

Pada Tabel 4.18 merupakan spesifikasi *use case* melihat laporan keuangan. Pada spesifikasi *use case* melihat laporan menggambarkan alur admin untuk melihat laporan keuangan.

Tabel 4.19 Spesifikasi *Use Case* Mencetak Laporan Keuangan

Spesifikasi *Use Case* Mencetak Laporan Keuangan	
Kode *Use Case*	UC-12
Actor	Admin
Description	Skenario untuk mencetak laporan keuangan
Usecase Goal	Aktor admin dapat mencetak laporan keuangan
Preconditions	Aktor admin berada pada halaman laporan keuangan
Main Flow	1. Untuk mencetak laporan keuangan langkah pertama yang dilakukan aktor admin yaitu aktor admin memilih menu laporan keuangan 2. Sistem menampilkan laporan keuangan 3. Aktor admin memilih tombol print 4. Laporan keuangan terunduh
Alternative Flow	-
Postconditions	Aktor admin berhasil mencetak laporan keuangan

Sumber: (Penulis, 2019)

Pada Tabel 4.19 merupakan spesifikasi *use case* mencetak laporan keuangan. Pada spesifikasi *use case* mencetak laporan keuangan menggambarkan alur admin untuk mencetak laporan keuangan.

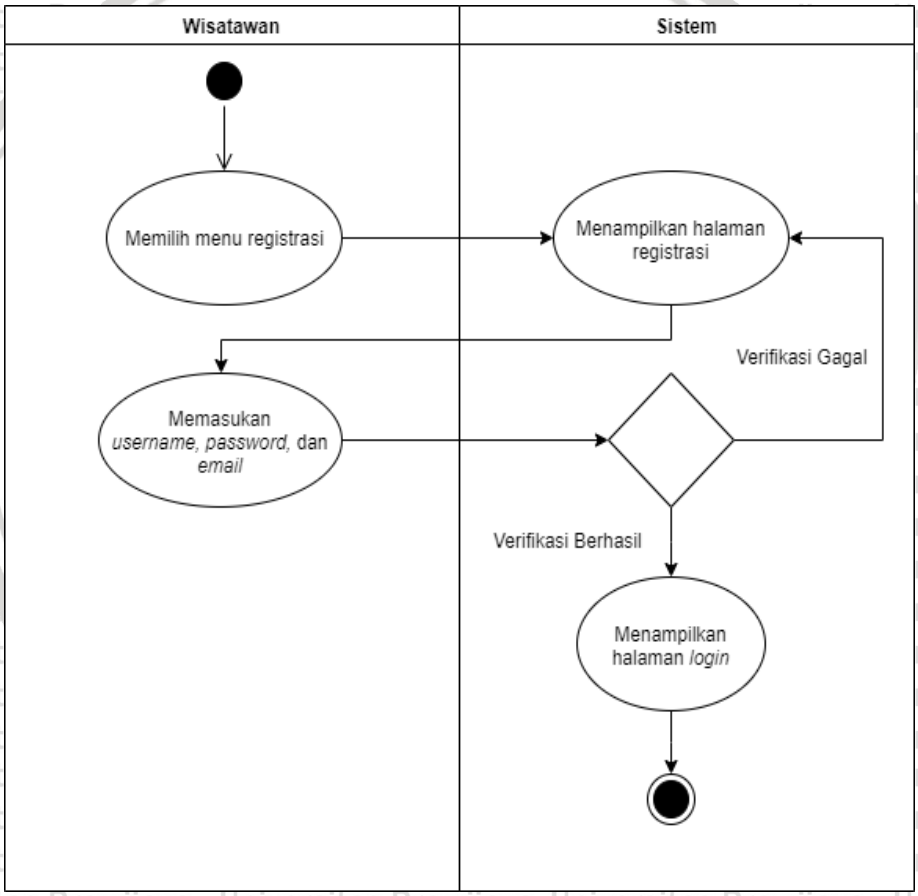
repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

4.5 *Activity Diagram*

Activity Diagram menggambarkan tentang aktivitas yang terjadi pada sistem. *Activity Diagram* aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* dijelaskan pada Gambar 4.10 sampai dengan Gambar 4.22.

4.5.1 *Activity Diagram* Register

Pada Gambar 4.10 menunjukkan *activity diagram* untuk melakukan register yang dilakukan oleh wisatawan. Aliran kerja atau aktivitas yang terjadi dimulai dari wisatawan memilih menu register. Kemudian sistem akan menampilkan halaman register. Wisatawan menginputkan *username, password*, dan email. Ketika wisatawan melakukan verifikasi namun gagal, maka sistem akan menampilkan halaman register kembali. Ketika wisatawan telah melakukan verifikasi dan statusnya berhasil, maka sistem akan menampilkan halaman *login*.



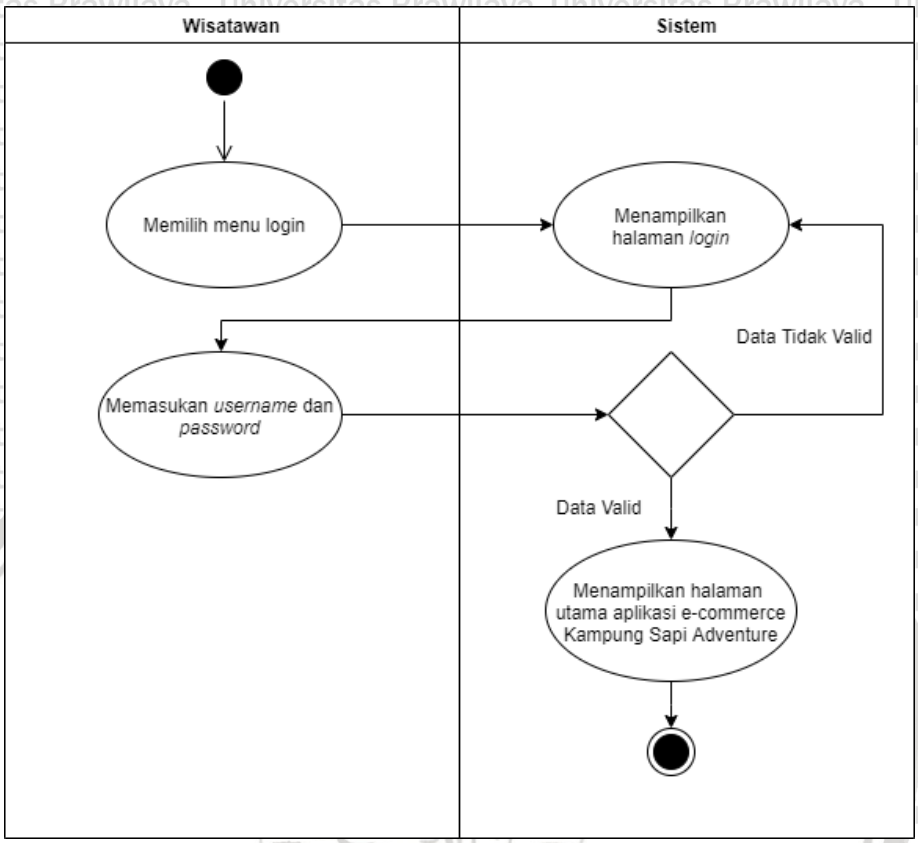
Gambar 4.10 *Activity Diagram* Register

Sumber: (Penulis, 2019)

4.5.2 *Activity Diagram Login*

Pada Gambar 4.11 menunjukkan *activity diagram* untuk melakukan *login* yang dilakukan oleh wisatawan. Aliran kerja atau aktivitas yang terjadi dimulai dari wisatawan memilih menu *login*. Kemudian sistem akan menampilkan halaman *login*. Wisatawan menginputkan *username* dan *password*. Ketika

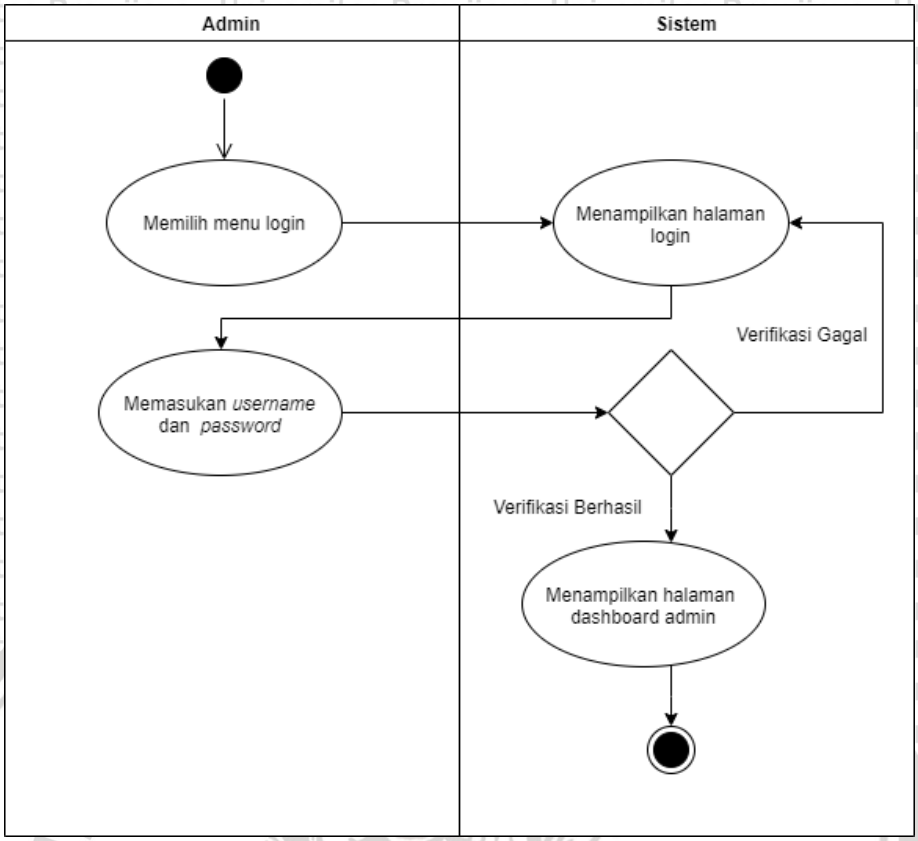
wisatawan melakukan *login* namun gagal, maka sistem akan menampilkan halaman *login* kembali. Ketika wisatawan berhasil melakukan *login*, maka sistem akan menampilkan halaman *dashboard* wisatawan.



Gambar 4.11 *Activity Diagram Login* Pada Wisatawan

Sumber: (Penulis, 2019)

Pada Gambar 4.12 menunjukkan *activity diagram* untuk melakukan *login* yang dilakukan oleh admin. Aliran kerja atau aktivitas yang terjadi dimulai dari admin memilih menu *login*. Kemudian sistem akan menampilkan halaman *login*. Admin menginputkan *username* dan *password*. Ketika admin melakukan *login* namun gagal, maka sistem akan menampilkan halaman *login* kembali. Ketika admin berhasil melakukan *login*, maka sistem akan menampilkan halaman *dashboard* admin.

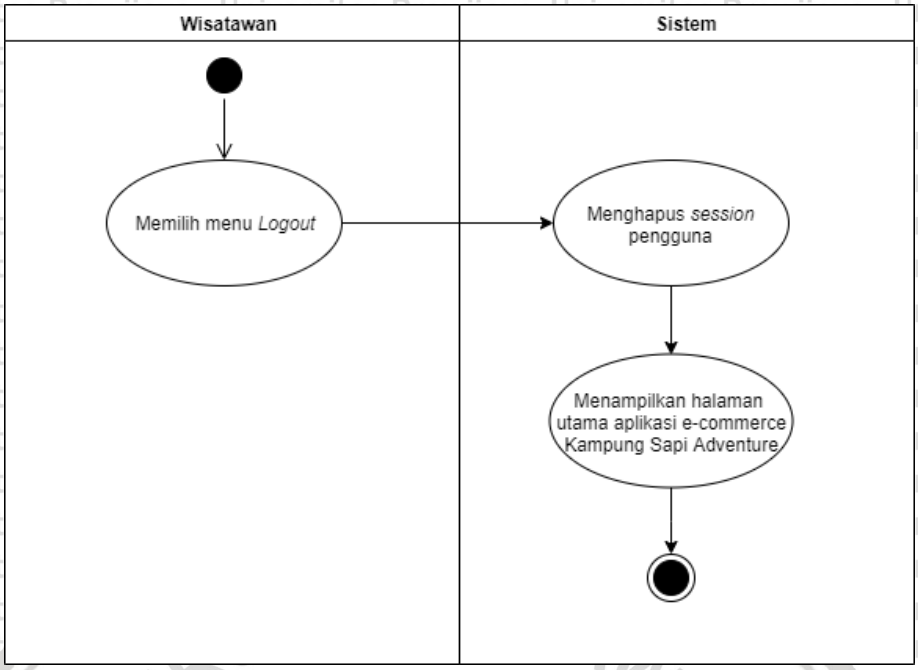


Gambar 4.12 *Activity Diagram Login* Pada Admin

Sumber: (Penulis, 2019)

4.5.3 *Activity Diagram Logout*

Pada Gambar 4.13 menunjukkan *activity diagram* untuk melakukan *logout* yang dilakukan oleh wisatawan dan admin. Aliran kerja atau aktivitas yang terjadi dimulai dari memilih menu *logout*. Kemudian sistem akan menghapus session pengguna. Setelah itu, sistem akan menampilkan halaman utama aplikasi *e-commerce* Kampung Sapi *Adventure*.

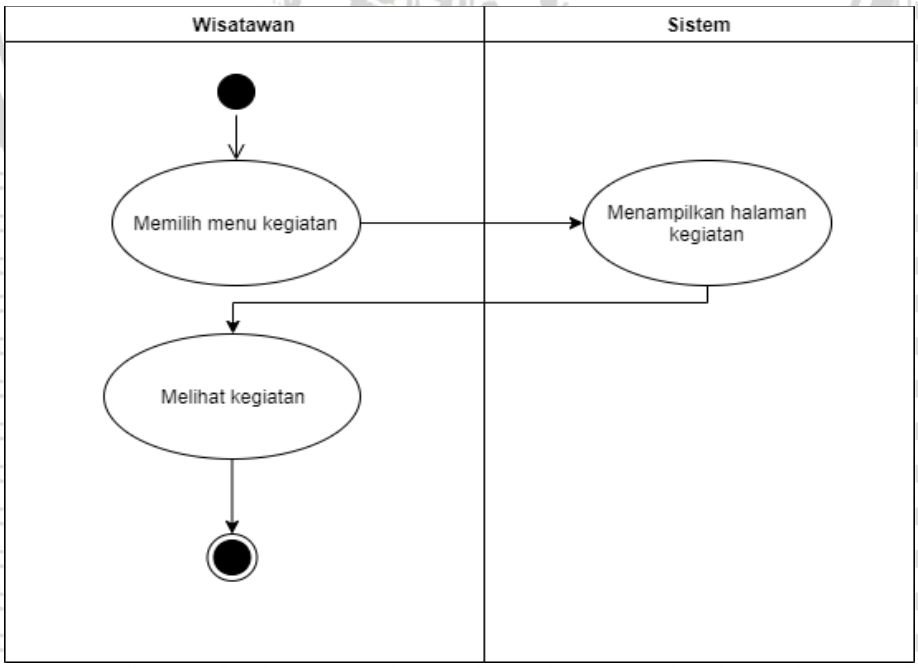


Gambar 4.13 *Activity Diagram Logout* Pada Wisatawan Dan Admin

Sumber: (Penulis, 2019)

4.5.4 *Activity Diagram* Melihat Kegiatan

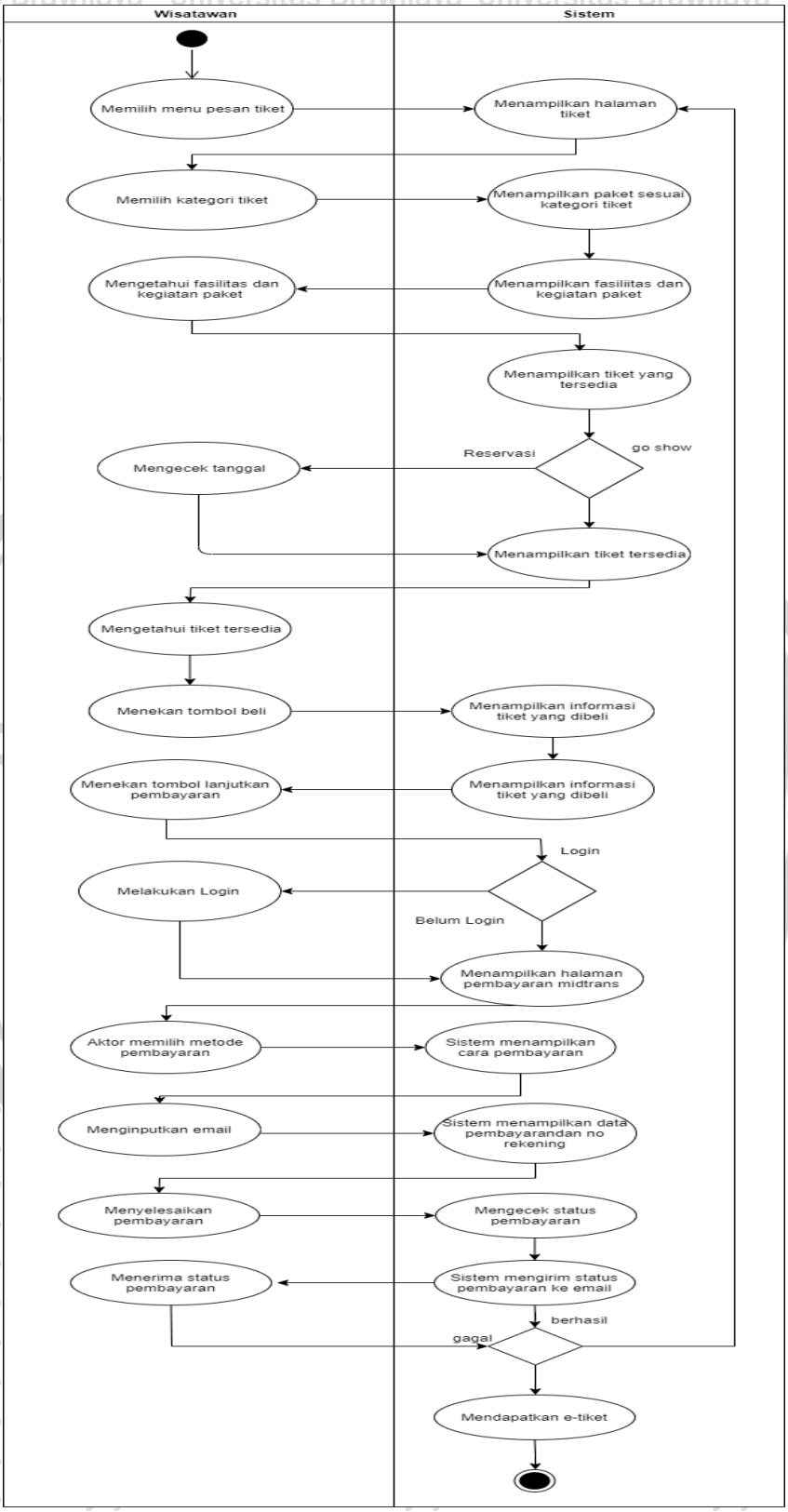
Pada Gambar 4.14 menunjukkan *activity diagram* untuk melihat kegiatan yang dilakukan oleh wisatawan. Aliran kerja atau aktivitas yang terjadi dimulai dari wisatawan memilih menu kegiatan. Kemudian sistem akan menampilkan halaman data kegiatan pada wisatawan. Wisatawan dapat melihat kegiatan.



Gambar 4.14 *Activity Diagram* Melihat Kegiatan

Sumber: (Penulis, 2019)

4.5.5 *Activity Diagram* Melakukan Pemesanan Tiket

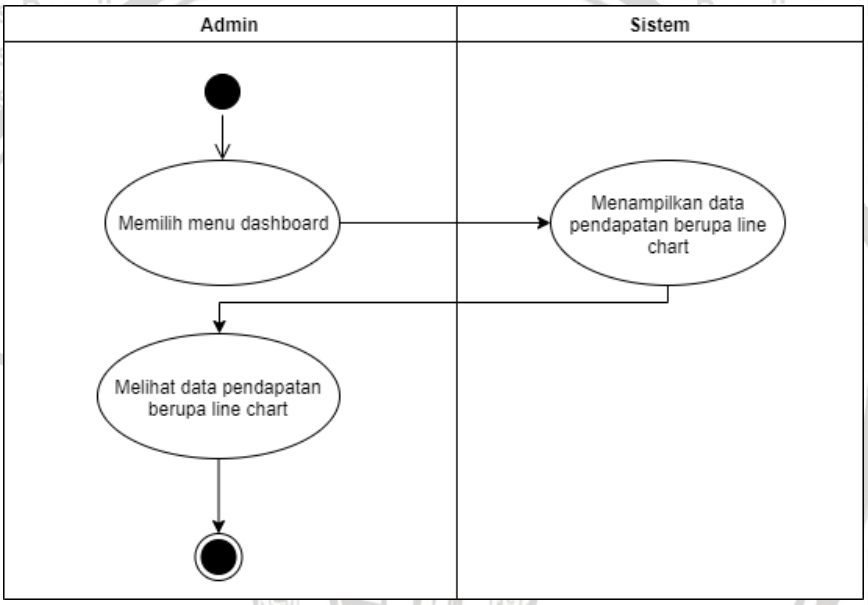


Gambar 4.15 *Activity Diagram* Melakukan Pemesanan Tiket

Sumber: (Penulis, 2019)

Pada Gambar 4.15 menunjukkan *activity diagram* untuk melakukan pemesanan tiket yang dilakukan oleh wisatawan. Aliran kerja atau aktivitas yang terjadi dimulai dari wisatawan memilih menu pesan tiket. Kemudian sistem akan menampilkan halaman tiket. Setelah itu, wisatawan memilih kategori tiket, maka sistem akan menampilkan paket sesuai kategori tiket. Wisatawan dapat mengetahui fasilitas dan kegiatan sesuai paket yang dipilih. Selanjutnya wisatawan dapat mengetahui ketersediaan tiket. Kemudian wisatawan melakukan pembelian tiket. Wisatawan dapat melakukan pembayaran dengan melakukan *login* terlebih dahulu. Jika sebelumnya wisatawan telah melakukan *login*, maka wisatawan secara langsung dapat melakukan pembayaran tiket. Jika status pembayaran berhasil, maka wisatawan mendapatkan elektronik tiket.

4.5.6 *Activity Diagram* Melihat Halaman *Dashboard*

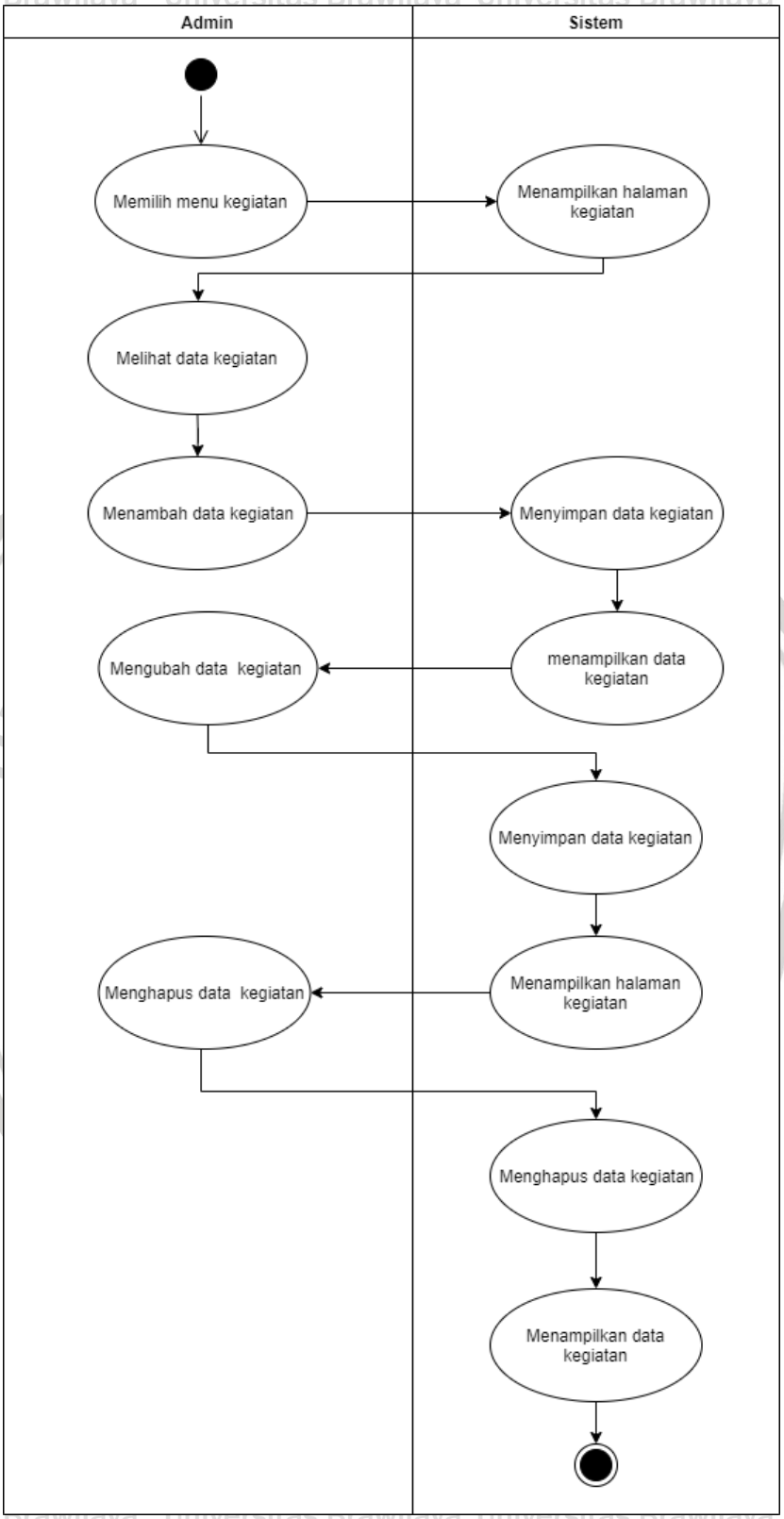


Gambar 4.16 *Activity Diagram* Melihat Halaman *Dashboard*

Sumber: (Penulis, 2019)

Pada Gambar 4.16 menunjukkan *activity diagram* untuk melihat halaman *dashboard* yang dilakukan oleh admin. Aliran kerja atau aktivitas yang terjadi dimulai dari admin memilih menu *dashboard*. Kemudian sistem akan menampilkan halaman *dashboard*. Pada halaman *dashboard* terdapat line chart mengenai data pendapatan penjualan tiket.

4.5.7 *Activity Diagram* Melakukan Mengelola Data Kegiatan



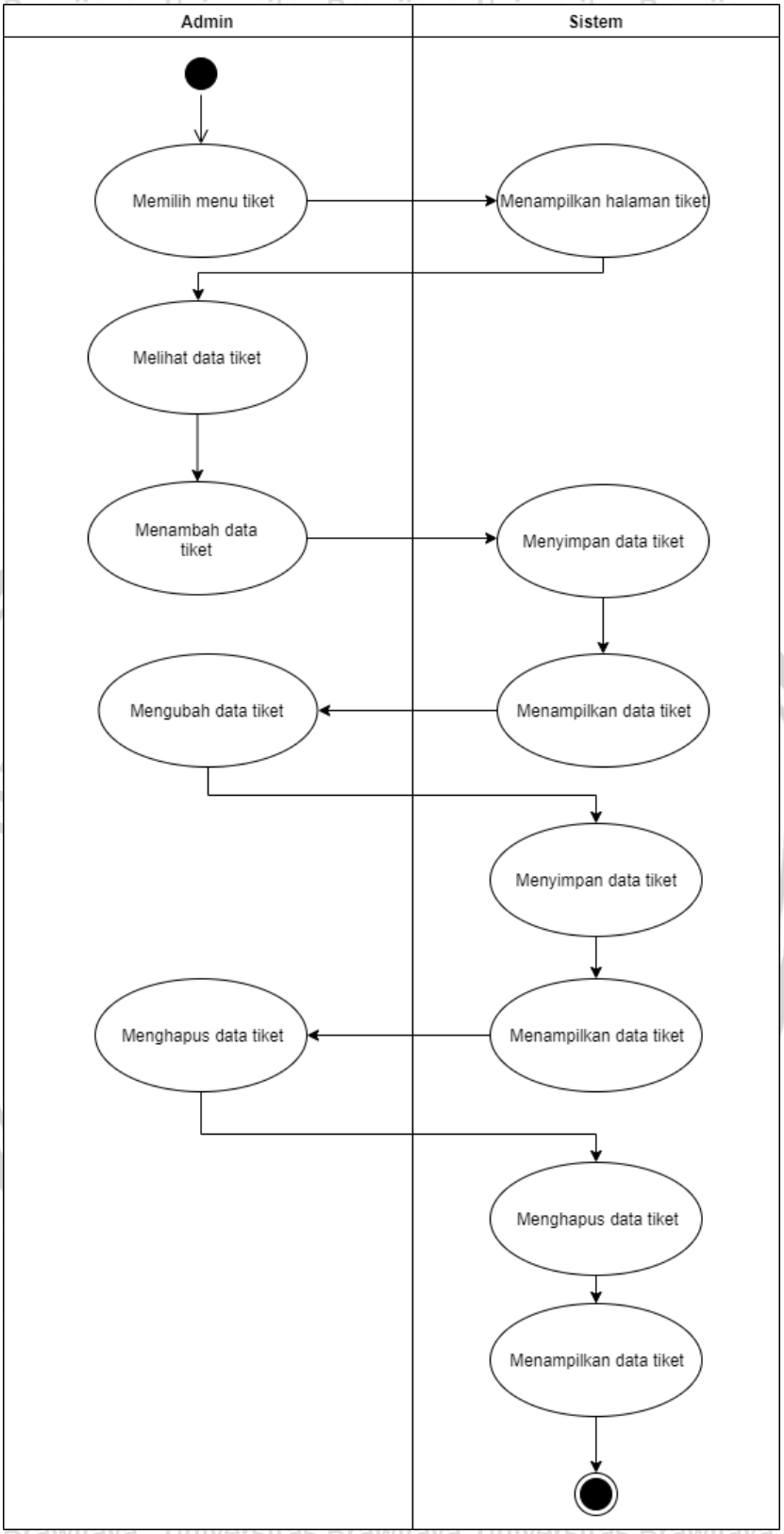
Gambar 4.17 *Activity Diagram* Melakukan Mengelola Data Kegiatan

Sumber: (Penulis, 2019)

Pada Gambar 4.17 menunjukkan *activity diagram* untuk mengelola data kegiatan yang dilakukan oleh admin. Aliran kerja atau aktivitas yang terjadi dimulai dari admin memilih menu kegiatan. Kemudian sistem akan menampilkan informasi halaman kegiatan pada admin dan admin dapat melihat kegiatan. Ketika admin menambahkan kegiatan, maka sistem akan menyimpan data kegiatan tersebut. Kemudian sistem akan menampilkan data kegiatan yang telah ditambahan oleh admin. Selanjutnya admin dapat mengubah data kegiatan, maka sistem akan menyimpan data kegiatan yang telah diubah tersebut. Setelah itu, sistem akan menampilkan kembali data kegiatan yang telah diubah tersebut. Ketika admin ingin menghapus kegiatan, maka sistem akan menghapus data kegiatan dan menampilkan kembali data kegiatan.

4.5.8 *Activity Diagram* Mengelola Data Tiket

Pada Gambar 4.18 menunjukkan *activity diagram* untuk mengelola data tiket yang dilakukan oleh admin. Aliran kerja atau aktivitas yang terjadi dimulai dari admin memilih menu tiket. Kemudian sistem akan menampilkan informasi halaman tiket pada admin dan admin dapat melihat tiket. Ketika admin menambahkan tiket, maka sistem akan menyimpan data tiket tersebut. Kemudian sistem akan menampilkan data kegiatan yang telah ditambahan oleh admin. Selanjutnya admin dapat mengubah data tiket, maka sistem akan menyimpan data tiket yang telah diubah tersebut. Setelah itu, sistem akan menampilkan kembali data tiket yang telah diubah tersebut. Ketika admin ingin menghapus tiket, maka sistem akan menghapus data tiket dan menampilkan kembali data tiket.

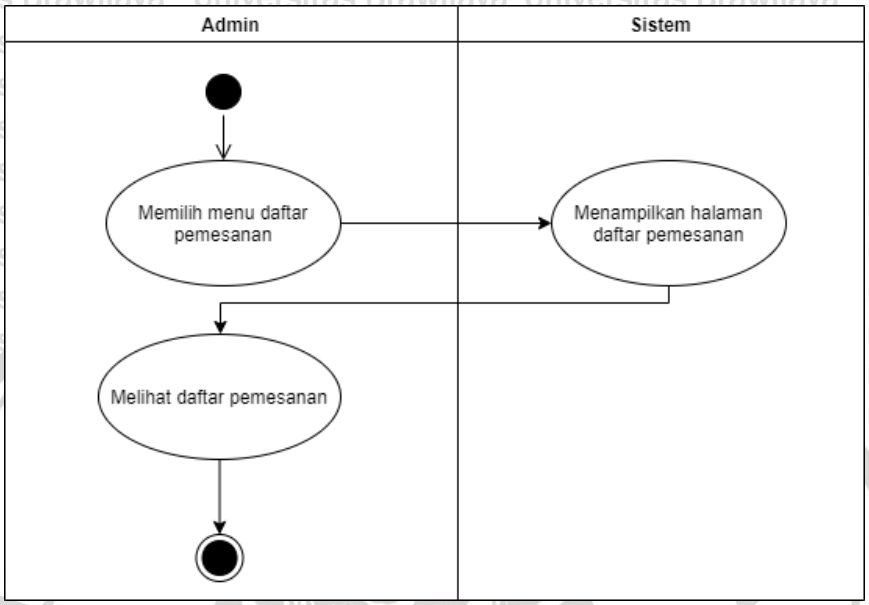


Gambar 4.18 *Activity Diagram* Mengelola Data Tiket

Sumber: (Penulis, 2019)

4.5.9 *Activity Diagram* Melihat Daftar Pemesanan

Pada Gambar 4.19 adalah *activity diagram* melihat daftar pemesanan yang dilakukan oleh admin. Aliran kerja atau aktivitas yang terjadi dimulai dari admin memilih menu daftar pemesanan. Kemudian sistem akan menampilkan halaman daftar pemesanan. Selanjutnya, wisatawan dapat melihat daftar pemesanan.

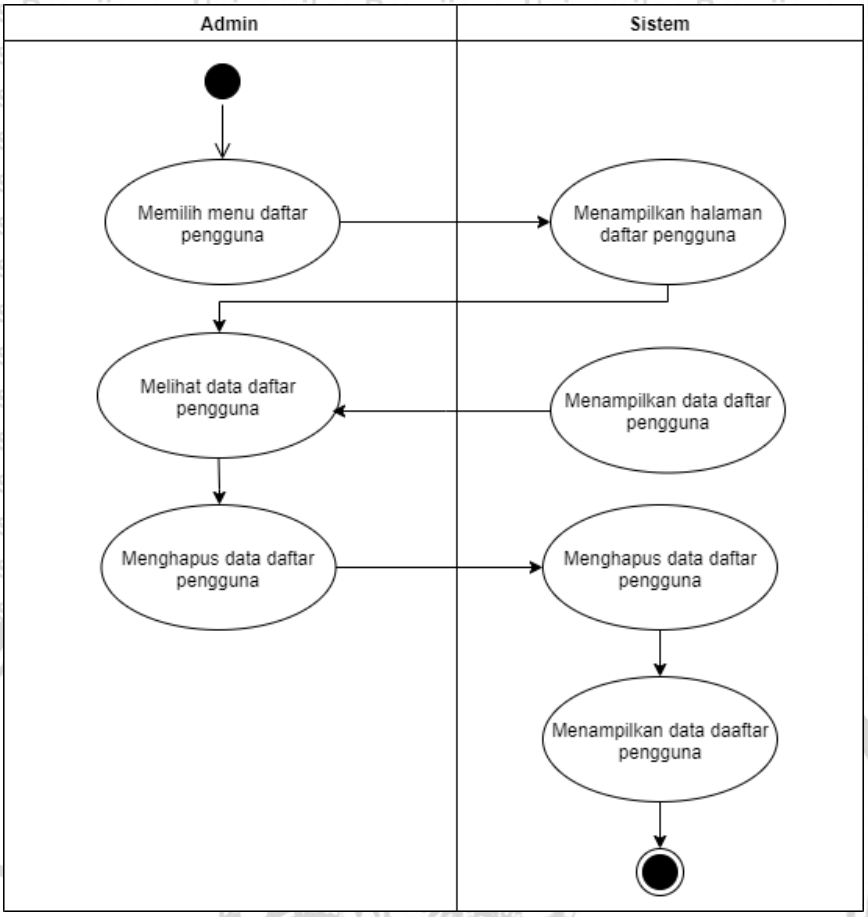


Gambar 4.19 *Activity Diagram* Melihat Daftar Pemesanan

Sumber: (Penulis, 2019)

4.5.10 *Activity Diagram* Mengelola Data Daftar Pengguna

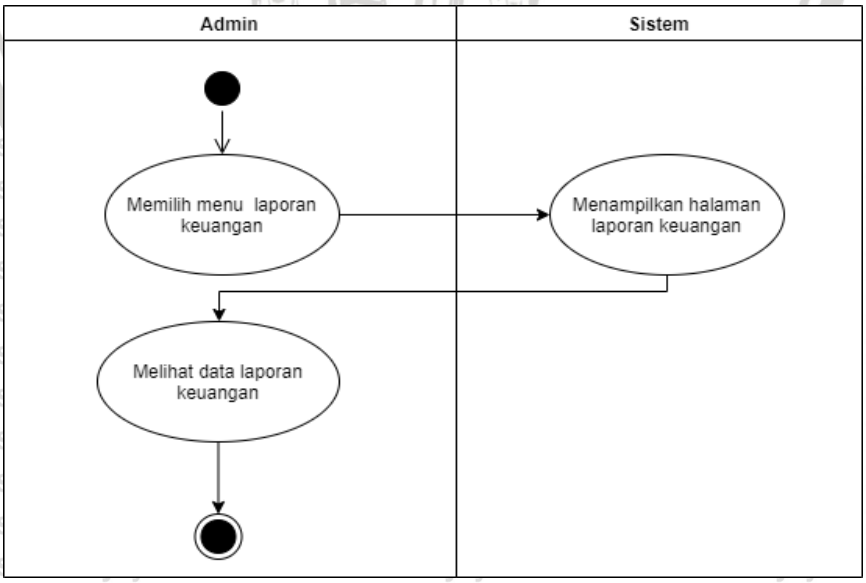
Pada Gambar 4.20 menunjukkan *activity diagram* untuk mengelola data pengguna yang dilakukan oleh admin. Aliran kerja atau aktivitas yang terjadi dimulai dari admin memilih menu daftar pengguna. Kemudian sistem akan menampilkan informasi halaman pengguna dan admin dapat melihat data pengguna. Ketika admin ingin menghapus data daftar pengguna, maka sistem akan menghapus data daftar pengguna dan menampilkan kembali data daftar pengguna.



Gambar 4.20 *Activity Diagram* Mengelola Data Daftar Pengguna

Sumber: (Penulis, 2019)

4.5.11 *Activity Diagram* Melihat Laporan Keuangan

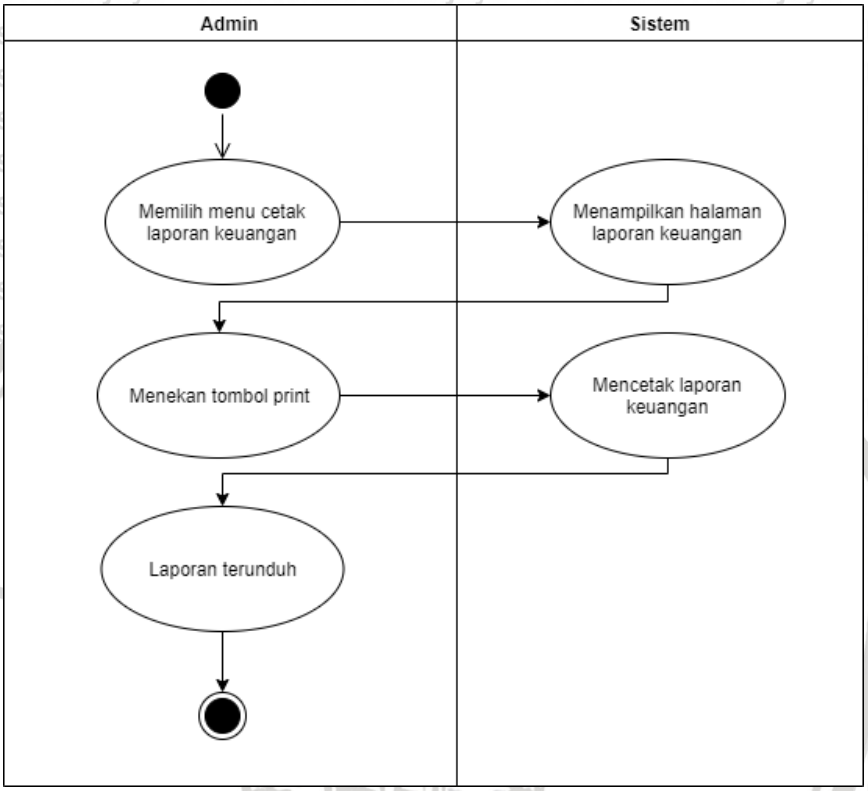


Gambar 4.21 *Activity Diagram* Melihat Laporan Keuangan

Sumber: (Penulis, 2019)

Pada Gambar 4.21 menunjukkan *activity diagram* untuk melihat laporan keuangan yang dilakukan oleh admin. Aliran kerja atau aktivitas yang terjadi dimulai dari admin memilih menu laporan keuangan. Kemudian sistem akan menampilkan informasi. Setelah itu, admin dapat melihat data laporan keuangan.

4.5.12 *Activity Diagram* Mencetak Laporan Keuangan



Gambar 4.22 Activity Diagram Mencetak Laporan Keuangan

Sumber: (Penulis, 2019)

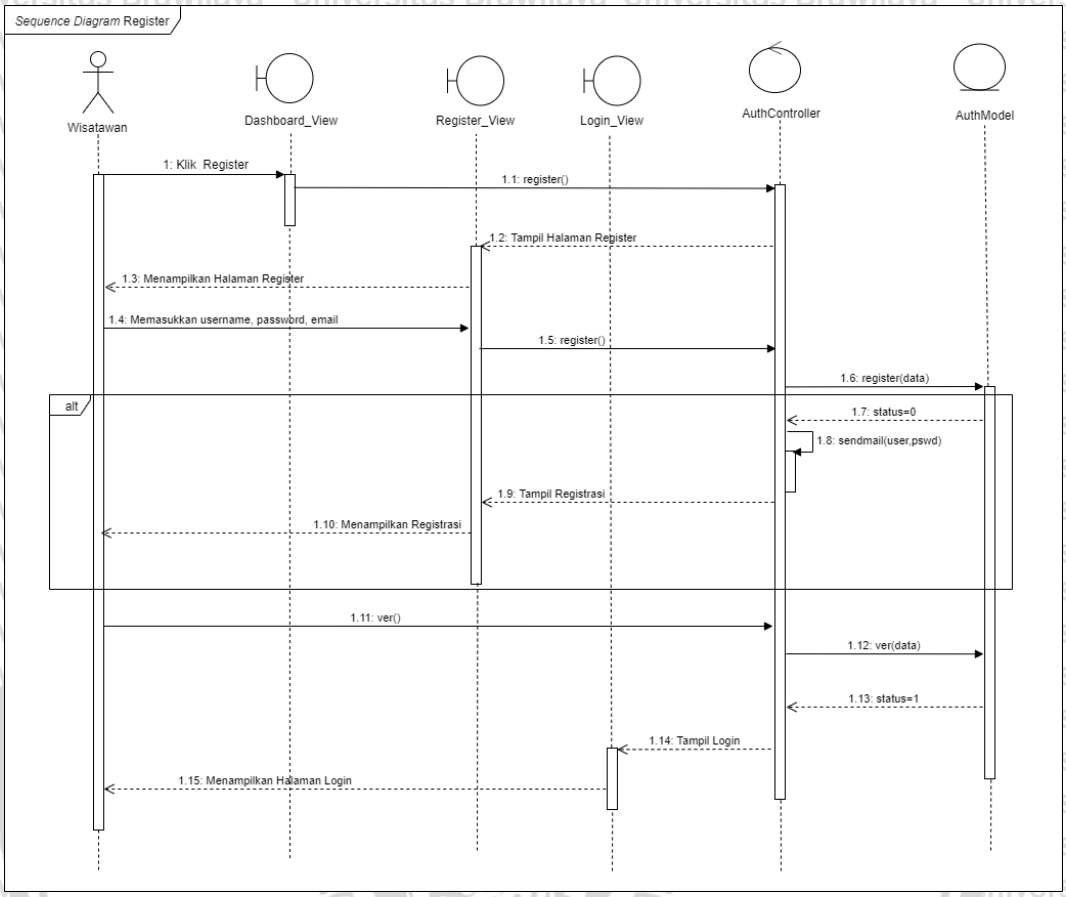
Pada gambar Gambar 4.22 menunjukkan *activity diagram* untuk mencetak laporan keuangan yang dilakukan oleh admin. Aliran kerja atau aktivitas yang terjadi dimulai dari admin memilih menu cetak laporan keuangan. Kemudian sistem akan menampilkan halaman laporan keuangan. Admin memilih menekan tombol print pada menu cetak laporan. Proses selanjutnya sistem akan mencetak laporan keuangan. Laporan akan terunduh oleh admin.

4.6 *Sequence Diagram*

Sequence diagram menggambarkan interaksi satu sama lain antara aktor, *boundary object, control object,* dan *entity object* dengan mengirim dan menerima pesan dalam urutan waktu. *Sequence diagram* dibuat berdasarkan pemodelan *use case* yang telah dibuat pada tahap sebelumnya.

Berikut ini adalah pemodelan *sequence diagram* aplikasi *ecommerce* Wisata Kampung Sapi *Adventure*.

4.6.1 *Sequence Diagram* Register

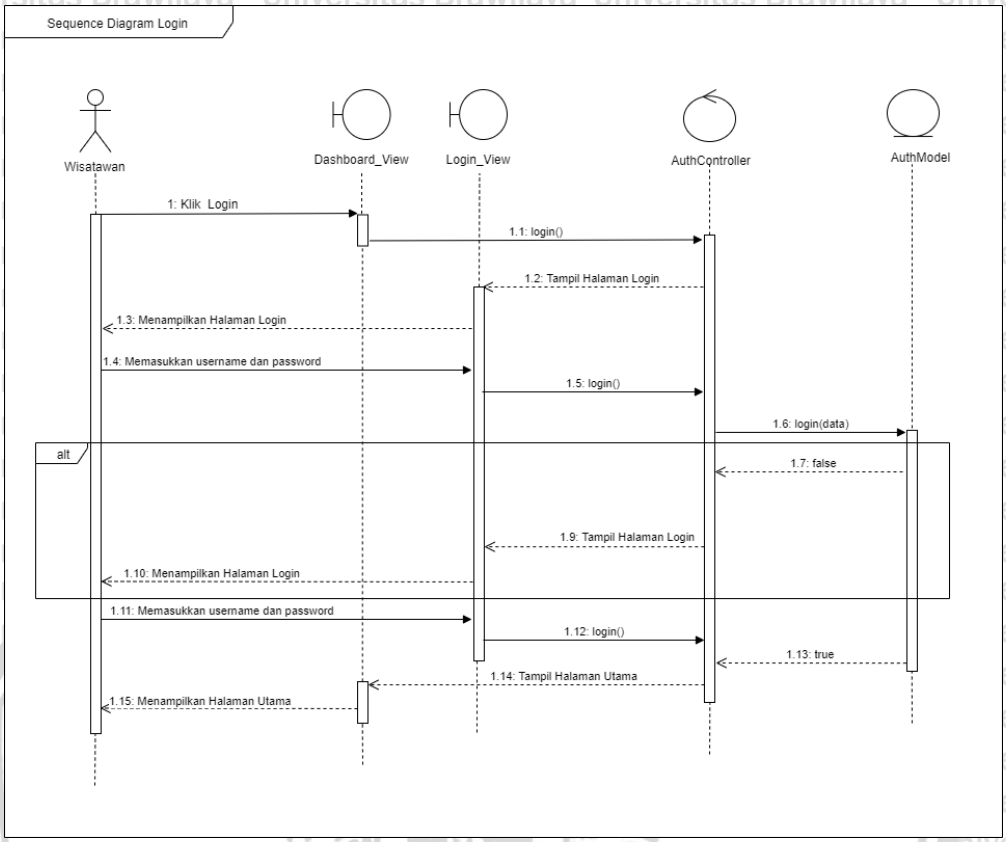


Gambar 4.23 Sequence Diagram Register

Sumber: (Penulis, 2019)

Sequence diagram pada Gambar 4.23 menjelaskan mekanisme untuk register yang dilakukan oleh aktor wisatawan. Wisatawan yang telah berada pada halaman utama kemudian memilih menu register. *AuthController* akan mengarahkan pada *view* halaman register pada wisatawan yang nantinya wisatawan dapat melakukan register dengan menginputkan *username*, *password*, dan email. Kemudian *AuthController* mengarahkan pada *AuthModel* untuk menyimpan inputan tersebut. Pada register terdapat *alternative flow* yang menjelaskan ketika telah melakukan register namun wisatawan belum melakukan verifikasi melalui email maka Auth*Controller* menampilkan halaman register. Jika wisatawan telah melakukan verifikasi melalui email maka Auth*Controller* akan menampilkan halaman *Login*.

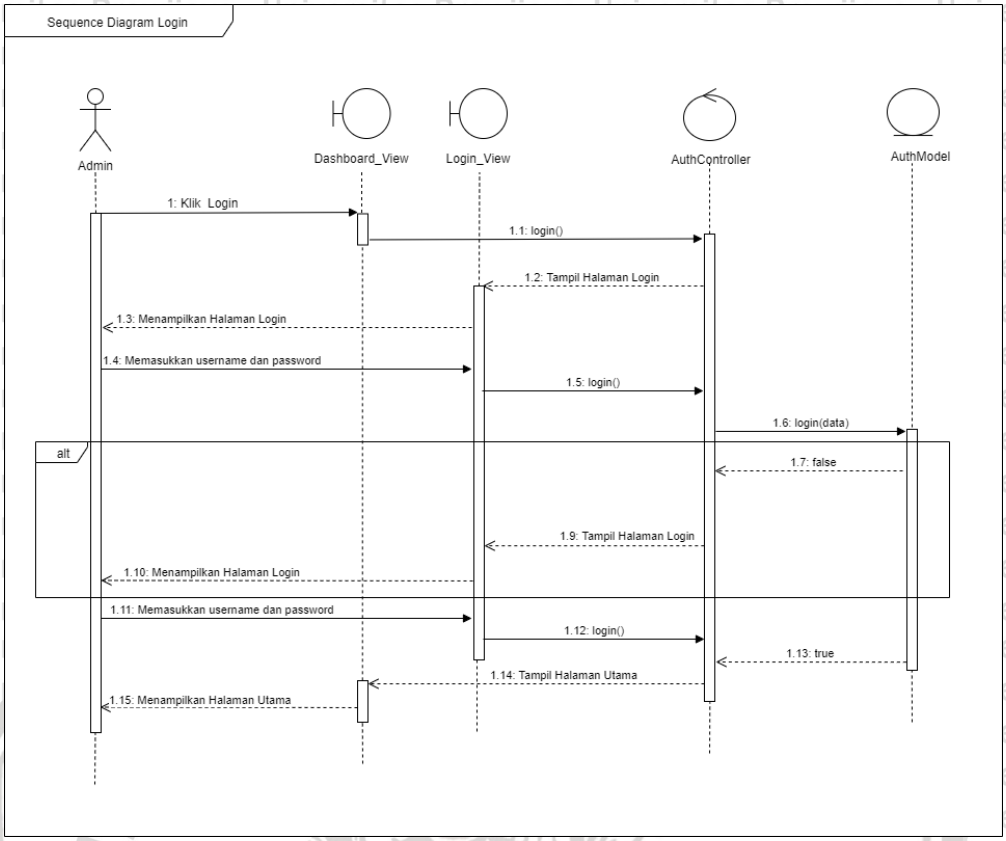
4.6.2 *Sequence Diagram Login*



Gambar 4.24 *Sequence Diagram Login* Pada Wisatawan

Sumber: (Penulis, 2019)

Sequence diagram pada Gambar 4.24 menjelaskan mekanisme *login* yang dilakukan wisatawan. *Sequence diagram* ini dimulai ketika aktor wisatawan berada pada halaman utama lalu memilih menu *login*. *AuthController* akan menampilkan *view login* kemudian wisatawan dapat menginputkan *username* dan *password. AuthController* akan mengarahkan pada *Authmodel* untuk melakukan pengecekan *username* dan *password*. Terdapat *alternative flow* jika *username* dan *password* yang telah diinputkan wisatawan mengalami kesalahan maka akan diarahkan pada halaman *login* kembali.

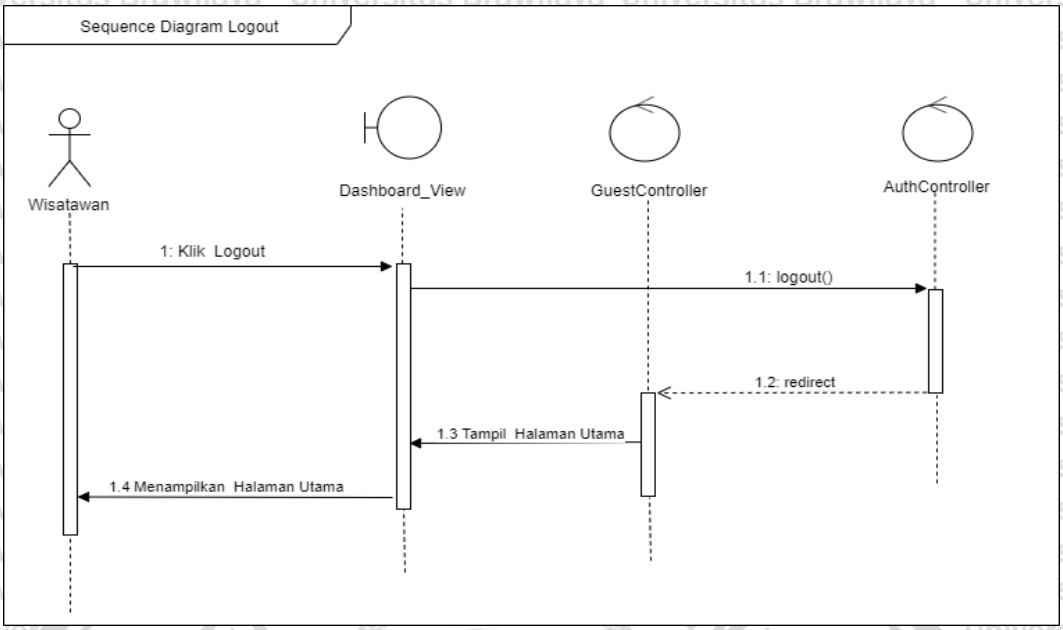


Gambar 4.25 *Sequence Diagram Login* Pada Admin

Sumber: (Penulis, 2019)

Sequence diagram pada Gambar 4.25 menjelaskan mekanisme *login* yang dilakukan wisatawan. *Sequence diagram* ini dimulai ketika aktor wisatawan berada pada halaman utama lalu memilih menu login. *AuthController* akan menampilkan view login kemudian wisatawan dapat menginputkan *username* dan *password. AuthController* akan mengarahkan pada *Authmodel* untuk melakukan pengecekan *username* dan *password*. Terdapat *alternative flow* jika *username* dan *password* yang telah diinputkan wisatawan mengalami kesalahan maka akan diarahkan pada halaman *login* kembali.

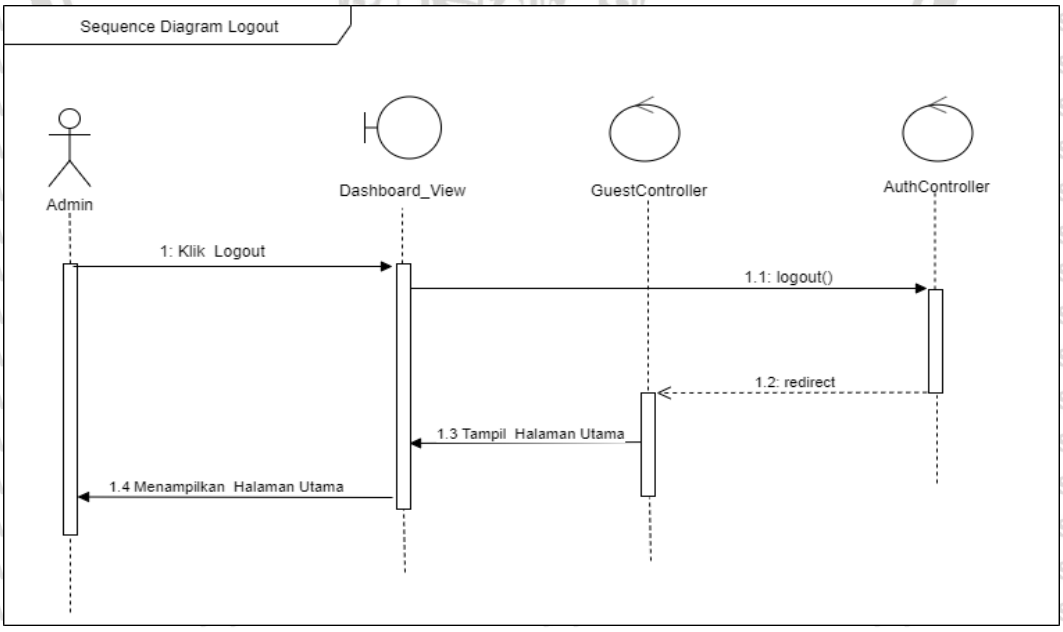
4.6.3 *Sequence Diagram Logout*



Gambar 4.26 *Sequence Diagram Logout* Pada Wisatawan

Sumber: (Penulis, 2019)

Sequence diagram pada Gambar 4.26 menjelaskan mekanisme *logout* yang dilakukan oleh wisatawan. *Sequence diagram* ini dimulai ketika wisatawan telah berada pada halaman *Dashboard*, kemudian wisatawan dapat melakukan *logout*. *GuestController* akan mengarah ke *AuthController* untuk menghapus *session*. Kemudian *AuthController* akan mengarahkan ke tampilan halaman utama sistem.

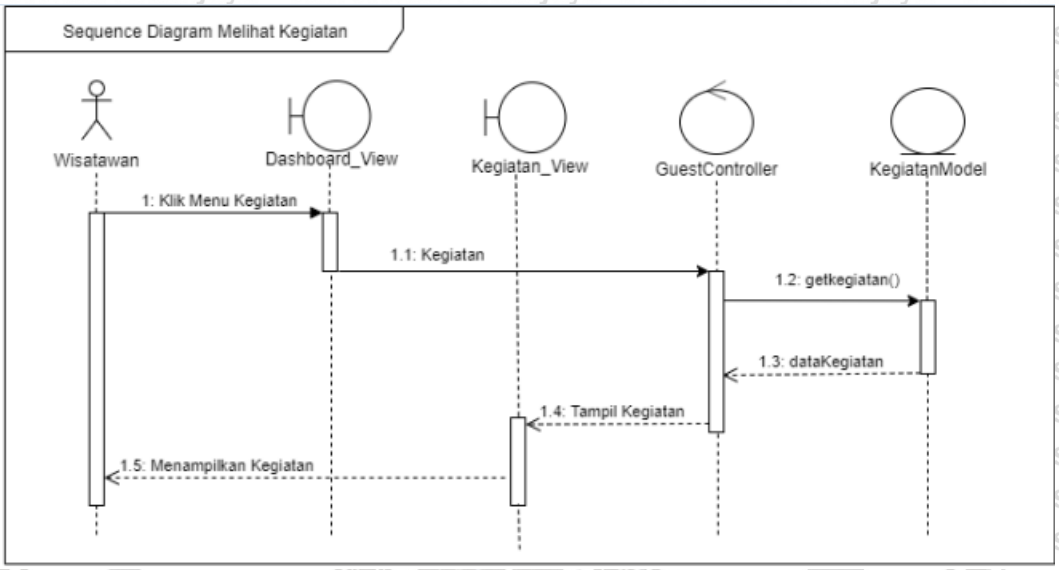


Gambar 4.27 *Sequence Diagram Logout* Pada Admin

Sumber: (Penulis, 2019)

Sequence diagram pada Gambar 4.27 menjelaskan mekanisme *logout* yang dilakukan oleh admin. *Sequence diagram* ini dimulai ketika admin telah berada pada halaman *Dashboard*, kemudian admin dapat melakukan *logout*. *GuestController* akan mengarah ke *AuthController* untuk menghapus *session*. Kemudian *AuthController* akan mengarahkan ke tampilan halaman utama sistem.

4.6.4 *Sequence Diagram* Melihat Kegiatan



Gambar 4.28 *Sequence Diagram* Melihat Kegiatan

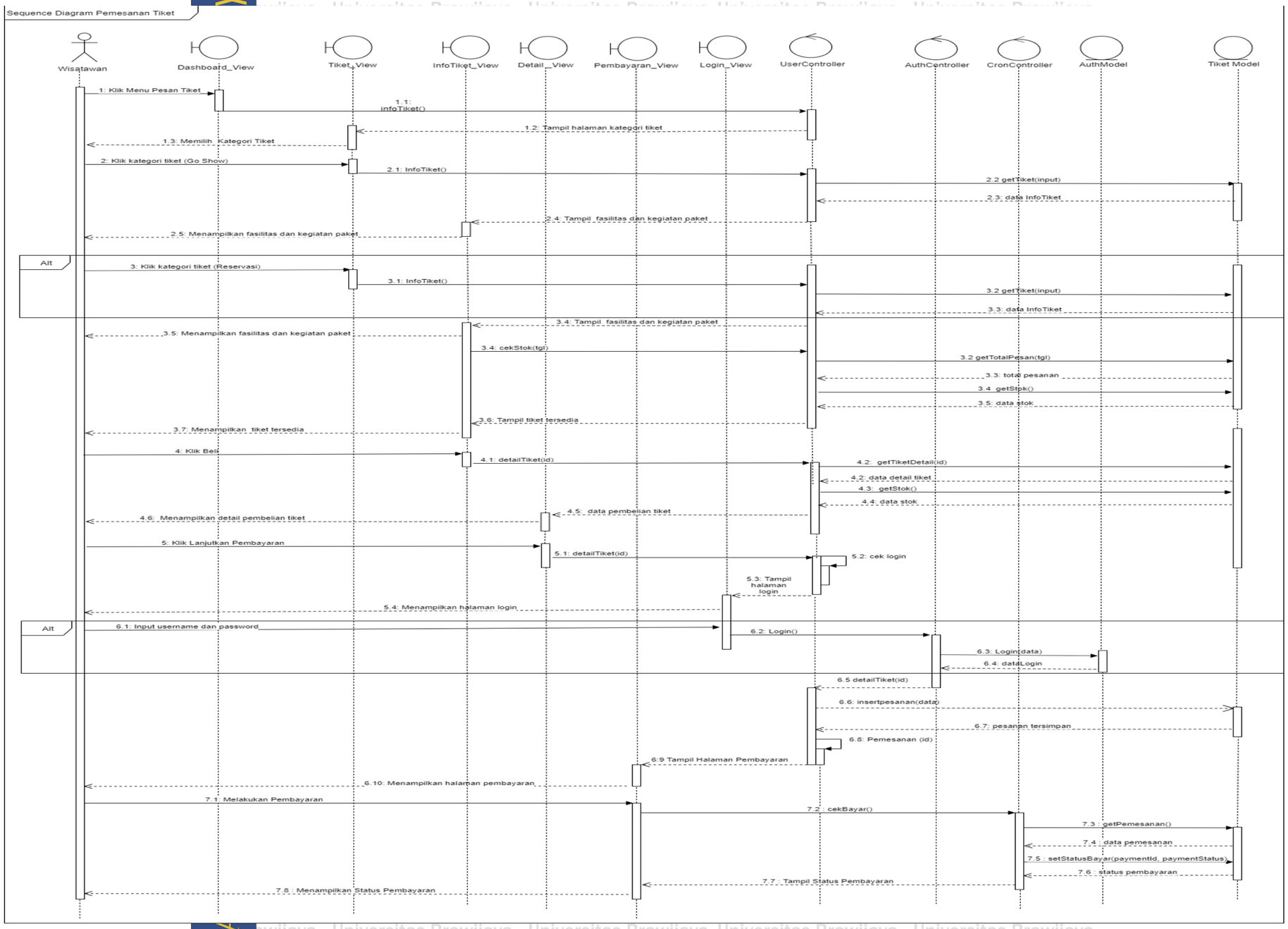
Sumber: (Penulis, 2019)

Sequence diagram pada Gambar 4.28 menjelaskan mekanisme untuk melihat kegiatan yang dilakukan oleh aktor wisatawan. Pada menu utama wisatawan memilih menu kegiatan. Kemudian *GuestController* memanggil fungsi *getkegiatan()* dari *KegiatanModel*. Setelah *GuestControler* meengarahkan ke *view* Kegiatan untuk menampilkan halaman Kegiatan.

4.6.5 *Sequence Diagram* Melakukan Pemesanan Tiket

Sequence diagram pada Gambar 4.29 menjelaskan mekanisme untuk melakukan pemesanan tiket yang dilakukan oleh aktor wisatawan. Wisatawan yang telah berada pada halaman utama kemudian memilih menu pesan tiket. *UserController* akan mengarahkan ke *view* Tiket untuk menampilkan halaman kategori tiket. Setelah itu wisatawan dapat memilih kategori tiket go show. Wisatawan dapat mengetahui fasilitas dan kegiatan paket tiket sesuai dengan kategori tiket go show. Mekanisme *sequence diagram* yang menggambarkan kondisi tersebut, yaitu *UserController* memanggil *gettiket()* dari *TiketModel*. Lalu *UserController* mengarahkan ke *view infoTiket* untuk menampilkan fasilitas dan kegiatan paket. Terdapat *alternative flow* jika wisatawan memilih kategori tiket reservasi, maka untuk mengetahui ketersediaan tiket wisatawan harus melakukan pengecekan tiket. *Sequence diagram* pada mekanisme tersebut dimulai dari *UserController* dengan fungsi *ceksStok(tgl)* kemudian memanggil

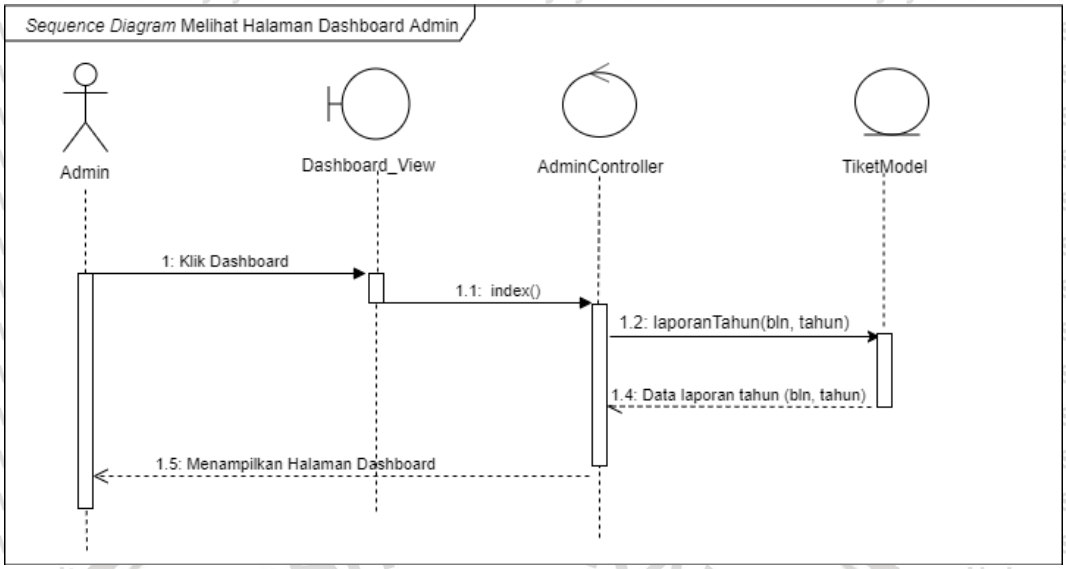
getTotalPesanan(tgl) dari *TiketModel* dan *getSok()*. Lalu *UserController* mengarahkan ke *view infoTiket* untuk menampilkan ketersedian tiket. *Sequence diagram* pada mekanisme pembelian tiket dimulai ketika wisatawan berada pada *view infoTiket* dan menekan tombol beli. *UserController* akan memanggil fungsi *detailTiket(id).* Kemudian menampilkan detail pembelian tiket dalam halaman *DetailTiket. Sequence diagram* pada mekanisme pembayaran tiket dimulai ketika wisatawan menekan tombol lanjutkan pembayaran. Kemudian *UserController* akan melakukan pengecekan apakah status wisatawan yang akan melanjutkan pembayaran sudah *login* atau belum. Jika sudah login maka *UserController* akan menampilkan halaman pembayaran. Terdapat *alternative flow* jika wisatawan yang ingin melanjutkan pembayaran namun belum melakukan login, maka *UserController* akan mengarahkan pada halaman *login*. Kemudian wisatawan akan melakukan pembayaran yang nantinya akan dilakukan pengecekan status pembayaran oleh *CornController* pada fungsi cekBayar dengan memanggil fungsi *getPemesanan()* dan *setStatusBayar(paymentId,paymentStatus)* dari Tiket Model.



Gambar 4.29 *Sequence Diagram* Melakukan Pemesanan Tiket

Sumber: (Penulis, 2019)

4.6.6 *Sequence Diagram* Melihat Halaman *Dashboard* Admin



Gambar 4.30 Sequence Diagram Melihat Halaman *Dashboard* Admin

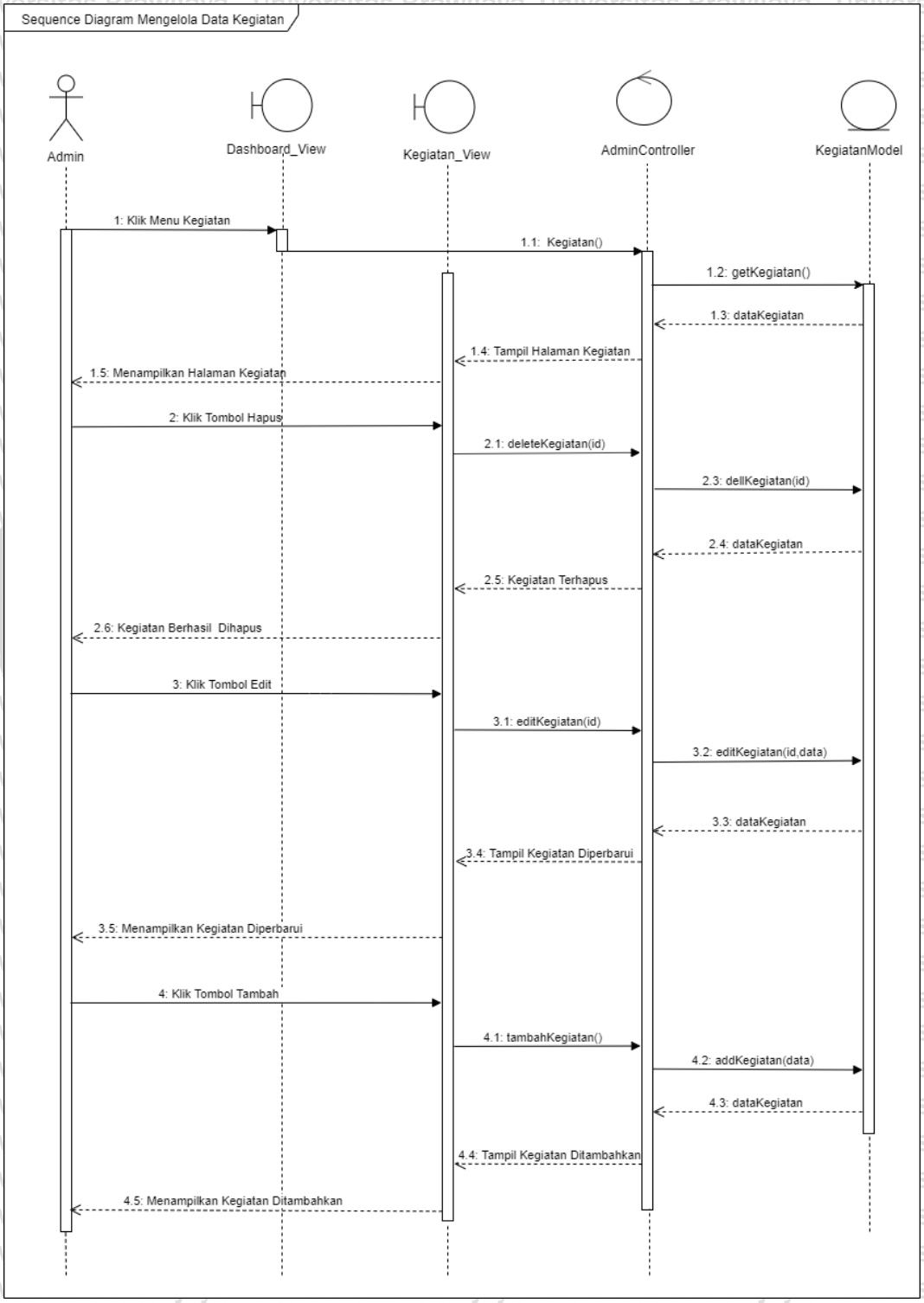
Sumber: (Penulis, 2019)

Sequence diagram pada Gambar 4.30 menjelaskan mekanisme bagi admin untuk melihat halaman *dashboard* admin. *Sequence diagram* dimulai ketika admin memilih menu *dashboard*, kemudian *AdminController* dengan fungsi *index()* akan memanggil fungsi *laporanTahun(bln,tahun)* dari *TiketModel*. Setelah itu *AdminController* akan mengarahkan ke *view dashboard*untuk menampilkan halaman *dashboard* admin.

4.6.7 *Sequence Diagram* Mengelola Data Kegiatan

Sequence diagram pada Gambar 4.31 menjelaskan mekanisme bagi admin untuk melakukan *create, read, update,* dan *delete* pada halaman menu kegiatan. *Sequence diagram* dimulai ketika admin berada pada halaman utama admin dan memilih menu kegiatan. Maka *AdminController* dengan fungsi *kegiatan()* dengan memanggil fungsi *getKegiatan()* dari *KegiatanModel*. Setelah itu, *AdminController* mengarahkan ke *view* Tiket untuk menampilkan halaman kegiatan. Ketika admin menekan tombol hapus kemudian terjadi pemanggilan fungsi *dellKegiatan(id)*, setelah itu *AdminController* memanggil rincian lengkap mengenai kegiatan yang akan dihapus dengan fungsi *dellKegiatan(id)* berdasarkan *id* yang dimiliki. Selanjutnya, halaman dikembalikan ke halaman kegiatan. Ketika admin menekan tombol tambah akan menjalankan fungsi *addKegiatan()* dan mengarahkan admin ke *view* Kegiatan. Setelah itu admin diharuskan untuk mengisi form kegiatan. Kemudian sistem melakukan proses terhadap penambahan kegiatan dengan method *addKegiatan(data)*. Data yang didapatkan dari penambahan kegiatan kemudian dilakukan penyimpanan pada fungsi *addKegiatan(data)* dari *KegiatanModel*. Ketika *admin* ingin mengubah isi kegiatan maka admin menekan tombol edit. Kemudian *AdminController* akan menampilkan rincian kegiatan yang dipilih melalui *editKegiatan(id).* Setelah itu

admin mengubah isi kegiatan pada form kegiatan . Lalu dijalankan fungsi *editKegiatan(id)* untuk mengubah data kegiatan yang berdasarkan *id* dan dikembalikan ke halaman awal kegiatan.

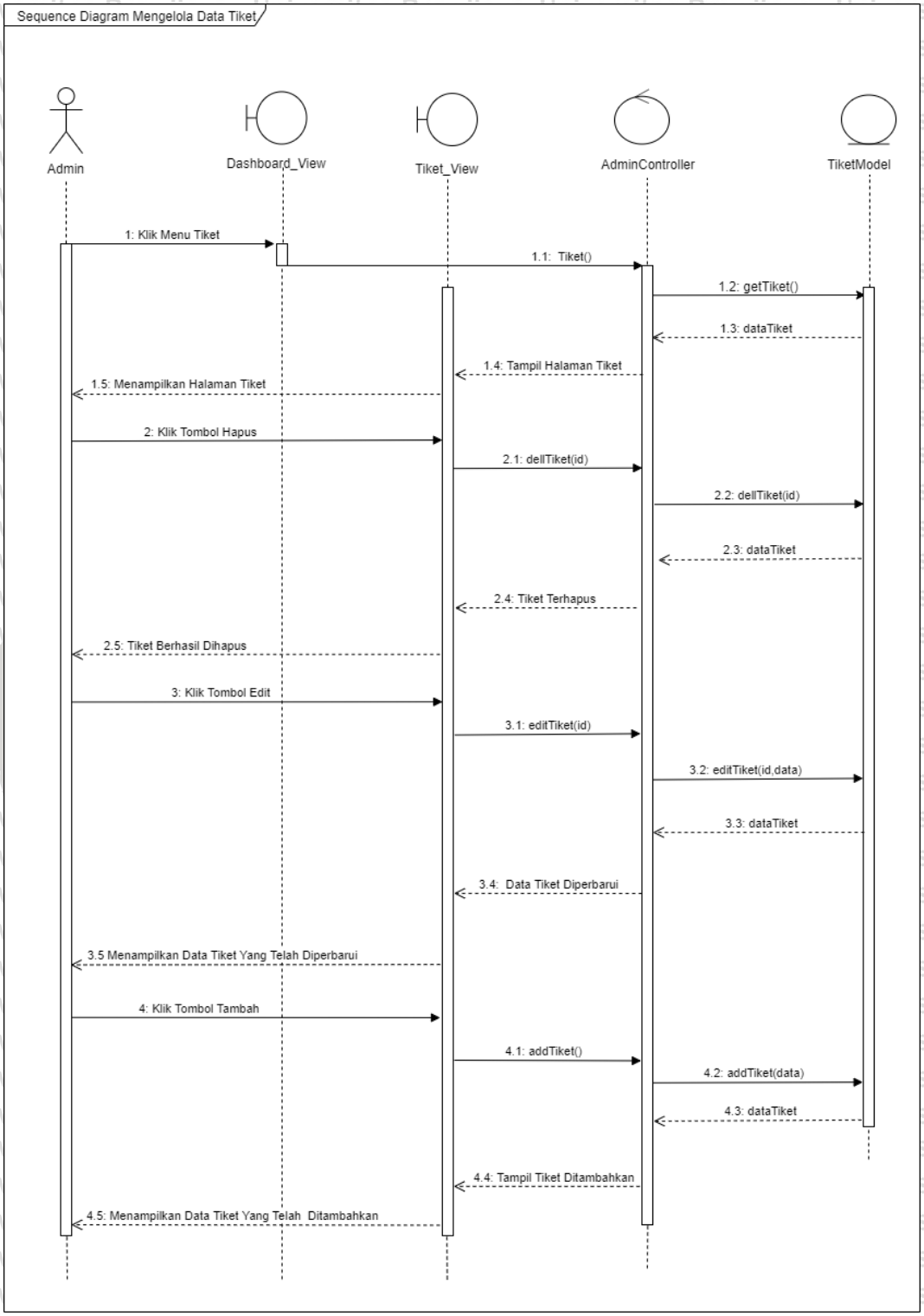


Gambar 4.31 *Sequence Diagram* Mengolah Data Kegiatan

Sumber: (Penulis, 2019)

4.6.8 *Sequence Diagram* Mengelola Data Tiket

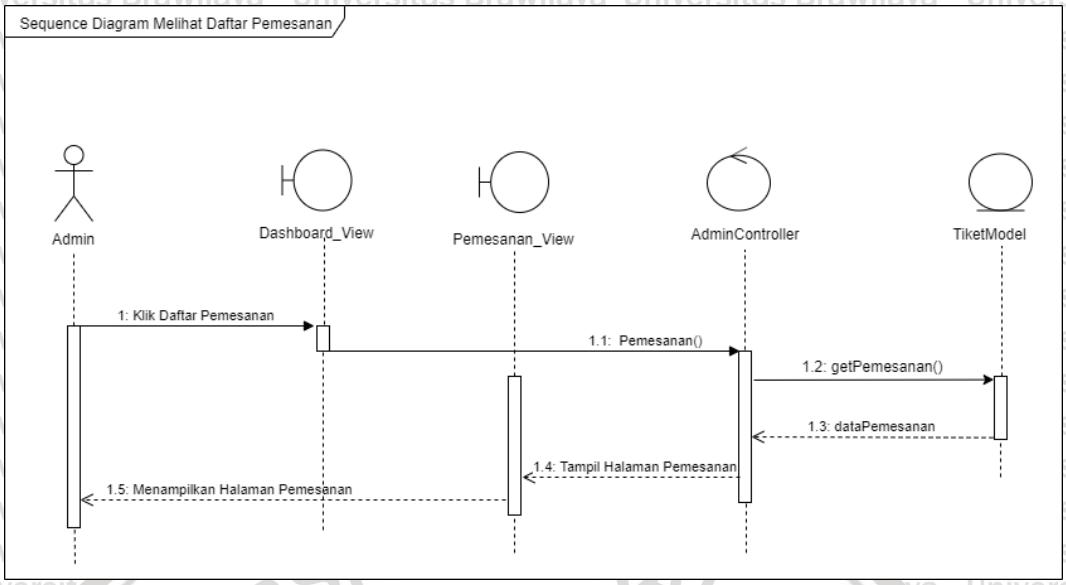
Sequence diagram pada Gambar 4.32 menjelaskan mekanisme bagi admin untuk melakukan *create, read, update,* dan *delete* pada halaman menu Tiket. *Sequence diagram* dimulai ketika admin berada pada halaman utama dan memilih menu Tiket. Maka *AdminController* dengan fungsi *Tiket()* akan memanggil fungsi *getTiket()* dari *KegiatanModel*. Setelah itu, *AdminController* akan mengarahkan ke *view* Tiket untuk menampilkan halaman Tiket. Ketika admin klik tombol hapus kemudian terjadi pemanggilan fungsi *dellTiket(id)* pada *AdminController*, setelah itu *AdminController* memanggil rincian lengkap mengenai kegiatan yang akan dihapus dengan fungsi *dellTiket(id)* pada *TiketModel* berdasarkan id yang dimiliki. Selanjutnya, halaman dikembalikan ke *view* Tiket. Ketika admin klik tombol tambah akan menjalankan fungsi *addTiket()* dan mengarahkan admin ke *view* Tiket. Setelah itu admin diharuskan untuk mengisi form Tiket. Kemudian sistem melakukan proses terhadap penambahan Tiket dengan method *addTiket(data)* pada *AdminController*. Data yang didapatkan dari penambahan kegiatan kemudian dilakukan penyimpanan pada fungsi *addTiket(data)* dari T*iketMode*l. Ketika admin ingin mengubah isi kegiatan maka admin menekan tombol edit. Kemudian *AdminControlle*r akan menampilkan rincian kegiatan yang dipilih melalui *editTiket(id)*. Setelah itu admin mengubah isi tiket pada form tiket. Lalu dijalankan fungsi *editTiket(id)* untuk mengubah data tiket berdasarkan *id* dan dikembalikan ke halaman Tiket.



Gambar 4.32 *Sequence Diagram* Mengelola Data Tiket

Sumber: (Penulis, 2019)

4.6.9 *Sequence Diagram* Melihat Daftar Pemesanan Tiket



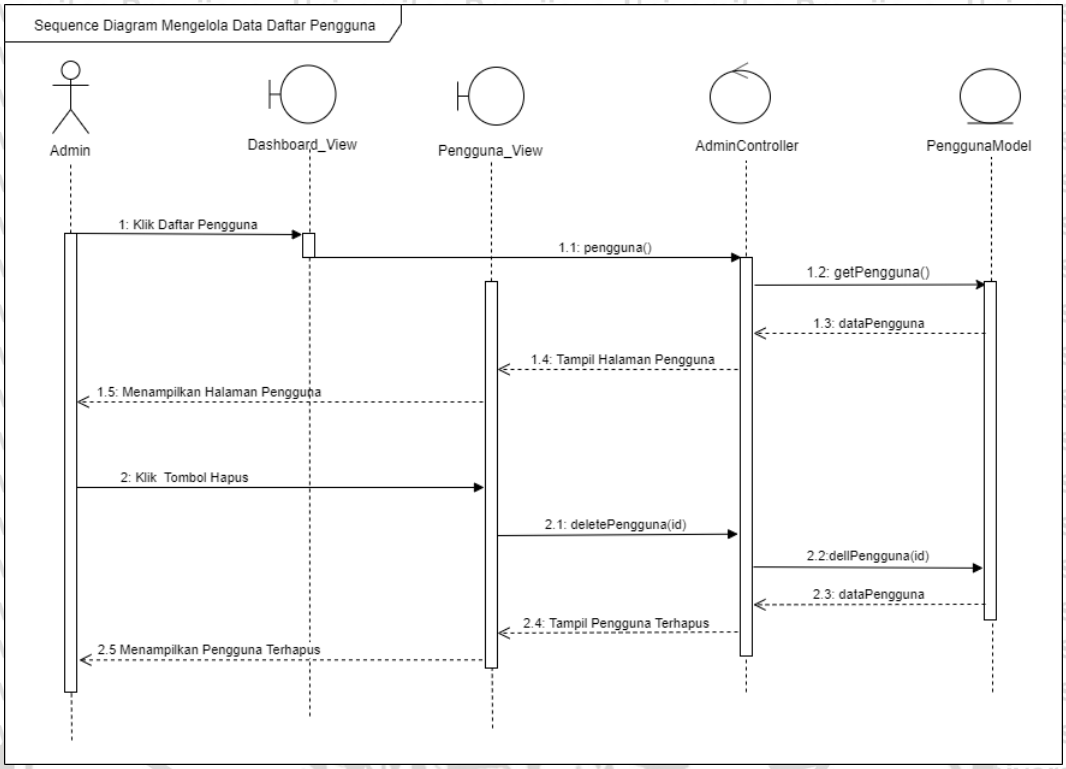
Gambar 4.33 *Sequence Diagram* Melihat Daftar Pemesanan

Sumber: (Penulis, 2019)

Sequence diagram pada Gambar 4.33 ini menjelaskan mekanisme admin dalam melihat daftar pemesanan tiket. *Sequence diagram* melihat daftar pemesanan tiket dimulai dari admin memilih menu daftar pemesanan pada halaman utama admin, kemudian *AdminController* dengan fungsi *Pemesanan()* akan memanggil *getPemesanan()* dari *TiketModel*. Setelah itu AdminController akan mengarahkan ke view Pemesanan untuk menampilkan halaman Pemesanan Tiket.

4.6.10 *Sequence Diagram* Mengelola Data Daftar Pengguna

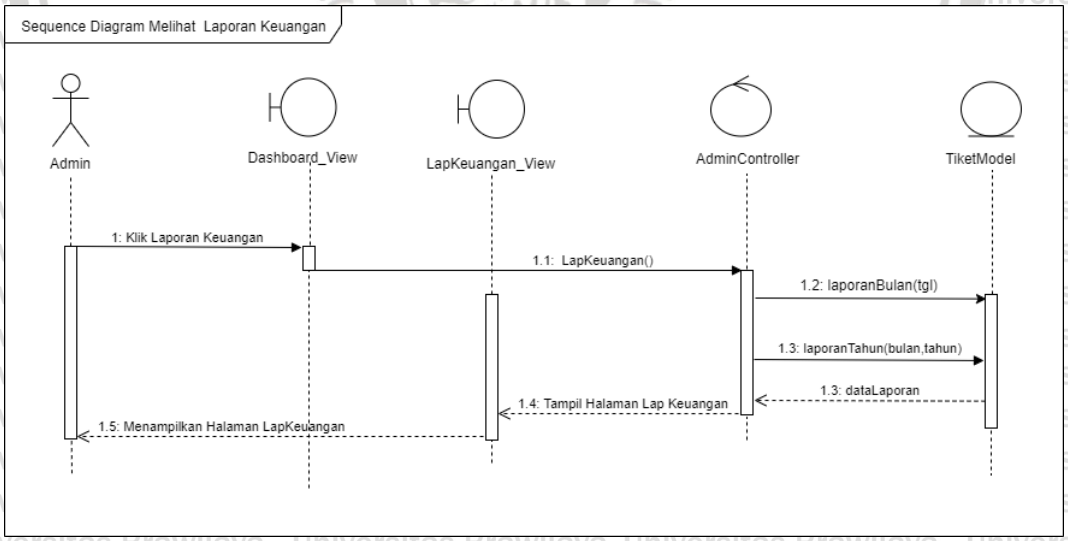
Sequence diagram pada Gambar 4.34 menjelaskan mekanisme admin dalam mengelola data daftar *pengguna*. *Sequence diagram* mengelola data daftar pengguna dimulai dari admin memilih menu daftar pengguna, kemudian *AdminController* dengan fungsi *pengguna()* akan memanggil fungsi *getPengguna()* dari *PenggunaModel*. Setelah itu *AdminController* akan mengarahkan ke *view* Pengguna untuk menampilkan halaman pengguna. Ketika admin menekan tombol hapus pada *view Pengguna* kemudian terjadi pemanggilan fungsi dellPengguna(id) dari *PenggunaModel,* setelah itu *AdminController* memanggil rincian lengkap mengenai pengguna yang akan dihapus dengan fungsi *dellTiket(id)* berdasarkan id yang dimiliki. Selanjutnya, halaman dikembalikan ke *view* Pengguna.



Gambar 4.34 *Sequence Diagram* Mengubah Data Daftar Pengguna

Sumber: (Penulis, 2019)

4.6.11 *Sequence Diagram* Melihat Laporan Keuangan



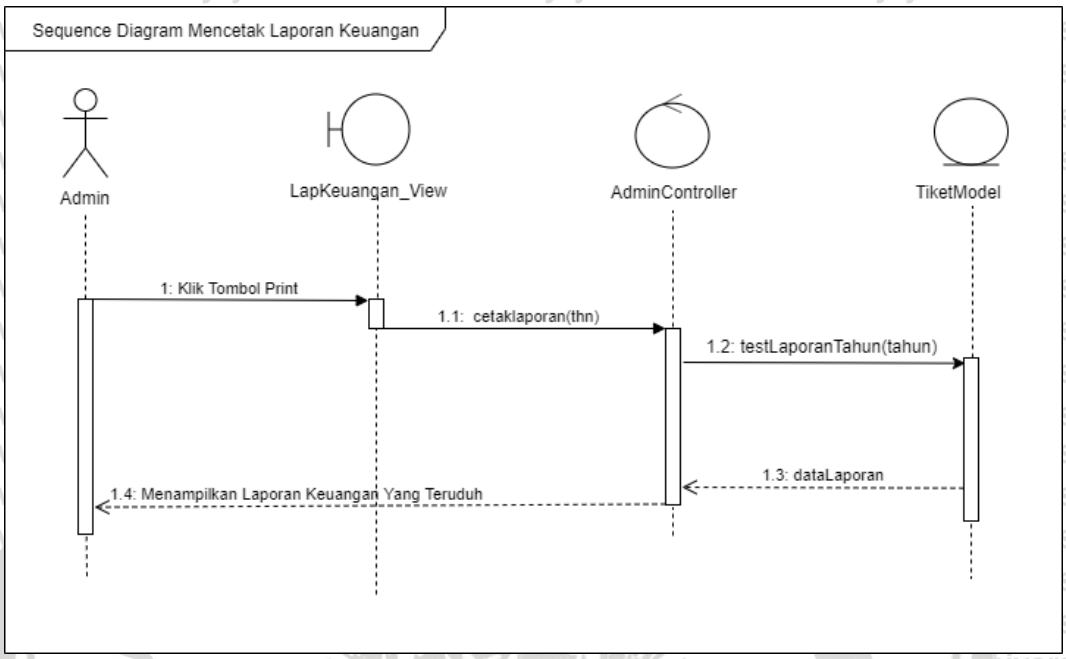
Gambar 4.35 *Sequence Diagram* Melihat Laporan Keuangan

Sumber: (Penulis, 2019)

Sequence diagram pada Gambar 4.35 menjelaskan mekanisme admin dalam melihat laporan keuangan. *Sequence diagram* melihat laporan keuangan tiket dimulai dari admin melakukan klik laporan keuangan pada halaman utama admin kemudian *AdminController* dengan fungsi *LapKeuangan()* akan memanggil

laporanBulan(tgl) dan *laporanTahun(bulan,tahun)* dari TiketModel. Setelah itu *AdminController* akan mengarahkan ke *view* LapKeuangan untuk menampilkan halaman Laporan Keuangan Tiket.

4.6.12 *Sequence Diagram* Mencetak Laporan Keuangan



Gambar 4.36 *Sequence Diagram* Mencetak Laporan Keuangan

Sumber: (Penulis, 2019)

Sequence diagram pada Gambar 4.36 menjelaskan mekanisme admin dalam mencetak laporan keuangan. *Sequence diagram* mencetak laporan keuangan tiket dimulai dari admin menekan tombol print pada *view* LapKeuangan kemudian *AdminController* dengan fungsi cetaklaporan() akan memanggil fungsi *testLaporanTahun(tahun)* dari *TiketModel*. Setelah itu *AdminController* akan mengarahkan ke *view* LapKeuangan untuk menampilkan laporan keuangan yang telah terunduh.

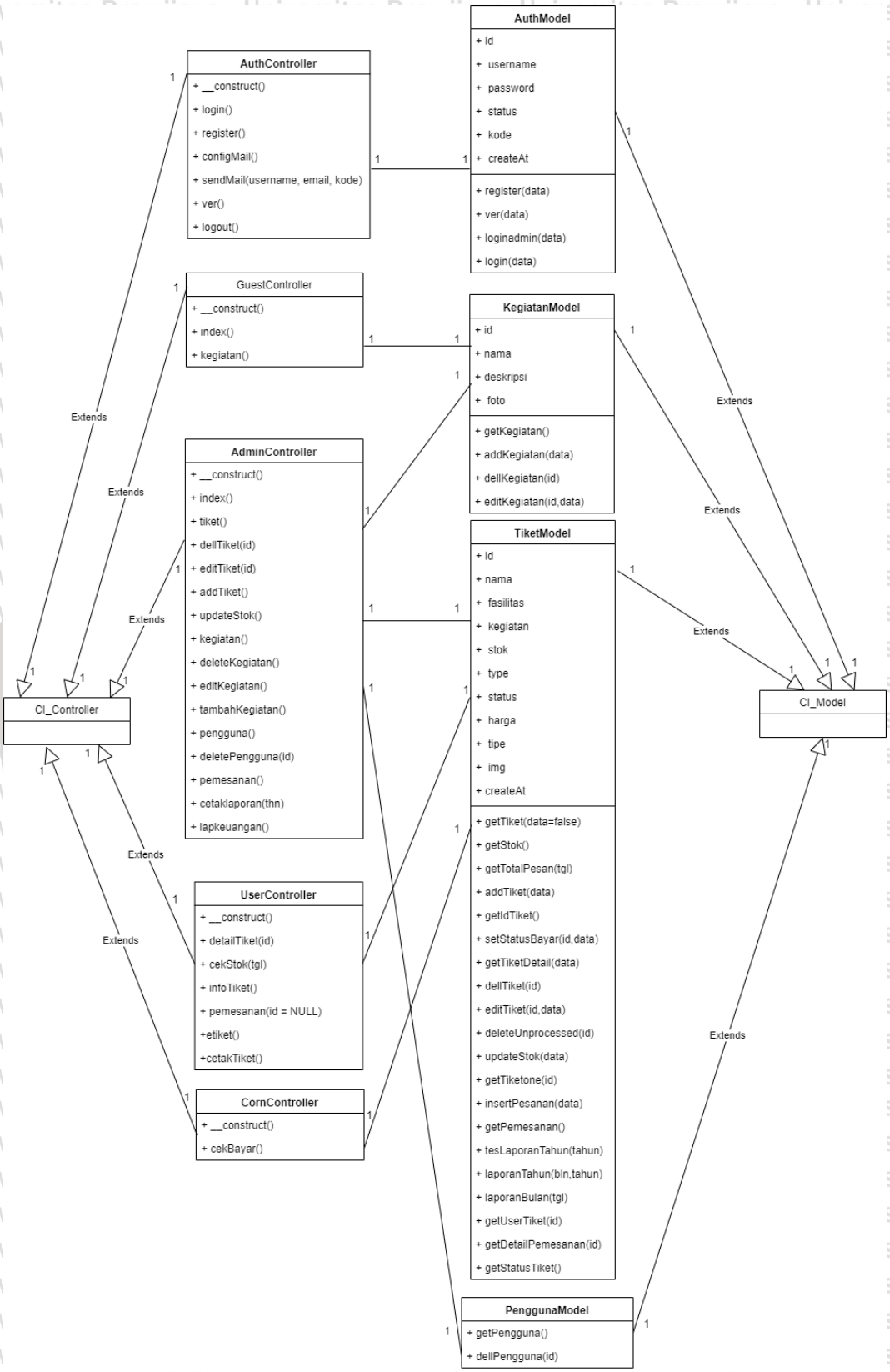
4.7 Class Diagram

Class Diagram merupakan penggambaran struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class Diagram* aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* akan dijelaskan pada Gambar 4.37.

Pada Gambar 4.37 menjelaskan tentang jenis *Class* yang digunakan pada aplikasi *e-commerce* Wisata Kampung Sapi *Adventure*. *Class* model digunakan untuk melakukan penyimpanan maupun pengolahan data. Sedangkan *Class controller* untuk melakukan pengolahan *input* dari *user* ketika mengakses aplikasi *e-commerce* Wisata Kampung Sapi *Adventure*. *Class View* tidak ditampilkan pada *class* diagram. Sedangkan kelas model dengan nama *CI_Model*, serta *Class controller* dengan nama *CI_Controller*. Masing masing *class model* merupakan

extends dari *CI_Model*. Begitu juga masing-masing *class controller* merupakan extends dari *CI_Controller*. Relasi antara *CI_Controller* dengan masing – masing *class controller* adalah *one to one* yaitu, satu *CI_Controller* memiliki satu *class controller* dan satu *class controller* memiliki satu *CI_Controller.* Relasi antara *CI_Model* dengan masing – masing *class model* adalah *one to one* yaitu, satu *CI_Model* memiliki satu *class model* dan satu *class model* memiliki satu *CI_Model.* Hubungan antara *class controller* dengan *class model* yaitu,asosiasi. Relasi antara *class controller* dengan *class model* yaitu, *one to one* yang berarti satu *class controller* memiliki satu *class model* dan satu *class model* memiliki satu *class controller*.

Terdapat lima *class controller* pada aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* yaitu *AuthController*, *UserController, GuestController, AdminController,* dan *CronController*. *AuthController* merupakan *class* yang digunakan untuk melakukan operasi *register, login, dan logout.* Pada *class AuthController* terdapat fungsi yang digunakan untuk mengambil data dari *class AuthModel*. *UserController* merupakan *class* yang digunakan untuk melakukan operasi pemesanan tiket dan pengecekan stok tiket. *GuestController* merupakan *class* yang digunakan untuk menampilkan halaman utama aplikasi *e-commerce* Wisata Kampung Sapi *Adventure*. Pada *class GuestController* terdapat fungsi yang digunakan untuk mengambil data dari *class TiketModel* dan *KegiatanModel. AdminController* merupakan *class* yang digunakan untuk melakukan *update* tiket, *update* kegiatan, *update* daftar pengguna, menampilkan halaman daftar pemesanan, menampilkan halaman laporan keuangan, dan mencetak halaman keuangan. Pada *class AdminController* terdapat fungsi yang digunakan untuk mengambil data dari *class TiketModel, KegiatanModel,* dan *PenggunaModel. CronController* merupakan *controller* yang digunakan untuk melakukan cek status pembayaran dengan memanggil fungsi cekBayar.



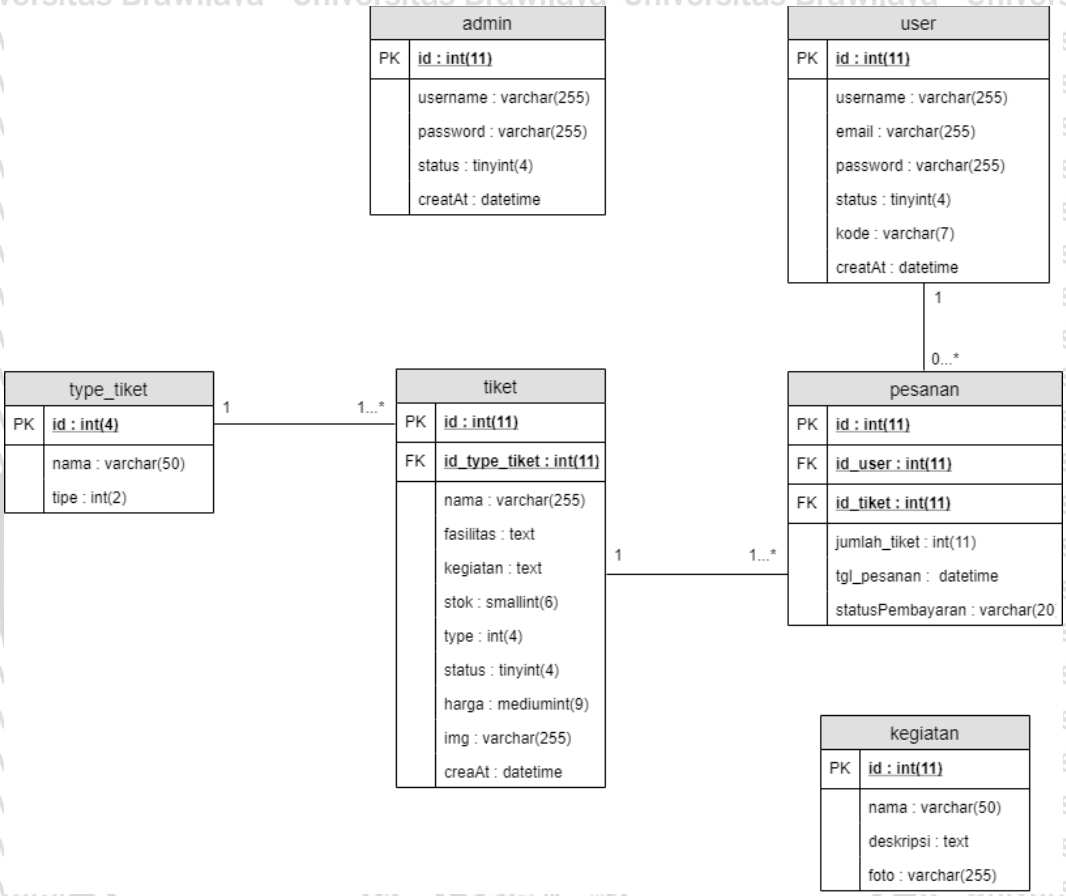
Gambar 4.37 Class Diagram Aplikasi *E-Commerce* Wisata Kampung Sapi *Adventure*

Sumber: (Penulis, 2019)

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA
Universitas Brawijaya

4.8 Physical Data Model

Pada bagian ini dijelaskan implementasi basis data yang digunakan pada rancangan aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* untuk menyimpan dan mengelola segala informasi dan data yang akan ditampilkan ke pengguna. Struktur basis data dapat dilihat pada *Physical Data Model* (PDM) aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* pada Gambar 4.38.



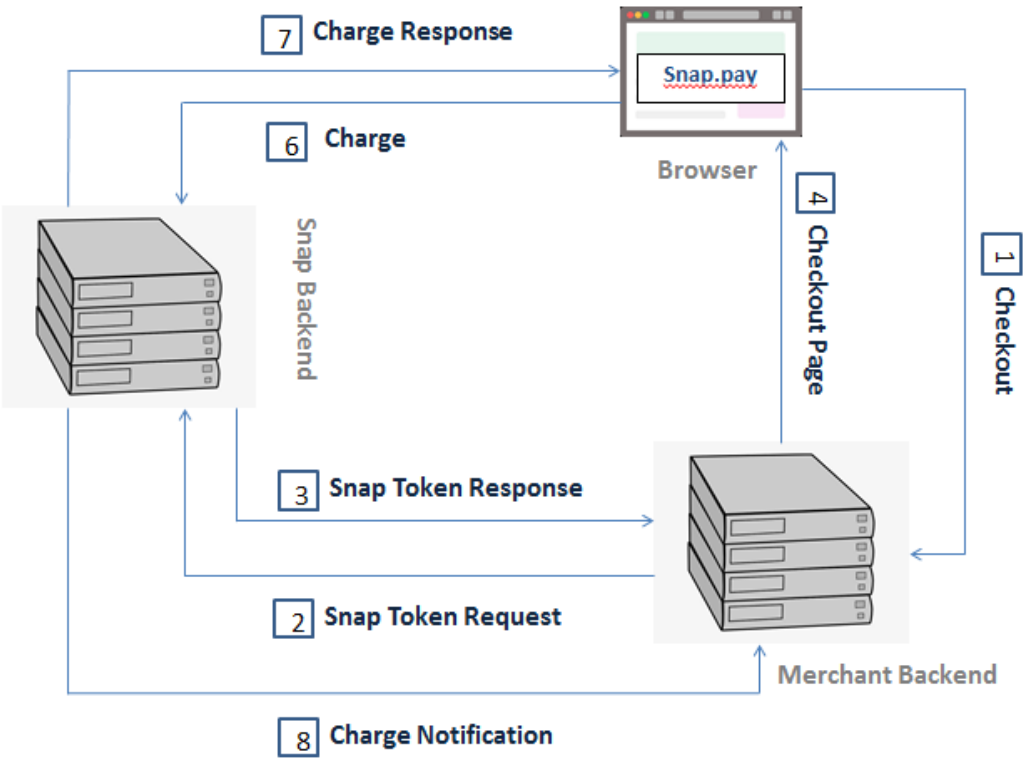
Gambar 4.38 *Physical Data Model* Aplikasi *E-Commerce* Wisata Kampung Sapi *Adventure*

Sumber: (Penulis, 2019)

Pada gambar 4.38 terlihat bahwa aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* menggunakan 6 tabel data untuk menyimpan data. Terlihat juga bagaimana relasi dan entitas tiap tabel data. Relasi antara tabel *type_tiket* dengan tiket yaitu *one to many* yang artinya satu *type* tiket memiliki beberapa paket tiket dan satu paket tiket hanya memiliki satu *type* tiket. Relasi antara tabel tiket dengan tabel pesanan yaitu *one to many* yang artinya satu paket tiket dapat dilakukan beberapa kali pesanan dan satu kali pesanan hanya dapat memesan satu paket tiket. Relasi yang terjadi antara tabel user dengan pesanan yaitu *one to many* yang artinya satu *user* dapat melakukan tidak sama sekali pesanan atau lebih dari pesanan dan satu pesanan dimiliki satu *user*.

4.9 Perancangan Integrasi Layanan API *Sandbox* Pada *Payment Gateway* Midtrans

Pada penelitian untuk membantu dalam proses pengembangan dan pengimplementasian aplikasi *e-commerce* Wisata Kampung Sapi *Adventure*, maka diperlukan perancangan pengintegrasian antara aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* dengan layanan API *sandbox* pada *payment gateway* midtrans. Pada Gambar merupakan perancangan integrasi API *sandbox* pada *payment gateway* midtrans.



Gambar 4.39 Perancangan Integrasi API *Sandbox* Pada *Payment Gateway* Midtrans

Sumber: (Penulis, 2019)

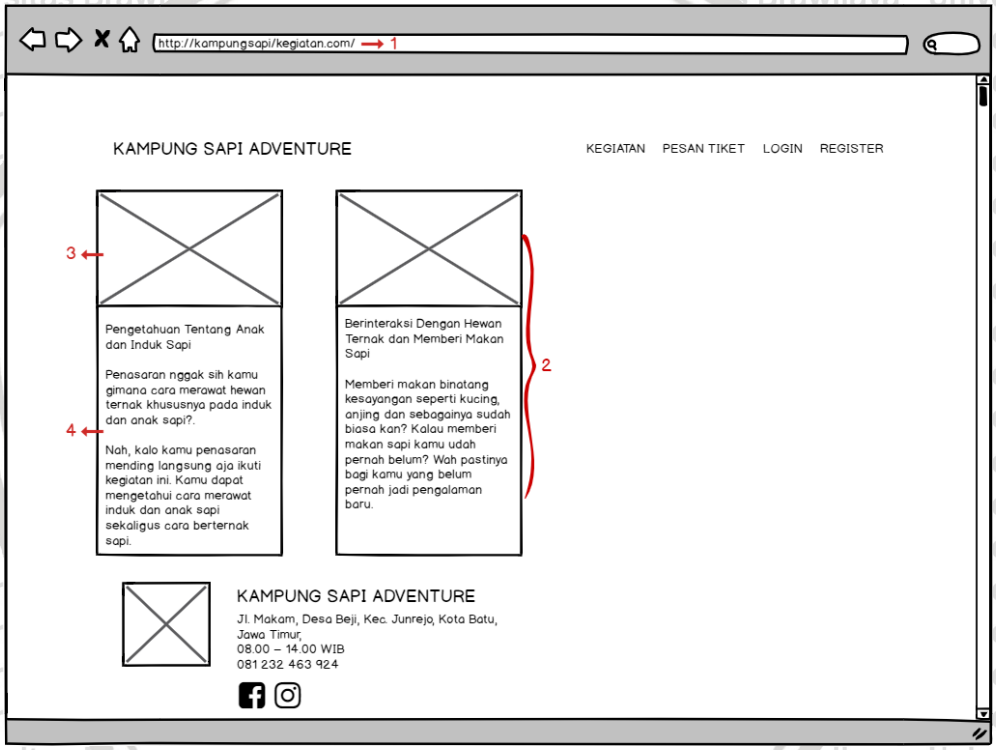
Pada Gambar 4.39 merupakan perancangan integrasi API *sandbox* pada *payment gateway* midtrans. Pada perancangan tersebut ketika pengguna melakukan operasi *checkout,* server pada website pedagang membuat API ke *snap backend* untuk mendapatkan *snap token*. Selanjutnya snap *backend* merespon panggilan API dengan *snap token*. *Snap* merupakan sebuah metode yang berfungsi untuk menampilkan *pop up* portal pembayaran pada website pengguna. Server pada website pedagang membuat halaman html dan mengirimkannya kembali ke browser. Kemudian pengguna memverifikasi detail pembelian dan menekan tombol bayar. Setelah itu, kode javascript pada website pedagang memanggil fungsi *snap.pay* (*snap token*). Pengguna kemudian mengisi rincian pembayaran dan menekan tombol konfirmasi. *Snap javascript* akan

mengirim detail pembayaran ke *snap backend*. *Snap backend* akan memproses detail pembayaran dan merespon dengan status pembayaran. *Snap javascript* akan memanggil kembali fungsi yang terkait yang disediakan oleh kode program pada website pedagang. Selanjutnya *snap backend* akan memberi tahu server merchant mengenai status pemabayaran yang telah dilakukan oleh pengguna.

4.10 Perancangan Antarmuka Pengguna

Perancangan antarmuka ini menjelaskan tentang rancangan antarmuka pengguna sistem dalam berinteraksi dengan penggunanya. Berikut ini adalah perancangan antarmuka dari aplikasi *e-commerce* Wisata Kampung Sapi *Adventure*:

4.10.1 Perancangan Antarmuka Halaman Pada Menu Kegiatan



Gambar 4.40 Rancangan Antarmuka Halaman Pada Menu Kegiatan

Sumber: (Penulis, 2019)

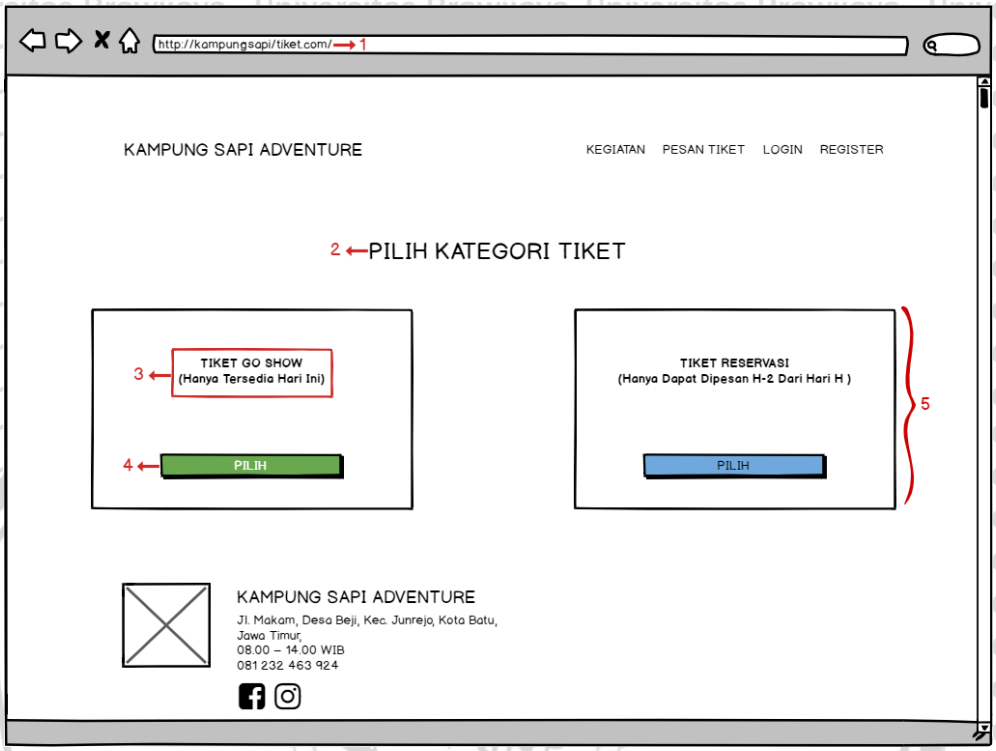
Perancangan antarmuka Pada Gambar 4.40 merupakan tampilan antarmuka halaman pada menu kegiatan. Pada halaman kegiatan wisatawan dapat melihat kegiatan yang ada pada Wisata Kampung Sapi *Adventure.*

Keterangan rancangan antarmuka halaman kegiatan

1. Alamat URL untuk mengakses halaman kegiatan dari menu kegiatan pada aplikasi *e-commerce* Wisata Kampung Sapi *Adventure*
2. *Card* berisi *card image* dan *card text*

3. *Card image* berisi gambar pada Wisata Kampung Sapi *Adventure*
4. *Card text* berisi deskrispi kegiatan pada Wisata Kampung Sapi *Adventure* sesuai isi dari *card image* dan *card text*

4.10.2 Perancangan Antarmuka Halaman Menu Pesan Tiket



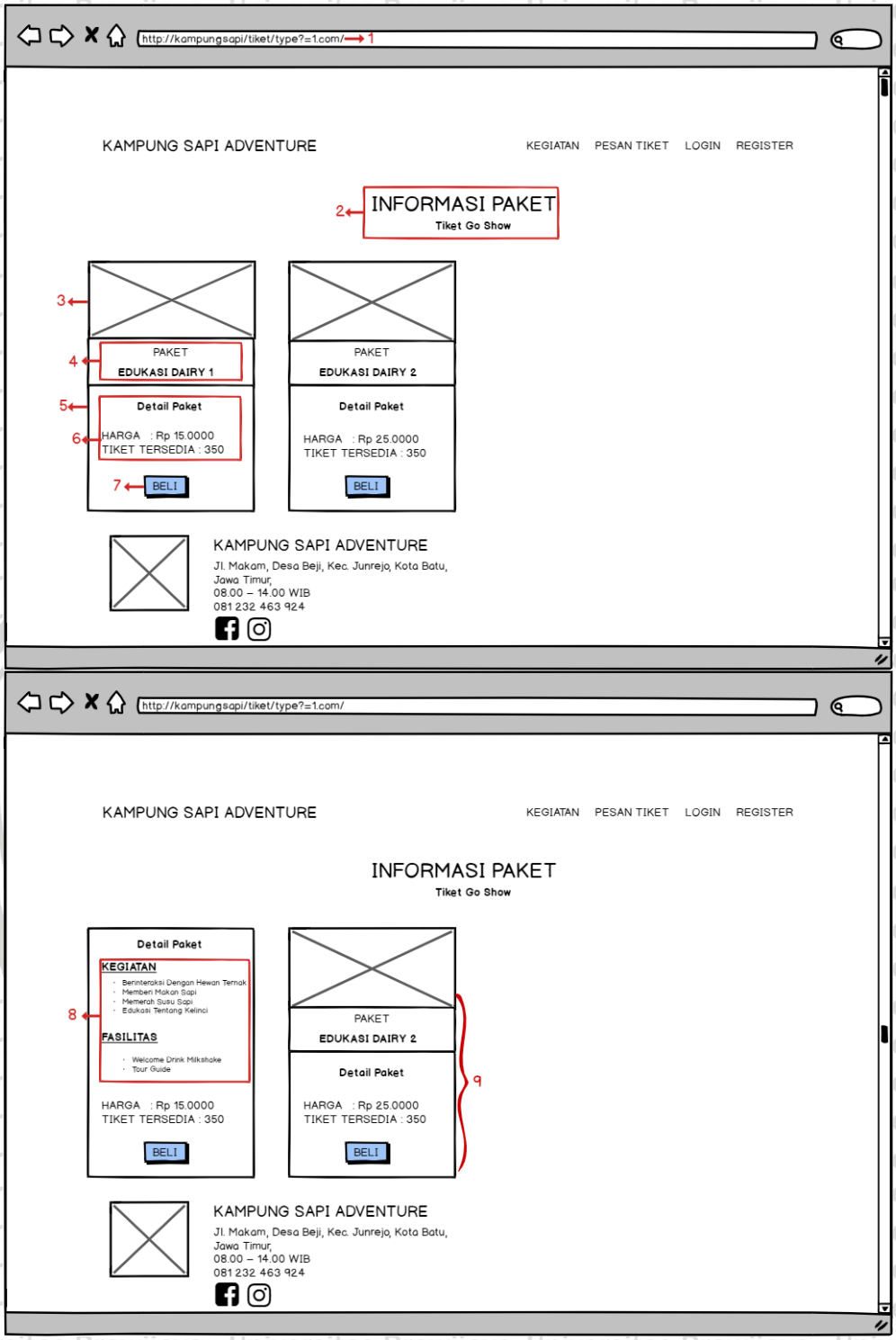
Gambar 4.41 Rancangan Antamuka Halaman Memilih Kategori tiket

Sumber: (Penulis, 2019)

Pada perancangan antarmuka Gambar 4.41 wisatawan dapat memilih kategori tiket.

Keterangan rancangan antarmuka halaman memilih kategori tiket

1. Alamat URL digunakan wisatawan untuk mengakses halaman kategori tiket pada aplikasi *e-commerce* Wisata Kampung Sapi *Adventure*
2. *Text* berisi judul halaman "Pilih Kategori Tiket"
3. *Text* berisi nama kategori tiket
4. Tombol dengan nama "PILIH" digunakan untuk wisatawan memilih kategori dengan menekan tombol "PILIH"
5. *Card* berisi *text* dan tombol



Gambar 4.42 Rancangaan Antarmuka Halaman Informasi Paket Tiket Go Show

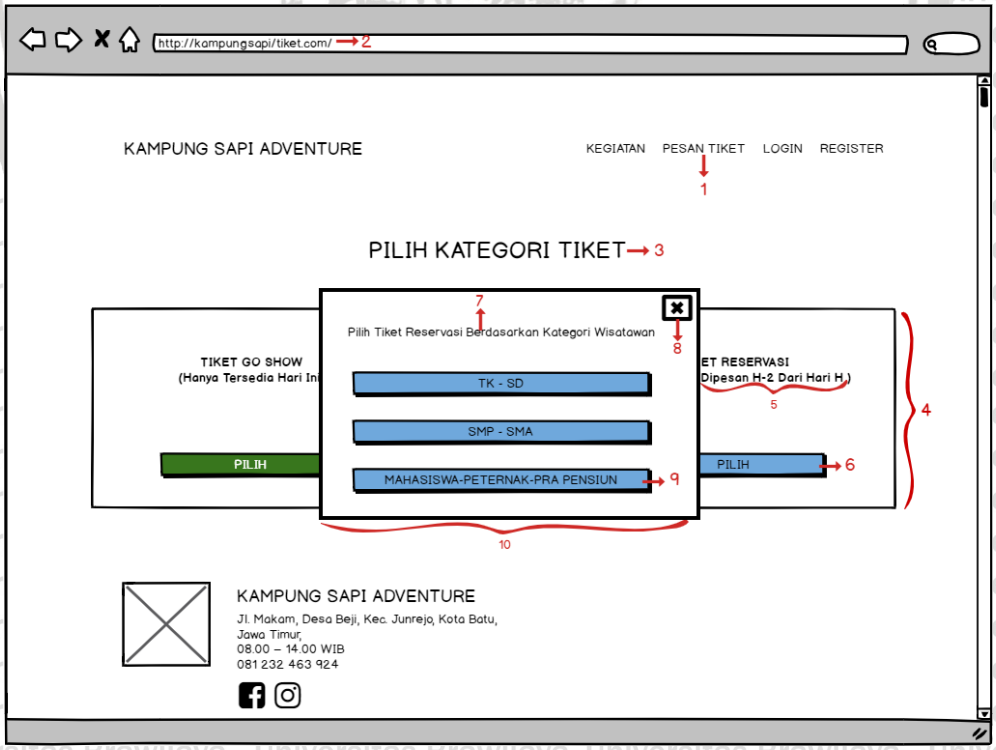
Sumber: (Penulis, 2019)

Perancangan antarmuka pada Gambar 4.42 merupakan tampilan antarmuka halaman informasi paket tiket go show. Halaman ini ditampilkan setelah wisatawan memilih kategori tiket go show. Pada halaman ini wisatawan

dapat mengetahui paket tiket apa saja yang terdapat pada kategori tiket go show dan informasi terkait masing – masing paket tiket.

Keterangan rancangan antarmuka halaman informasi paket tiket go show

1. Alamat URL digunakan wisatawan untuk mengakses halaman paket tiket go show
2. *Text* berisi judul halaman "Informasi Paket Tiket Go Show"
3. *Card image* berisi gambar paket tiket
4. *Text* berisi nama paket tiket
5. *Card body* berisi *text*, *card text,* dan tombol
6. *Text* berisi nama "Detail Paket"dan *Card Text* berisi informasi harga dan tiket tersedia
7. Tombol dengan nama "BELI" digunakan wisatawan dapat melakukan pembelian paket tiket dengan menekan tombol "BELI"
8. *Text* "Detail Tiket", wisatawan dapat menekan *text* "Detail Tiket" untuk mengetahui kegiatan dan fasilitas dari paket tiket tersebut
9. *Card* berisi card *image, card body, card text*, *text* dan tombol



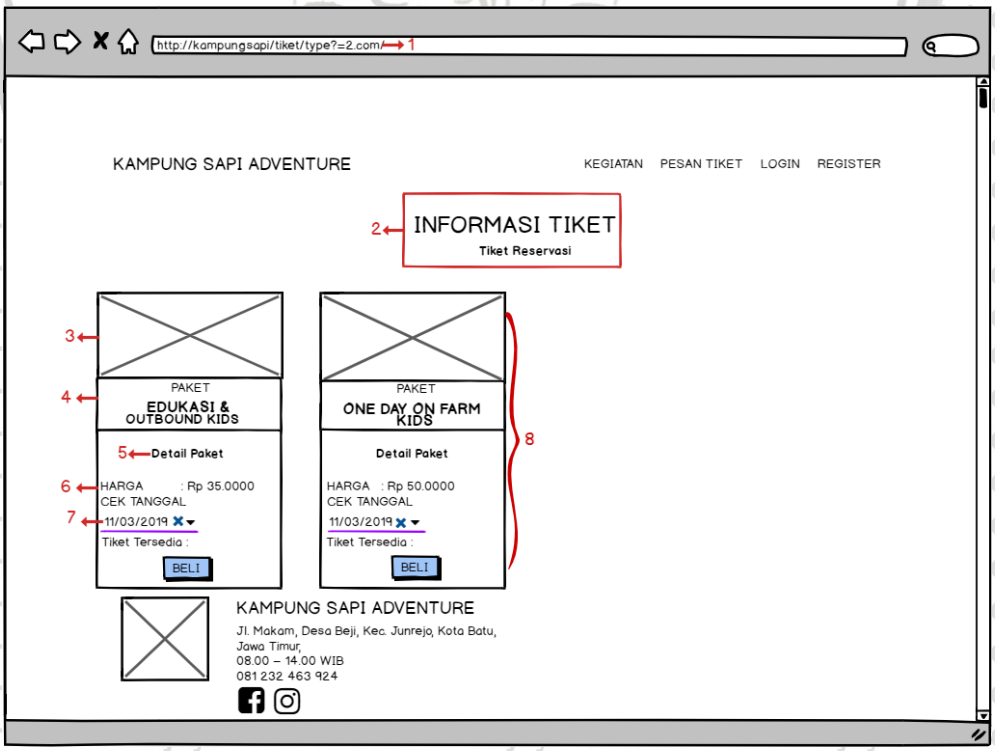
Gambar 4.43 Rancangan Antarmuka Halaman Kategori Tiket Reservasi

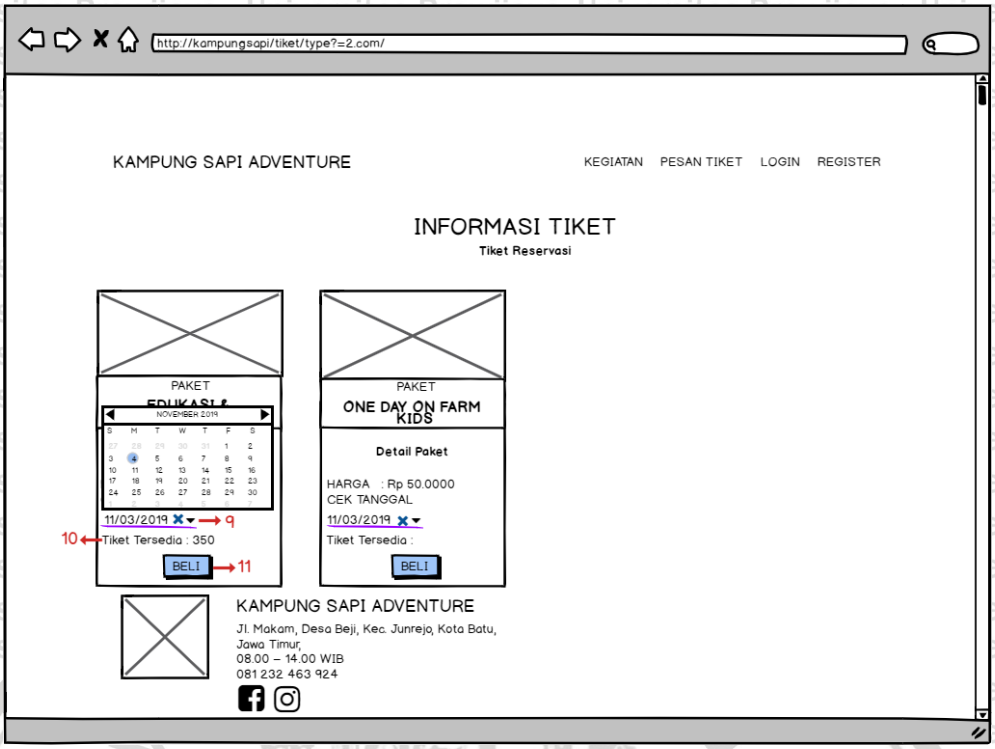
Sumber: (Penulis, 2019)

Perancangan antarmuka pada Gambar 4.43 merupakan tampilan antarmuka halaman Tiket Reservasi. Halaman ini ditampilkan setelah wisatawan memilih menu pesan tiket.

Keterangan rancangan antarmuka halaman kategori tiket reservasi

1. Menu "Pesan Tiket"
2. Alamat URL digunakan wisatawan untuk mengakses halaman tiket pada aplikasi *e-commerce* Wisata Kampung Sapi *Adventure*
3. *Text* berisi judul halaman "Pilih Kategori Tiket"
4. *Card* berisi *text* dan tombol
5. *Text* berisi keterangan nama kategori tiket reservasi
6. Tombol dengan nama "PILIH" digunakan wisatawan untuk memilih kategori tiket reservasi dengan melakukan klik pada tombol "PILIH"
7. *Text* berisi petunjuk untuk wisatawan memilih salah satu jenis paket tiket
8. *Icon window close* untuk menutup *modal*
9. Tombol dengan nama jenis paket wisatawan dapat memilih salah satu jenis paket tiket dengan menekan pada tombol tersebut
10. *Modal* berisi *text, icon window close*, dan tombol





Gambar 4.44 Perancangan Antarmuka Informasi Paket Tiket Reservasi

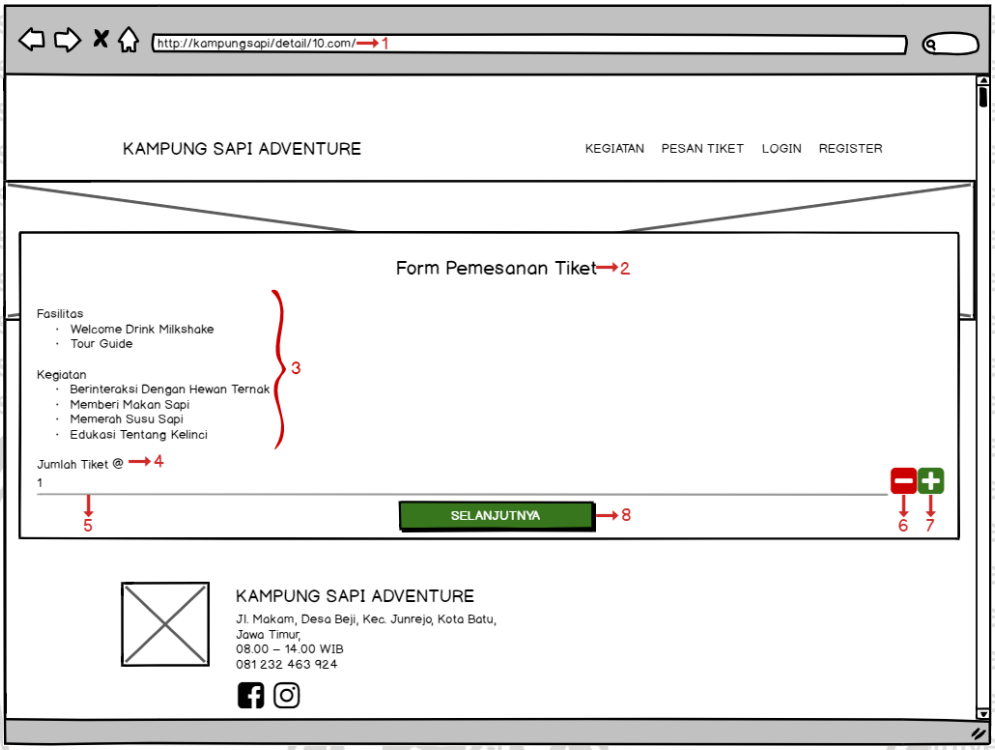
Sumber: (Penulis, 2019)

Perancangan antarmuka pada Gambar 4.44 merupakan tampilan antarmuka halaman informasi paket Tiket Reservasi. Halaman ini ditampilkan setelah wisatawan memilih kategori Tiket Reservasi. Pada halaman ini wisatawan dapat mengetahui paket tiket apa saja yang terdapat pada kategori Tiket Reservasi dan informasi terkait masing – masing paket tiket.

Keterangan rancangan antarmuka halaman informasi tiket reservasi

1. Alamat URL digunakan wisatawan untuk mengakses halaman informasi tiket reservasi
2. *Text* berisi judul halaman "Informasi Paket Tiket Reservasi"
3. *Card Image* berisi gambar paket tiket
4. *Text* berisi nama paket tiket
5. *Text* berisi "Detail Tiket" dapat digunakan wisatawan untuk mengetahui kegiatan dan fasilitas dari paket tiket tersebut
6. *Text* berisi informasi harga paket tersebut
7. *Date* digunakan untuk cek tanggal (wisatawan yang akan membeli paket tersebut diharuskan untuk melakukan cek tanggal terlebih dahulu)
8. *Card* berisi *card image*, *text*, *date*, dan tombol
9. Inputan berisi tanggal yang telah dipilih oleh wisatawan

10. *Text* berisi tiket tersedia, jumlah tiket tersedia akan tampil jika wisatawan telah memilih tanggal
11. Tombol dengan nama "BELI" digunakan wisatawan untuk melakukan pembelian paket tiket dengan menekan tombol "BELI"



Gambar 4.45 Rancangan Antarmuka Halaman Form Pemesanan Tiket

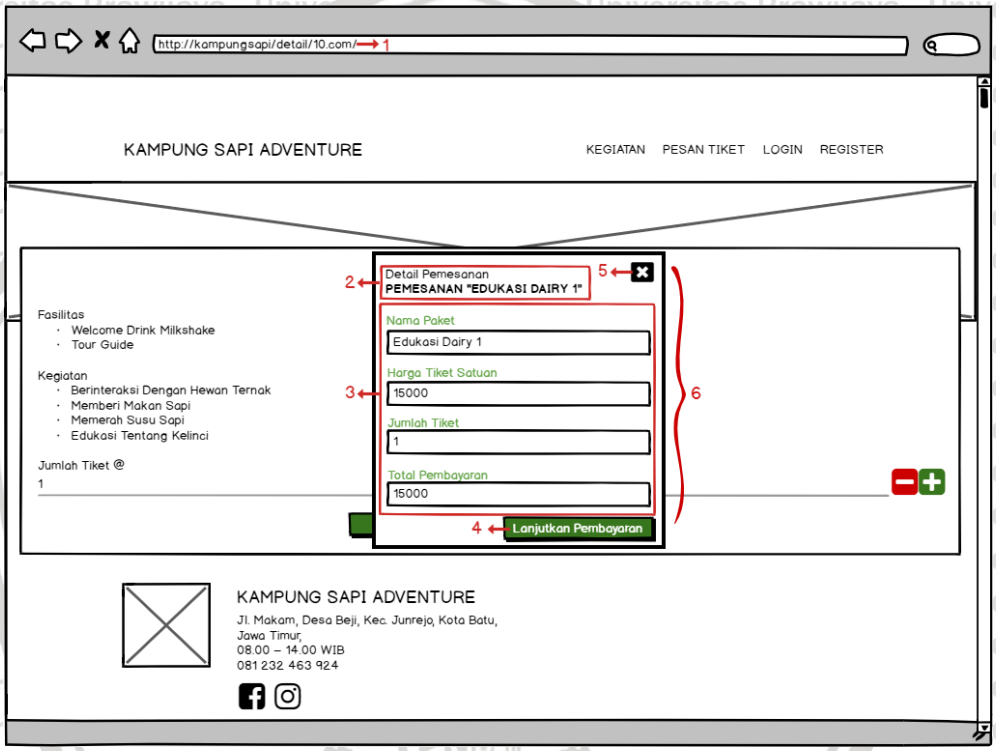
Sumber: (Penulis, 2019)

Perancangan antarmuka pada Gambar 4.45 merupakan tampilan antarmuka halaman form pemesanan tiket. Halaman ini ditampilkan setelah wisatawan memilih kategori tiket go show maupun Tiket Reservasi. Pada halaman ini wisatawan dapat mengetahui informasi paket yang akan dibeli dan wisatawan diminta untuk menginputka jumlah tiket yang akan dibeli.

Keterangan rancangan antarmuka halaman form pemesanan tiket

1. Alamat URL digunakan wisatawan untuk mengakses halaman form pemesanan tiket
2. *Text* berisi judul halaman "Form Pemesanan Tiket"
3. *Text* berisi fasilitas dan kegiatan paket tiket yang akan dibeli
4. *Text* berisi jumlah tiket digunakan untuk wisatawan yang akan membeli tiket diminta untuk menginputkan jumlah tiket pada *text input* yang telah tertera

5. *Text input* digunakan untuk wisatawan meginputkan jumlah tiket yang akan dibeli
6. Tombol *minus* digunakan untuk wisatawan mengurangi jumlah tiket yang akan dibeli
7. Tombol *plus* digunakan untuk menambah jumlah tiket yang akan dibeli
8. Tombol dengan nama "SELANJUTNYA" digunakan wisatawan yang telah menginputkan jumlah tiket dan ingin melanjutkan proses pembelian dapat menekan tombol "SELANJUTNYA"



Gambar 4.46 Perancangan Antarmuka Halaman Detail Tiket

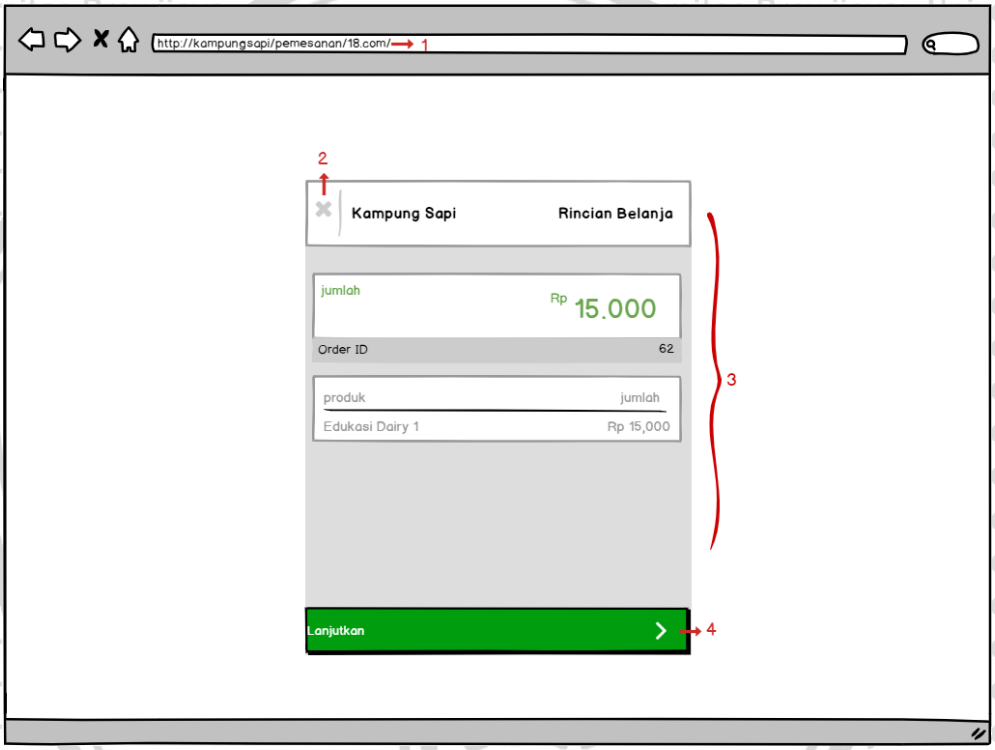
Sumber: (Penulis, 2019)

Perancangan antarmuka pada Gambar 4.46 merupakan tampilan antarmuka halaman detail tiket. Halaman ini ditampilkan setelah wisatawan menekan tombol "SELANJUTNYA" yang terdapat pada halaman form pemesanan tiket. Pada halaman ini wisatawan dapat mengetahui informasi detail pemesanan pada paket tiket yang akan dibeli.

Keterangan rancangan antarmuka halaman detail tiket

1. Alamat URL untuk mengakses halaman detail tiket
2. *Text* berisi judul *modal* " Detail Pemesanan" yang disertai nama paket tiket yang dipesan

3. Terdapat informasi terkait detail pemesanan tiket yaitu, nama paket, harga tiket satuan, jumlah tiket, dan total pembayaran (informasi tersebut bersifat *read only,* jadi wisatawan hanya dapat melihat informasi tersebut)
4. Tombol dengan nama "Lanjutkan Pembayaran" digunakan wisatawan yang ingin melanjutkan untuk menyelesaikan proses pembayaran dengan menekan tombol "Lanjutkan Pembayaran"
5. *Icon window close* untuk menutup *modal*
6. *Modal* berisi *text,* tombol, dan *icon window close.*



Gambar 4.47 Perancangan Antarmuka Halaman Pembayaran Pada Midtrans

Sumber: (Penulis, 2019)

Perancangan antarmuka pada Gambar 4.47 merupakan tampilan antarmuka halaman pembayaran pada midtrans. Pada halaman ini wisatawan dapat melanjutkan proses pembayaran pemesanan tiket.

Keterangan rancangan antarmuka pemesanan pada midtrans

1. Alamat URL digunakan wisatawan untuk mengakses halaman pemesanan
2. *Icon close* yang dapat digunakan wisatawan untuk memilih tidak melanjutkan pembayaran
3. Modal berisi informasi nama organisasi dan rincian belanja

4. Tombol dengan nama “Lanjutkan” (wisatawan dapat melakukan klik pada tombol “Lanjutkan” jika ingin melanjutkan pembayaran pemesanan tiket

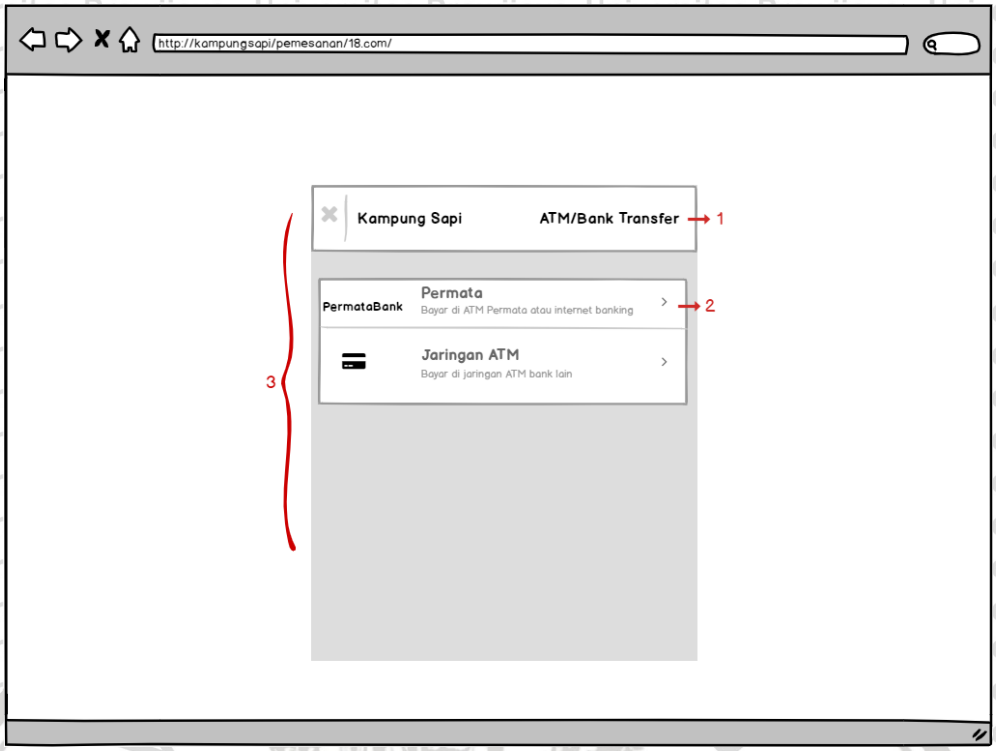
Gambar 4.48 Perancangan Antarmuka Halaman Memilih Metode Pembayaran Pada Midtrans

Sumber: (Penulis, 2019)

Perancangan antarmuka pada Gambar 4.48 ini merupakan tampilan antarmuka halaman memilih metode pembayaran pada midtrans. Pada halaman ini untuk dapat melanjutkan proses pembayaran pemesanan tiket, wisatawan diharuskan untuk memilih salah satu metode pembayaran yang telah disediakan oleh midtrans.

Keterangan rancangan antarmuka pilih pembayaran pada midtrans

1. Modal berisi informasi nama organisasi dan metode pembayaran yang telah disediakan midtrans
2. Terdapat menu metode pembayaran digunakan wisatawan untuk melakukan pembayaran dengan memilih salah satu metode pembayaran



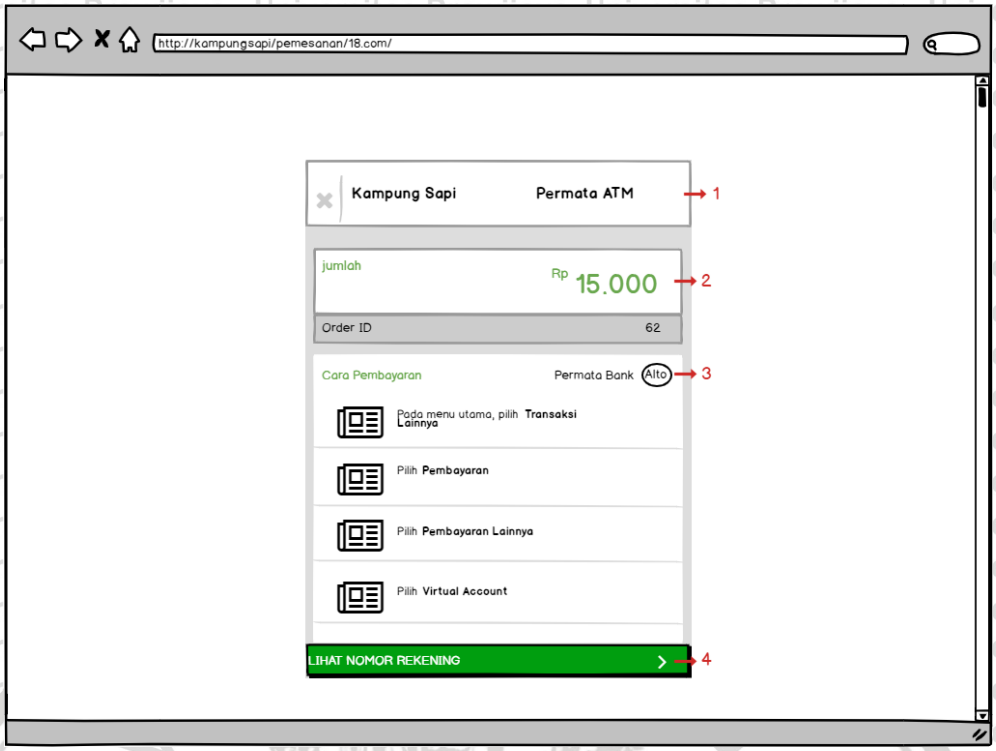
Gambar 4.49 Perancangan Antarmuka Halaman Metode Pembayaran Menggunakan ATM/Bank Transfer Pada Midtras

Sumber: (Penulis, 2019)

Perancangan antarmuka pada Gambar 4.49 ini merupakan tampilan antarmuka halaman metode pembayaran menggunakan atm/bank transfer. Pada halaman ini wisatawan memilih melakukan pembayaran melalui Permata Bank atau jaringan atm bank lain

Keterangan rancangan antarmuka pada salah satu metode pembayaran

1. Terdapat *text* yang berisi informasi metode pembayaran yang telaah dipilih wisatawan
2. Terdapat pilihan menu Permata yang artinya wisatawan jika ingin melakukan pembayaran melalui Permata Bank dapat memilih menu tersebut sedangkan jika wisatawan ingin melakukan pembayaran melalui bank lain maka dapat memilih menu Jaringan ATM
3. Modal berisi text dan menu



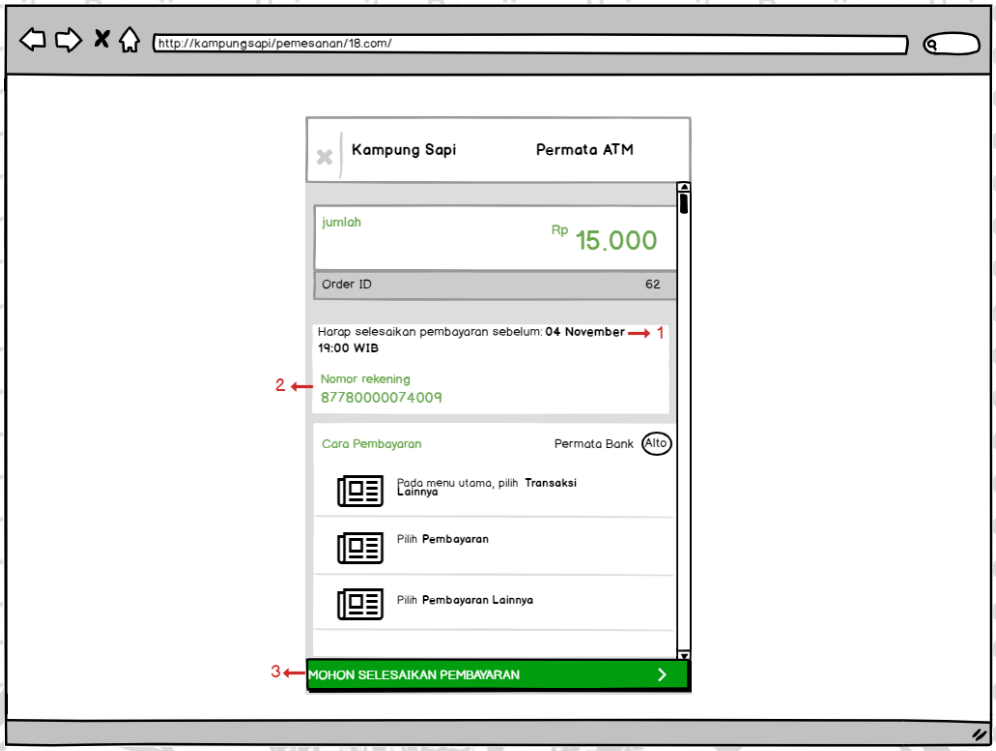
Gambar 4.50 Perancangan Antarmuka Halaman Cara Pembayaran Melalui Permata Pada Midtrans

Sumber: (Penulis, 2019)

Perancangan antarmuka pada Gambar 4.50 merupakan tampilan antarmuka halaman metode pembayaran melalui Permata Bank. Pada halaman ini wisatawan dapat mengetahui jumlah yang akan dibayarkan dan informasi cara pembayaran melalui Permata Bank

Keterangan rancangan antarmuka halaman metode pembayaran melalui Permata Bank

1. Terdapat *text* yang berisi informasi pembayaran melalui Pertama ATM
2. Terdapat *text* yang berisi informasi jumlah yang harus dibayarkan dan order ID
3. Terdapat *text* yang berisi informasi cara pembayaran melalui Permata Bank
4. Tombol dengan nama "LIHAT NOMOR REKENING" digunakan wisatawan untuk melanjutkan pembayaran ke nomor rekening Permata Bank dengan menekan "LIHAT NOMOR REKENING"



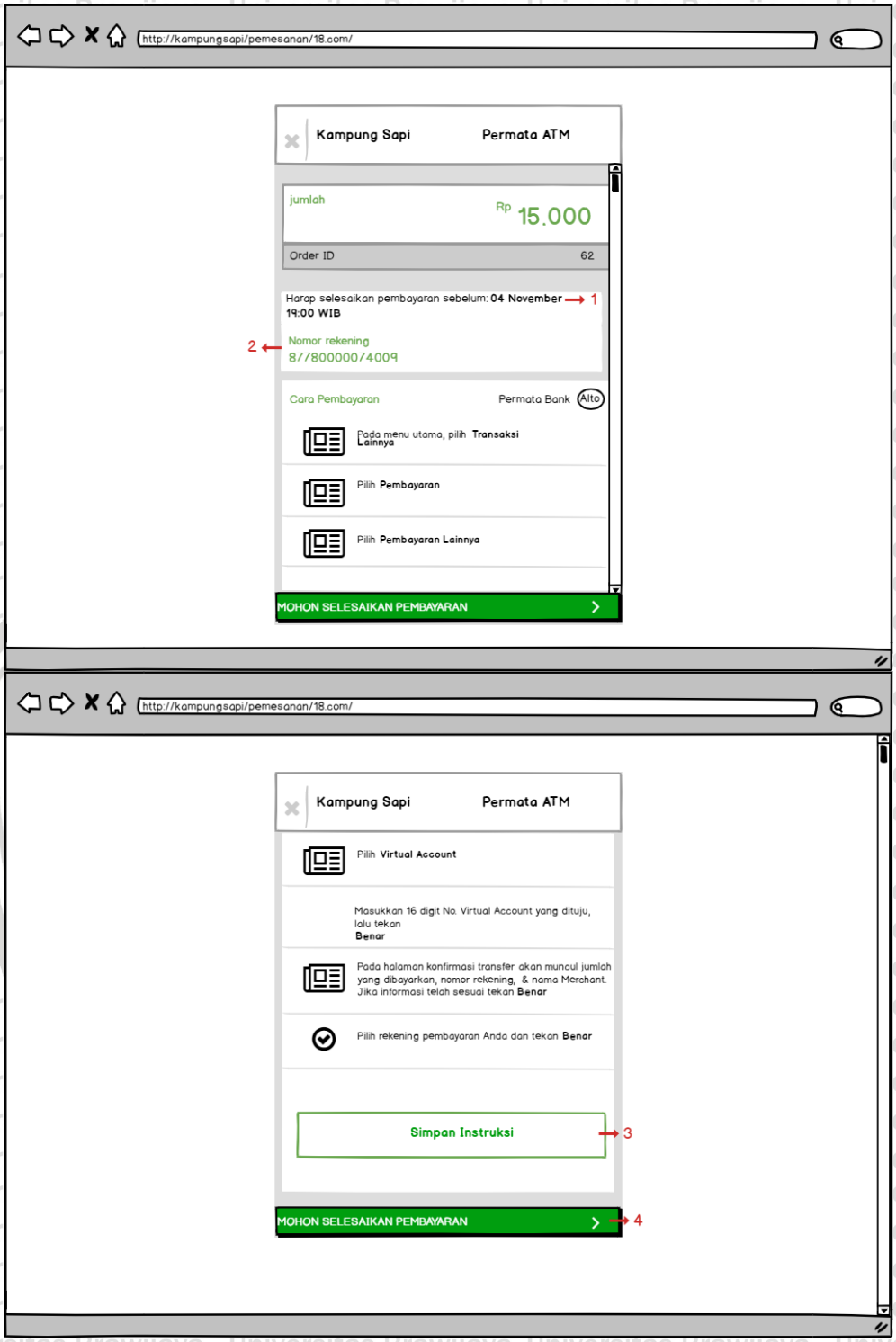
Gambar 4.51 Perancangan Antarmuka Halaman Cara Pembayaran Melalui Permata Bank Pada Midtrans

Sumber: (Penulis, 2019)

Perancangan antarmuka pada Gambar 4.51 merupakan tampilan antarmuka halaman informasi pembayaran melalui Permata Bank. Pada halaman ini wisatawan dapat mengetahui informasi jumlah yang harus dibayarkan, batas waktu pembayaran, nomor rekening yang dituju, dan cara pembayaran.

Keterangan rancangan antarmuka halaman metode pembayaran melalui Permata Bank

1. *Text* berisi informasi batas waktu pembayaran
2. *Text* berisi informasi nomor rekening yang dituju
3. *Text* berisi informasi cara pembayaran melalui Permata Bank
4. Tombol dengan nama "MOHON SELESAIKAN PEMBAYARAN" digunakan wisatawan untuk menyelesaikan pembayaran dengan menekan tombol "MOHON SELESAIKAN PEMBAYARAN"



Gambar 4.52 Perancangan Antarmuka Halaman Metode Pembayaran Melalui Permata Bank Pada Midtrans

Sumber: (Penulis, 2019)

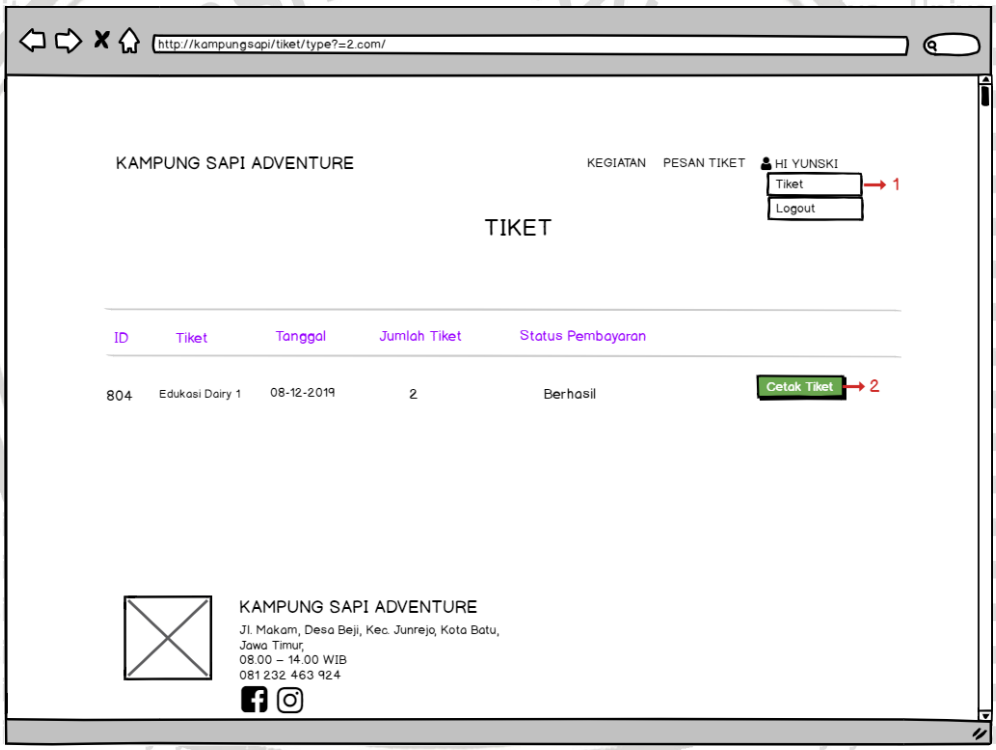
Perancangan antarmuka pada Gambar 4.52 merupakan tampilan antarmuka halaman informasi pembayaran melalui Permata Bank. Pada halaman

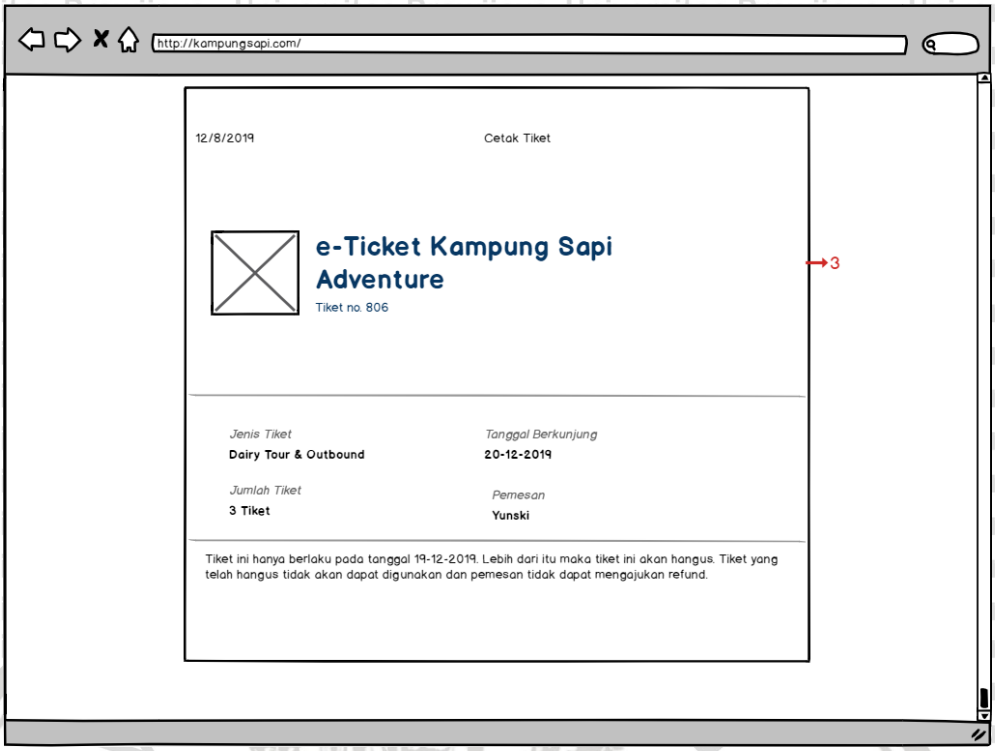
ini wisatawan dapat mengetahui informasi jumlah yang harus dibayarkan, batas waktu pembayaran, nomor rekening yang dituju, dan cara pembayaran.

Keterangan rancangan antarmuka halaman metode pembayaran melalui Permata Bank

1. *Text* berisi informasi batas waktu pembayaran
2. *Text* berisi informasi nomor rekening yang dituju
3. Terdapat tombol dengan nama "Simpan Instruksi", wisatawan diharuskan untuk melakukan klik pada tombol simpan instruksi
4. Tombol dengan nama "MOHON SELESAIKAN PEMBAYARAN" digunakan wisatawan untuk menyelesaikan pembayaran dengan menekan Tombol "MOHON SELESAIKAN PEMBAYARAN"

Gambar 4.53 Implementasi Antarmuka Halaman Pada Menu Tiket



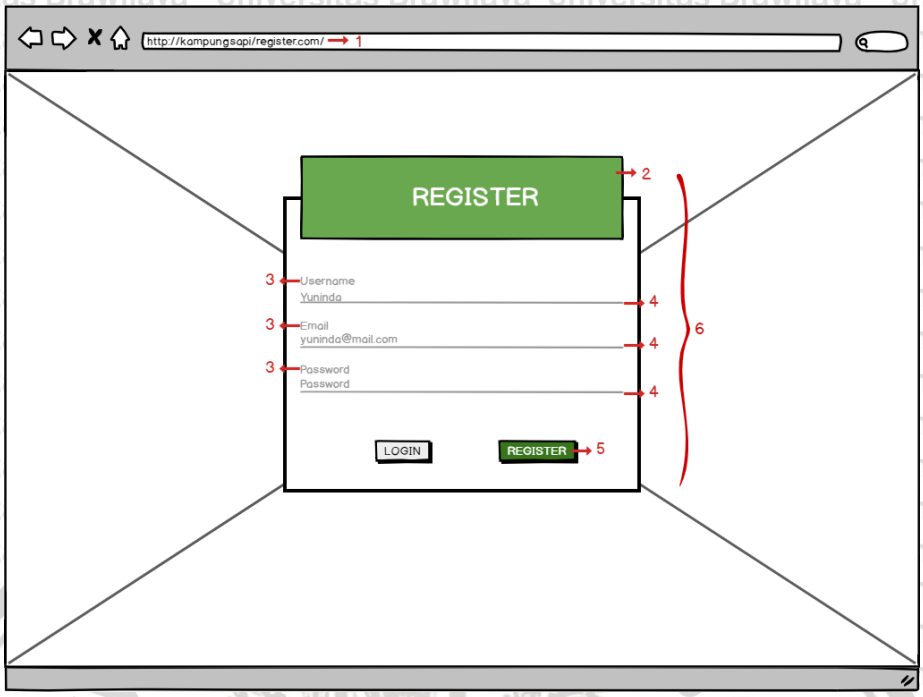


Perancangan antarmuka pada Gambar 4.53 merupakan tampilan antarmuka halaman pada menu tiket. Pada halaman ini wisatawan dapat mengetahui informasi status pembayaran tiket dan mencetak tiket

Keterangan rancangan antarmuka halaman menu tiket

1. Menu “Tiket” digunakan wisatawan untuk melihat status pembayaran dan mencetak tiket
2. Button “Cetak Tiket” digunakan wisatawan untuk mencetak tiket
3. Tiket elektronik Wisata Kampung Sapi Adventure

4.10.3 Perancangan Antarmuka Register



Gambar 4.54 Perancangan Antarmuka Register

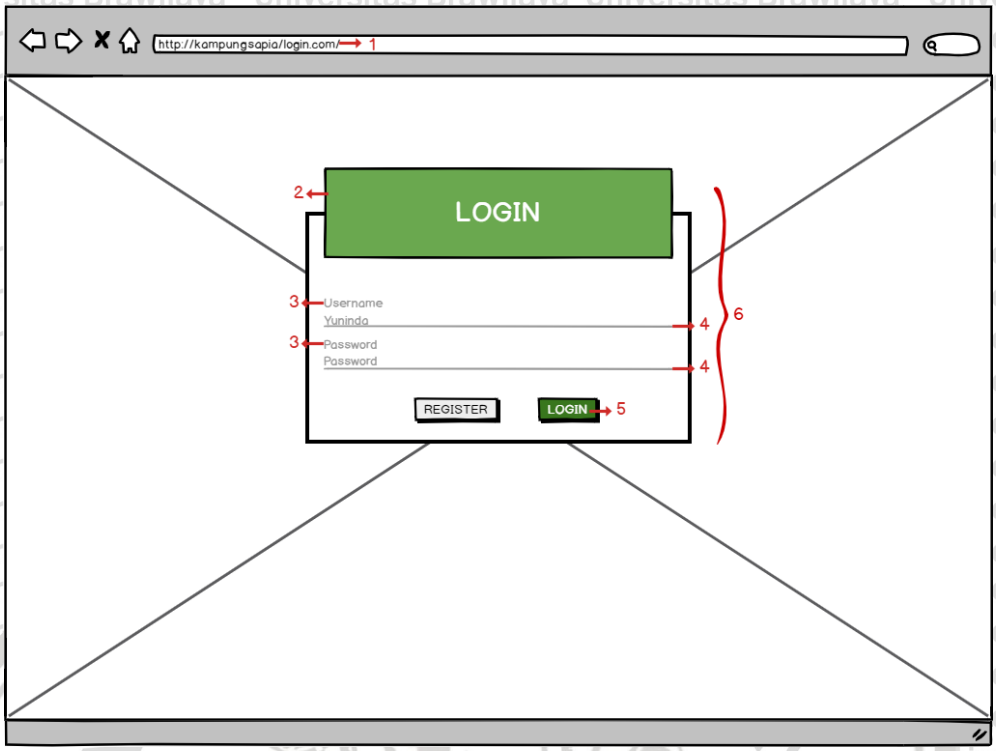
Sumber: (Penulis, 2019)

Perancangan antarmuka pada Gambar 4.54 ini merupakan tampilan antarmuka halaman register. Pada halaman ini wisatawan dapat melakukan register dengan menginputkan *username*, *password,* dan email.

Keterangan rancangan antarmuka register

1. Alamat URL digunakan wisatawan untuk mengakses halaman register
2. *Text* berisi judul *card* "REGISTER"
3. *Text* berisi *username*, *password,* dan email yang digunakan untuk memberikan informasi terkait data yang diinputkan untuk melakukan register
4. *Text input* yang digunakan wisatawan untuk menginputkan *username*, *password,* dan email
5. Tombol dengan nama "REGISTER" digunakan wisatawan untuk menyimpan *username password,* dan email yang telah diinputkan dengan menekan tombol "REGISTER"
6. *Card* berisi *text, text input,* dan tombol

4.10.4 Perancangan Antarmuka *Login*



Gambar 4.55 Perancangan Antarmuka *Login*

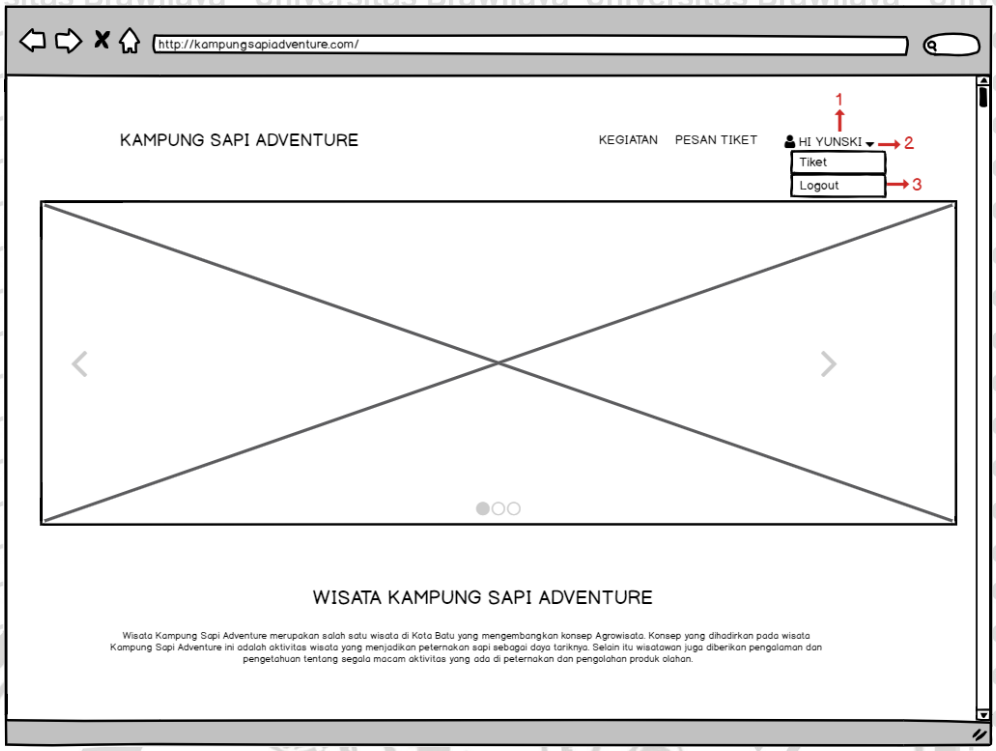
Sumber: (Penulis, 2019)

Perancangan antarmuka pada Gambar 4.55 merupakan tampilan antarmuka halaman *login*. Halaman ini ditampilkan pada dua kondisi yaitu wisatawan memilih menu *login* yang terdapat pada halaman utama aplikasi dan wisatawan setelah menekan tombol "Lanjutkan Pembayaran" yang terdapat pada halaman detail tiket. Pada halaman ini wisatawan dapat melakukan *login* dengan menginputkan *username* dan *password*.

Keterangan rancangan antarmuka *login*

1. Alamat URL digunakan wisatawan untuk mengakses halaman *login*
2. *Text* berisi judul *card* "LOGIN"
3. *Text* berisi *username* dan *password* yang digunakan untuk memberikan informasi data diinputkan untuk melakukan *login*
4. *Text input* yang digunakan wisatawan untuk menginputkan *username* dan *password*
5. Tombol dengan nama "LOGIN" digunakan wisatawan melakukan *login* dengan menekan tombol "LOGIN"
6. *Card* yang berisi *text, text input,* dan tombol

4.10.5 Perancangan Antarmuka Halaman *Logout*



Gambar 4.56 Perancangan Antarmuka Halaman *Logout* Pada Wisatawan

Sumber: (Penulis, 2019)

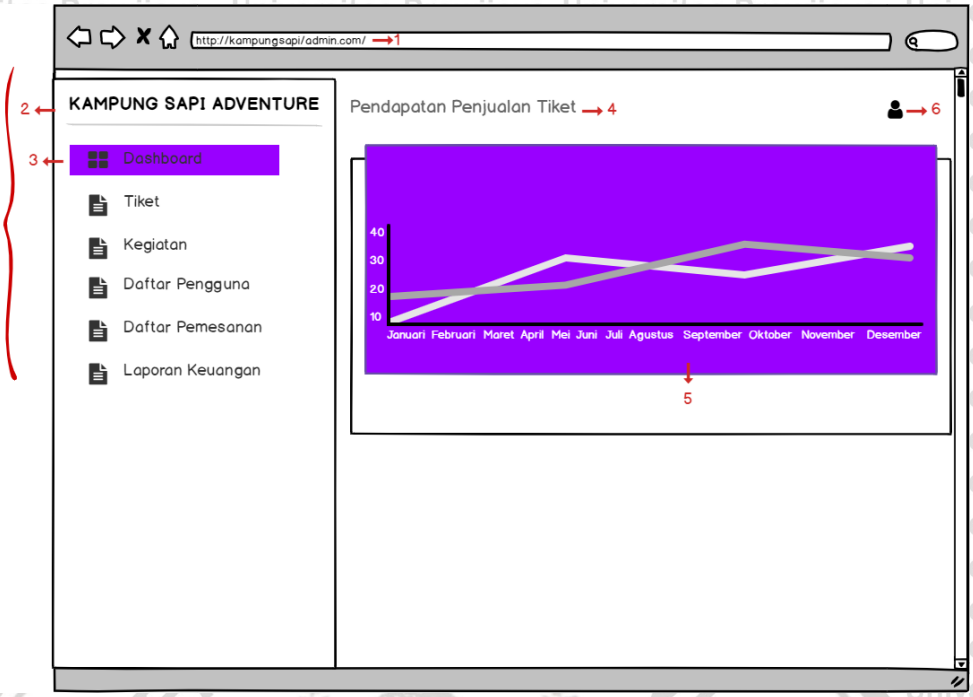
Perancangan antarmuka pada Gambar 4.56 merupakan tampilan antarmuka halaman *logout* pada wisatawan. Pada halaman ini wisatawan yang telah melakukan *login* dapat melakukan *logout*.

Keterangan rancangan antarmuka *logout* pada wisatawan

1. Terdapat *username* wisatawan yang telah melakukan *login*
2. Terdapat menu *dropdown*
3. Menu "LOGOUT" digunakan wisatawan dapat melakukan *logout* dengan memilih menu "LOGOUT"

4.10.6 Perancangan Antarmuka Halaman *Dashboard* Admin

Perancangan antarmuka pada Gambar 4.57 merupakan tampilan antarmuka halaman *dashboard* admin. Pada halaman ini admin dapat mengetahui informasi pendapatan penjualan tiket melalui grafik.



Gambar 4.57 Perancangan Antarmuka Halaman *Dashboard* Admin

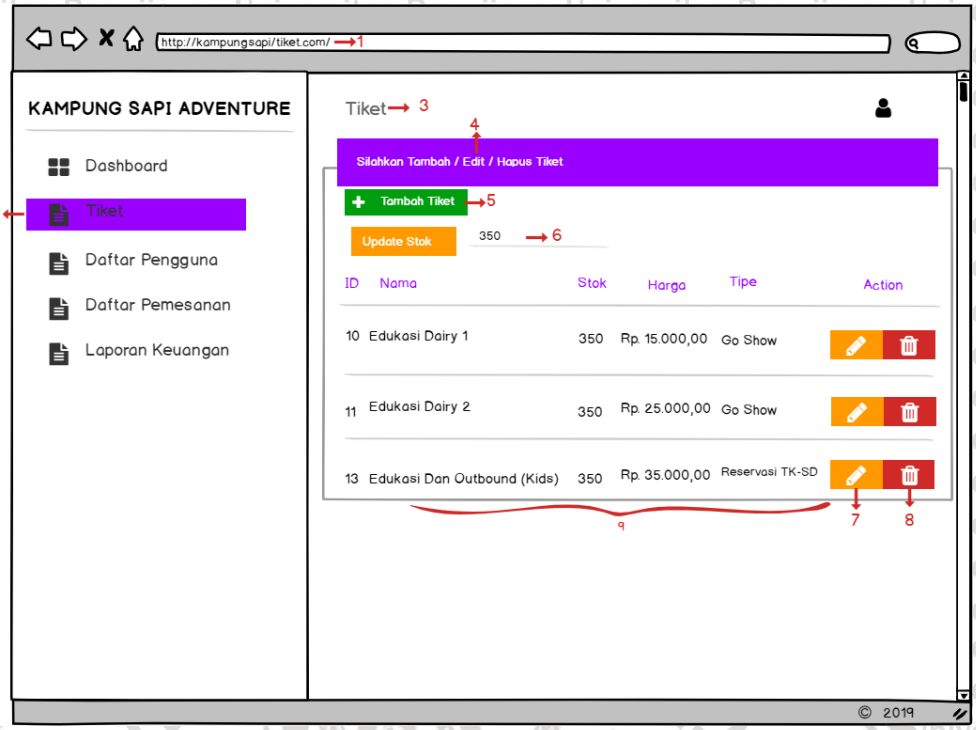
Sumber: (Penulis, 2019)

Keterangan rancangan antarmuka halaman *dashboard* admin

1. Alamat URL digunakan admin untuk mengakses halaman *dashboard* admin
2. *Text* berisi Kampung *Sapi Adventure*
3. Menu *dashboard* digunakan admin untuk melihat grafik pendapatan penjualan tiket
4. *Text* berisi judul halaman "Pendapatan Penjualan Tiket"
5. *Line chart* berguna untuk menampilkan tinggi rendahnya pendapatan penjualan tiket
6. *Icon user* digunakan admin untuk melakukan *logout* dengan menekan
7. *Sidebar* berisi menu dashboard, tiket, daftar pengguna, daftar pemesanan, dan laporan keuangan

4.10.7 Perancangan Antarmuka Halaman Menu Tiket

Perancangan antarmuka pada Gambar 4.58 merupakan tampilan antarmuka halaman menu tiket. Pada halaman ini admin dapat mengelola data tiket dengan melakukan CRUD (*create, read, update,* dan *delete*).



Gambar 4.58 Perancangan Antarmuka Halaman Menu Tiket

Sumber: (Penulis, 2019)

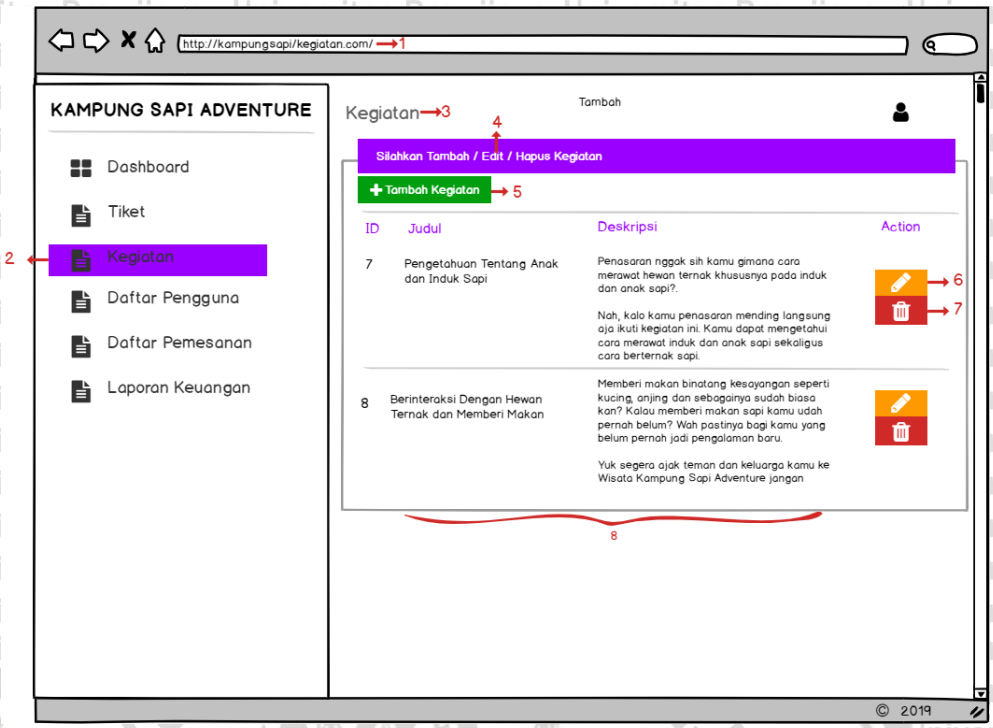
Keterangan rancangan antarmuka halaman tiket

1. Alamat URL digunakan admin untuk mengakses halaman tiket
2. Menu tiket digunakan admin untuk melakukan CRUD yaitu *create, read, update,* dan *delete* pada data tiket
3. *Text* berisi judul halaman "Tiket"
4. *Text* berisi informasi yang dapat dilakukan admin pada halaman tiket
5. Tombol *plus* digunakan untuk menambah data tiket
6. Tombol update stok digunakan untuk mengubah data tiket
7. Tombol *pencil* dapat digunakan admin untuk mengubah data tiket
8. Tombol *delete* dapat digunakan admin untuk menghapus data tiket
9. *Table* beserta isinya yang terdiri dari kolom ID, nama, stok, harga, *type*, dan *action*

4.10.8 Perancangan Antarmuka Halaman Kegiatan

Perancangan antarmuka pada Gambar 4.59 merupakan tampilan antarmuka halaman kegiatan. Pada halaman ini admin dapat melakukan CRUD (*create, read, update,* dan *delete*).

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA



Gambar 4.59 Perancangan Antarmuka Halaman Kegiatan

Sumber: (Penulis, 2019)

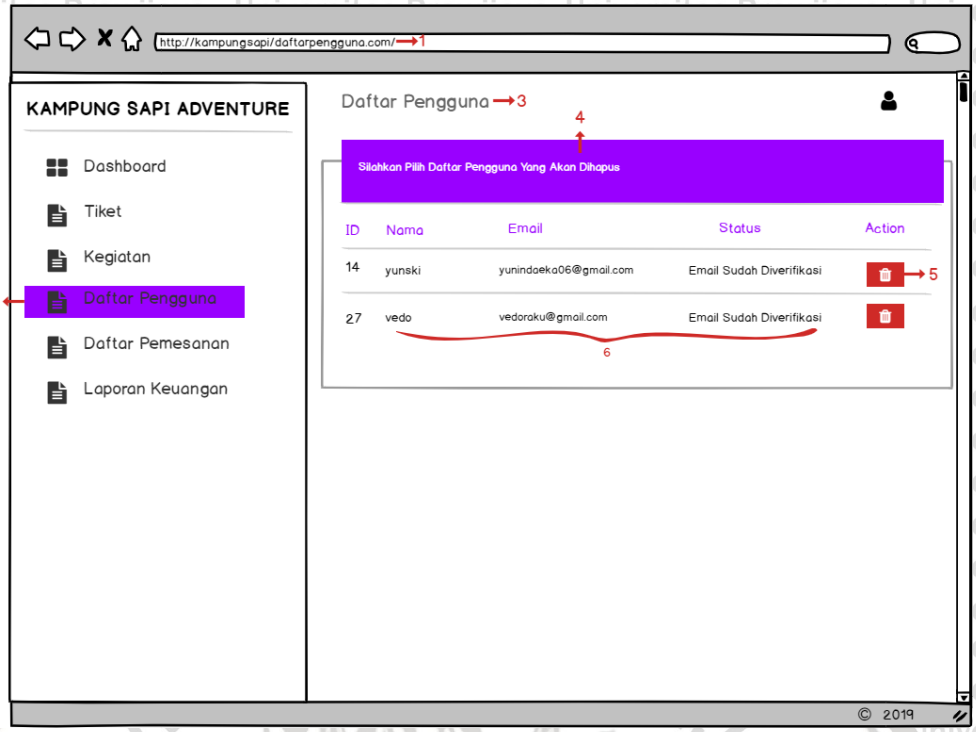
Keterangan rancangan antarmuka halaman kegiatan

1. Alamat URL untuk mengakses halaman kegiatan
2. Menu kegiatan (admin dapat melakukan CRUD yaitu *create, read, update,* dan *delete*)
3. *Text* berisi judul halaman "Kegiatan"
4. *Text* yang berisi informasi yang dapat dilakukan admin pada halaman kegiatan
5. *Button plus* digunakan untuk menambah kegiatan
6. *Icon pencil* dapat digunakan admin untuk melakukan *update* pada kegiatan
7. *Icon delete* dapat digunakan admin untuk menghapus pada kegiatan
8. *Table* beserta isinya yang terdiri dari kolom ID, judul, deskripsi, dan *action*

4.10.9 Perancangan Antarmuka Halaman Menu Daftar Pengguna

Perancangan antarmuka pada Gambar 4.60 merupakan tampilan antarmuka halaman daftar pengguna. Pada halaman ini admin dapat melihat dan menghapus data daftar pengguna.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA



Gambar 4.60 Perancangan Antarmuka Halaman Daftar Pengguna

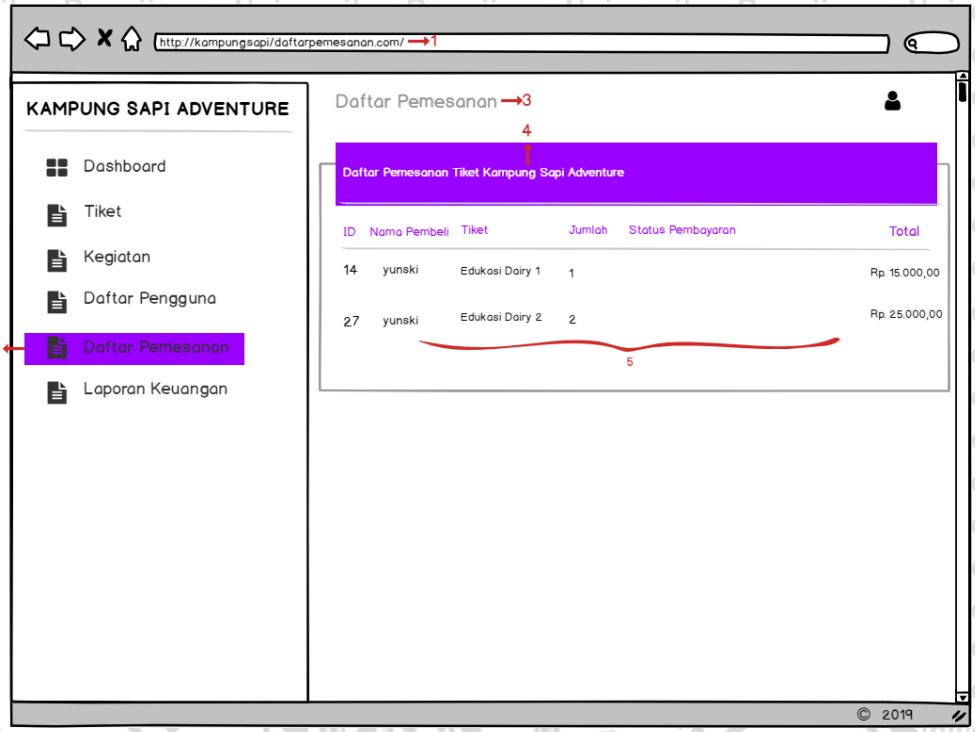
Sumber: (Penulis, 2019)

Keterangan rancangan antarmuka halaman daftar pengguna

1. Alamat URL digunakan admin untuk mengakses halaman daftar pengguna
2. Menu daftar pengguna digunkan admin untuk melihat dan menghapus dta daftar pengguna
3. *Text* berisi judul halaman "Daftar Pengguna"
4. *Text* berisi informasi yang dapat dilakukan admin pada halaman daftar pengguna
5. Tombol *minus* digunakan admin untuk menghapus pada halaman daftar pengguna
6. *Table* yang terdiri dari kolom ID, nama, email, status, dan *action*

4.10.10 Perancangan Antarmuka Halaman Menu Daftar Pemesanan

Perancangan antarmuka pada Gambar 4.61 merupakan tampilan antarmuka halaman menu daftar pemesanan. Pada halaman ini admin dapat melihat daftar pemesanan.



Gambar 4.61 Perancangan Antarmuka Halaman Daftar Pemesanan

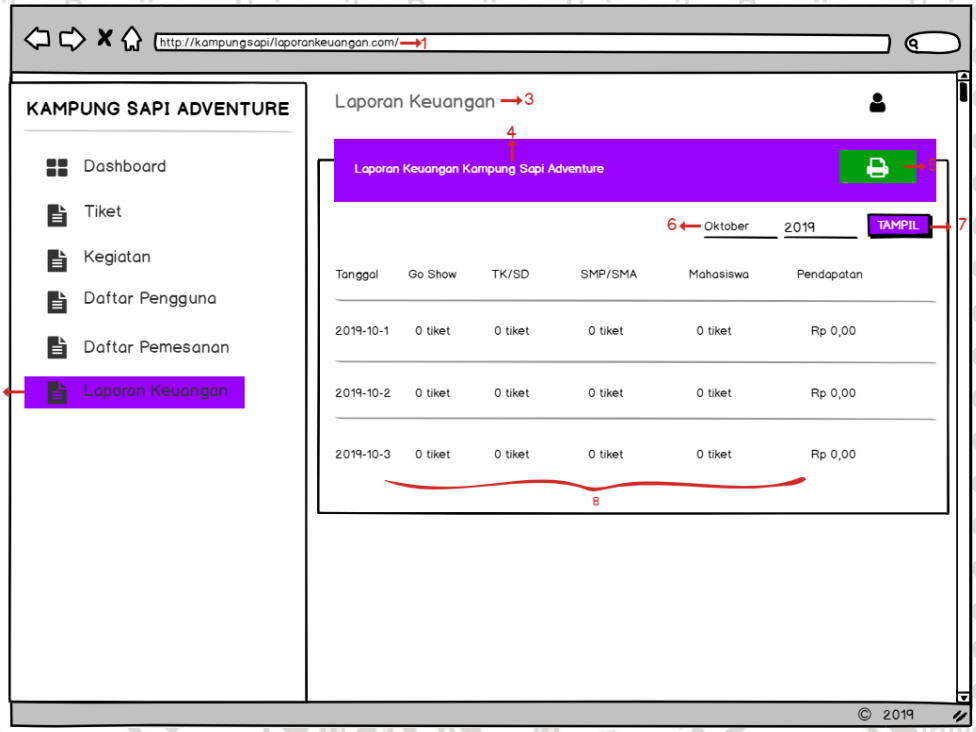
Sumber: (Penulis, 2019)

Keterangan rancangan antarmuka halaman daftar pemesanan

1. Alamat URL digunakan admin untuk mengakses halaman daftar pemesanan
2. Menu daftar pengguna digunakan admin untuk melihat dan menghapus daftar pemesanan)
3. *Text* berisi judul halaman "Daftar Pemesanan"
4. *Text* berisi judul *table* "Daftar Pemesanan Tiket Kampung Sapi *Adventure"*
5. *Table* yang terdiri dari kolom ID, nama pembeli, tiket, jumlah, status pembayaran, metode pembayaran, dan total

4.10.11 Perancangan Antarmuka Halaman Menu Laporan Keuangan

Perancangan antarmuka pada Gambar 4.62 merupakan tampilan antarmuka halaman laporan keuangan. Pada halaman ini admin dapat melihat laporan keuangan berdasarkan bulan dan tahun serta dapat mengunduh laporan keuangan.



Gambar 4.62 Perancangan Antarmuka Halaman Menu Laporan Keuangan

Sumber: (Penulis, 2019)

Keterangan rancangan antarmuka halaman laporan keuangan

1. Alamat URL digunakan admin untuk mengakses halaman laporan keuangan
2. Menu laporan keuangan digunakan admin untuk melihat laporan keuangan berdasarkan bulan dan tahun serta dapat mengunduh laporan keuangan)
3. *Text* berisi judul halaman "Laporan Keuangan"
4. *Text* berisi judul *table* "Laporan Keuangan Kampung Sapi *Adventure"*
5. Tombol *print* digunakan admin untuk mengunduh laporan keuangan
6. *Admin* dapat melihat laporan keuangan berdasarkan bulan yang diinginkan dengan memilih bulan dan tahun
7. Tombol dengan nama "TAMPIL" digunakan admin untuk menampilkan laporan berdasarkan bulan dan tahun yang dipilih dengan menekan Tombol "TAMPIL"
8. *Table* terdiri dari kolom tanggal, Go Show, TK/SD, SMP/SMA, Mahasiswa, dan pendapatan

BAB 5 IMPLEMENTASI DAN PENGUJIAN

Pada bagian ini akan dijelaskan mengenai implementasi dalam membangun sistem yang terdiri dari penjelasan spesifikasi lingkungan implementasi baik spesifikasi perangkat keras maupun perangkat lunak, batasan implementasi, implementasi kode program sistem, serta implementasi antarmuka dari rancangan pada bab sebelumnya.

5.1 Spesifikasi Lingkungan Implementasi

Pada sub bab ini menjelaskan spesifikasi lingkungan implementasi yang digunakan dalam membangun sistem ini yang terdiri dari spesifikasi perangkat keras dan perangkat lunak.

5.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam membangun sistem ini dapat dilihat pada Tabel 5.1.

Tabel 5.1 Spesifikasi Perangkat Keras Komputer

Laptop	
Processor	Intel® Core™ i3-500 5U CPU @ 2.00Ghz (4 CPUs)
***Memory* (RAM)**	4096MB
Graphic Card	Intel® HD Graphics 5500

Sumber: (Penulis, 2019)

5.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat keras yang digunakan dalam membangun sistem ini dapat dilihat pada Tabel 5.2.

Tabel 5.2 Spesifikasi Perangkat Lunak Komputer

Laptop	
Sistem Operasi	Windows 8.1 Pro 64-bit
Text Editor	Visual Studio Code versi 1.39.2
Web Server	Xampp versi 7.1.32-0
Bahasa Pemrograman	PHP, HTML, CSS, dan Javascript

Sumber: (Penulis, 2019)

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

5.2 Batasan Implementasi

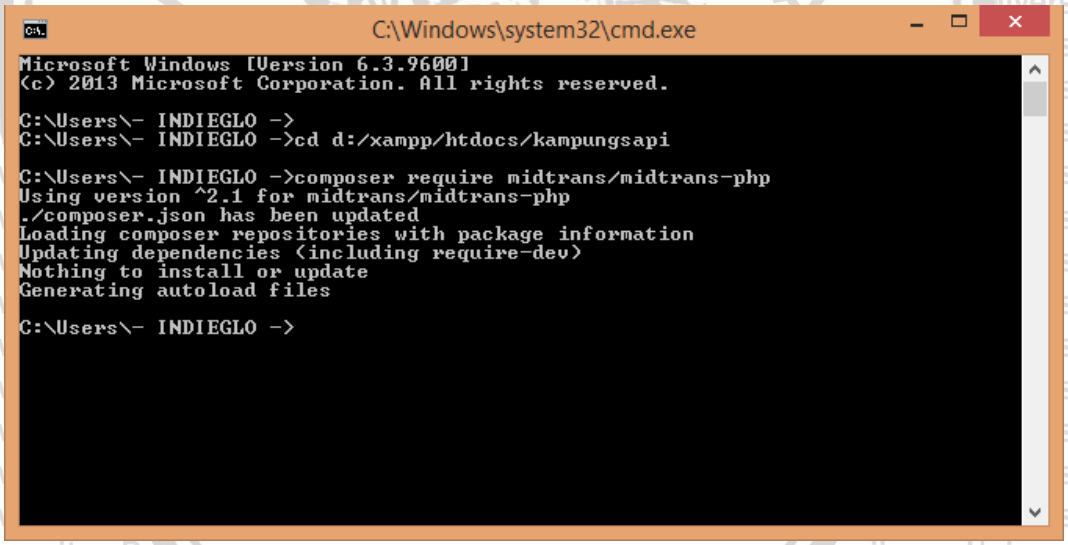
Batasan implementasi pada pembangunan sistem ini adalah sebagai berikut:

1. Aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* harus dijalankan menggunakan koneksi internet.
2. Aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* dikembangkan menggunakan konsep web dengan memanfaatkan API sandbox dari midtrans dengan *framework CodeIgniter*.

5.3 Konfirgurasi Penggunaan Layanan API *Sandbox* Pada *Payment Gateway* Midtrans

Pada bagian ini akan dijelaskan konfirgurasi penggunaan layanan API *sandbox* pada *payment gateway* midtrans. Berikut ini adalah tahapan – tahapan untuk melakukan konfirgurasi penggunaan layanan API sandbox pada *payment gateway* midtrans:

1. Melakukan instalasi *composer* pada file PHP untuk menggunakan *library* dari midtrans.

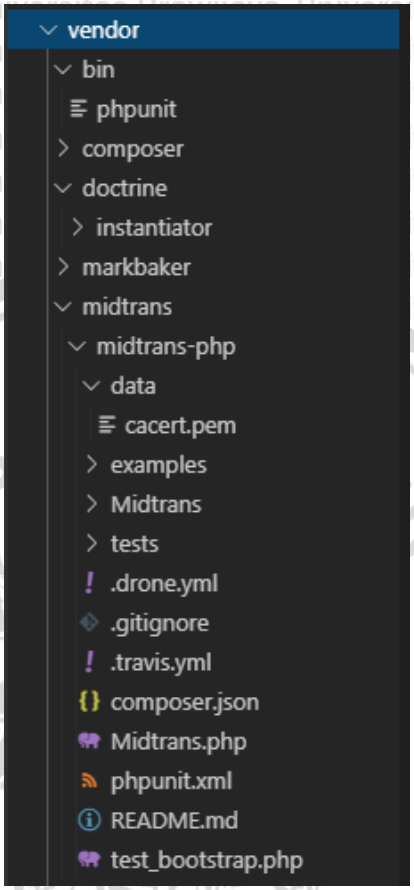


Gambar 5.1 *Installasi Composer*

Sumber: (Penulis, 2019)

Pada Gambar 5.1 merupakan instalasi *composer* pada direktori file PHP yang dilakukan melalui terminal. Pada penelitian ini instalasi *composer* dilakukan pada direktori d dan pada folder PHP dengan nama kampungsapi. Proses instalasi *composer* dilakukan dengan menuliskan *syntax* `cd d:/xampp/htdocs/kampungsapi`. Setelah itu menjalankan *syntax* instalasi *composer* yaitu `composer require midtrans/midtrans-php`. Jika proses dari instalasi *composer* berhasil, maka akan men*generated* file *autoload* pada folder PHP.

2. Instalasi *composer* berhasil, kemudian pada file php secara otomatis terdapat folder midtrans yang terdapat pada folder *vendor.* Di dalam folder midtrans terdapat folder midtrans-php yang berisi kode program penggunaan dan testing midtrans.



Gambar 5.2 *Library Midtrans*

Sumber: (Penulis, 2019)

3. Membuat kode program mengenai konfigurasi *composer autoload* pada file *config.php* untuk mengakses *library* midtrans pada folder *vendor.* Kode program yang digunakan untuk konfigurasi *composer autoload* yaitu

```
$config['composer_autoload'] = 'vendor/autoload.php';
```

```
$config['composer_autoload'] = 'vendor/autoload.php';
```

Gambar 5.3 Konfigurasi *Composer Autoload*

Sumber: (Penulis, 2019)

5.4 Implementasi Kode Program

Implementasi kode program dibuat berdasarkan kebutuhan fungsional yang terdapat pada aplikasi *e-commerce* Wisata Kampung Sapi *Adventure*. Kode program ditulis menggunakan bahasa pemrograman PHP, HTML, Java*script*, dan CSS.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 5.3 Kode Program *Payment Gateway* Midtrans

```
1  \Midtrans\Config::$serverKey = '';
2  \Midtrans\Config::$isProduction = false;
3  \Midtrans\Config::$isSanitized = true;
4  \Midtrans\Config::$is3ds = true;
```

Sumber: (Penulis, 2019)

Kode program pada Tabel 5.3 merupakan kode program untuk menggunakan layanan dari *payment gateway* midtrans.

Tabel 5.4 Kode Program Register, Login, dan Logout

AuthController.php

```
1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  class AuthController extends CI_Controller {
5  public function __construct()
6  {
7  parent::__construct();
8  $this->load->model('AuthModel','authM');
9  }
10 /*
11 Role
12 0 -> user
13 1 -> admin
14
15 public function register()
16     {
17 $data=$this->input->post(NULL, TRUE);
18 if($data){
19 $data['password']=hash('sha512', $data['password']);
20 $data['kode']=substr(md5(mt_rand()), 0, 7);
21 $result=$this->authM->register($data);
22 if ($result['stat']) {
23 $this->sendMail($data['username'],$data['email'],$data['kode']);
24 $this->session->set_flashdata('message', 'Link Verifikasi Sudah Dikirim Ke Email '.$data['email']);
25 redirect(base_url('login'), 'refresh');
26 }else{
27 $this->session->set_flashdata('message', 'Register Gagal, Silahkan Ulangi Lagi');
28 redirect(base_url('register'), 'refresh');
29 }
30 }else{
31 $data['header']['title'] = "Register";
```

Tabel 5.4 Kode Program Register, Login, dan Logout (Lanjutan)

AuthController.php

```
return $this->load->view('auth/register', $data);
}
}

function sendMail($username,$email,$kode)
{
$ci = get_instance();
$this->email->initialize($this->configMail());
$ci->email->from(NAMEORG,NAMEORG);
$list = array($email);
$ci->email->to($list);
$data=base64_encode(base64_encode(json_encode(array($username,$email,$kode))));
$ci->email->subject('Verifikasi Pendaftaran');
$ci->email->message('Silahkan Buka Tautan Berikut Untuk Melakukan Verifikasi Pendaftaran <a href="'.base_url('ver?q=').$data.'">Disini</a>');
$this->email->set_newline("\r\n");
if ($this->email->send()) {
} else {
show_error($this->email->print_debugger());
}
}

function configMail(){
$config['smtp_host'] = 'smtp.gmail.com';
$config['smtp_port'] = '465';
$config['smtp_user'] = 'foneazmi0@gmail.com';
$config['smtp_auth'] = TRUE;
$config['smtp_pass'] = 'fone azmi';
$config['smtp_crypto'] = 'ssl';
$config['protocol'] = 'smtp';
$config['mailtype'] = 'html';
$config['send_multipart'] = FALSE;
$config['charset'] = 'utf-8';
$config['wordwrap'] = TRUE;
return $config;
}

public function ver()
{
$q=$this->input->get('q');
$q=json_decode(base64_decode(base64_decode($q)));
```

Tabel 5.4 Kode Program Register, Login, dan Logout (Lanjutan)

AuthController.php

```
$stat=$this->authM->ver($q);
if ($stat['stat']) {
$this->session->set_flashdata('message', 'Akun Berhasil Diverifikasi!');
redirect(base_url('login'));
}else{
$this->session->set_flashdata('message', 'Akun Gagal Diverifikasi!');
redirect(base_url('register'));
}
}

    public function login()
    {
$data=$this->input->post(NULL, TRUE);
if($this->session->role!=0 || $this->session->username==null){
if($data){
$data['password']=hash('sha512', $data['password']);
$result=$this->authM->login($data);
if ($result['stat']) {
$this->session->set_userdata([
"id"=>$result['data']->id,
"username"=>$result['data']->username,
"role"=>$result['role']
]);
$this->session->set_flashdata('message', 'Login Berhasil');
$referred_from = $this->session->userdata('referred_from');
if($referred_from != null){
redirect($referred_from, 'refresh');
}else{
redirect(base_url(), 'refresh');
}
}else{
$this->session->set_flashdata('message', 'Login Gagal');
redirect(base_url('login'), 'refresh');
}
}else{
$data['header']['title'] = "Login";
return $this->load->view('auth/login', $data);
redirect(base_url(), 'refresh');
}
}
```

Tabel 5.4 Kode Program Register, Login, dan Logout (Lanjutan)

AuthController.php

```
114  public function logout()
115  {
116  $this->session->sess_destroy(); // destroy all session
117  redirect(base_url('GuestController')); // redirect to home
118  }
119  }
```

Sumber: (Penulis, 2019)

Penjelasan kode program pada Tabel 5.4 adalah sebagai berikut:

1. Baris 17 - 80 merupakan baris kode yang berfungsi untuk menjankan fitur register. Untuk menjalankan fitur register terdapat empat method yang saling berkaitan yaitu method *register()*, method *sendMail (username, email, kode)*, method *configMail()* dan method *ver()*. Pada method register() terdapat pengecekan apakah register berhasil dilakukan atau tidak. Jika statusnya berhasil , maka akan menjalankan method *sendMail (username, email, kode)* yang didalamnya telah diinisialisasi method *configMail()* yang berfungsi untuk mengirimkan link verifikasi akun ke email. Kemudian jika user melakukan verifikasi pada link yang telah dikirimkan melalui email, maka akan dijalankan method *ver()* yang berfungsi untuk melakukan pengecekan status apakah akun berhasil diverifikasi atau gagal diverifikasi. Jika akun sudah diverifikasi user dapat melakukan *login* sedangkan jika akun gagal diverifikasi user akan diarahkan pada halaman register.
2. Baris 82 - 111 merupakan baris kode yang berfungsi untuk menjalankan fitur *login.* Untuk menjalankan fitur login terdapat method *login()*. Pada Method *login()* berfungsi untuk mengecek *username* dan *password* yang telah diinputkan user valid atau tidak dan mengecek status user sebagai wisatawan atau admin.
3. Baris 114 - 118 merupakan baris kode yang berfungsi untuk menjalankan fitur *logout.* Untuk menjalankan fitur *logout* terdapat method *logout()*. Pada method *logout()* berfungsi untuk menghapus semua *session authentication* user admin maupun wisatawan yang telah melakukan *login.*

Tabel 5.5 Kode Program Kegiatan

GuestController.php

```
1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  class GuestController extends CI_Controller
5  {
6  public function __construct()
7  {
8  parent::__construct();
9  $this->load->model('KegiatanModel', 'kegiatan');
10 }
11
12 public function kegiatan()
13 {
14 $data['header']['title'] = "Kegiatan";
15 $data['body'] = $this->kegiatan->getKegiatan();
16 return $this->load->view('user/kegiatan/Kegiatan', $data);
17 }
18 }
```

Sumber: (Penulis, 2019)

Penjelasan kode program pada Tabel 5.5 adalah sebagai berikut:

1. Baris 12 - 16 merupakan baris kode yang berfungsi untuk menjalankan fitur kegiatan. Untuk menjalankan fitur login terdapat method *kegiatan()*. Method *kegiatan()* berfungsi untuk menampilkan halaman kegiatan dengan memanggil method *getKegiatan()* yang terdapat pada Kegiatan Model.

Tabel 5.6 Kode Program Pemesanan Tiket

UserController.php

```
1  <?php
2  defined('BASEPATH') or exit('No direct script access allowed');
3  class UserController extends CI_Controller
5  public function __construct()
6  {
7  parent::__construct();
8  $this->load->model('TiketModel', 'tiket');
9  $this->load->helper('Khan');
10 date_default_timezone_set('Asia/Jakarta');
11 // if($this->session->role!=0 || $this->session->username==null){
12 // redirect(base_url('logout'), 'refresh');
13 // }
14 }
15
16 public function infoTiket()
17 {
18 $data['header']['title'] = "Info Tiket";
19 $input = $this->input->get(NULL, TRUE);
20 if ($input) {
21 $data['body'] = $this->tiket->getTiket($input);
22 $data['type'] = $input;
23 return $this->load->view('user/tiket/InfoTiket', $data);
24 } else {
25 return $this->load->view('user/tiket/Tiket', $data);
26 }
27 }
28 }
29
30 public function detailTiket($id)
31 {
32
33 $data = $this->input->post(NULL, TRUE);
34 if ($data) {
35 if($this->session->has_userdata('tanggal')){
36 $data['tgl_pesanan'] = $this->session->userdata('tanggal');
37 } else {
38 $data['tgl_pesanan'] = date('Y-m-d H:i:s');
39 } if (isset($this->session->id)) {
40 $this->tiket->insertPesanan($data);
41 redirect(base_url('pemesanan/') . $this->session->id, 'refresh');
42 } $this->session->set_userdata('referred_from',
base_url('pemesanan/'));
```

Tabel 5.6 Kode Program Pemesanan Tiket (Lanjutan)

UserController.php

```
45 $this->session->set_userdata('datainput', $data);
46 redirect(base_url('login'));
47 } else {
48 $data['header']['title'] = "Detail Tiket";
49 $data['body'] = $this->tiket->getTiketDetail(['id' => $id]);
50 $data['stokTiket'] = $this->tiket->getStok();
51 if($this->session->has_userdata('tanggal')){
$data['tgl_pesanan'] = $this->session->userdata('tanggal'); // u
52 reservasi
53 } else {
54 $data['tgl_pesanan'] = date('Y-m-d H:i:s'); //u go show
55 }
56 if (!isset($data['body'])) {
57 redirect(base_url('tiket'), 'refresh');
58 }
59 return $this->load->view('user/tiket/DetailTiket', $data);
60 }
61 }
62
63 public function cekStok($tgl)
64 {
65 $data = $this->tiket->getTotalPesan($tgl);
66 $this->session->set_userdata('tanggal', $tgl);
67 $stok = $this->tiket->getStok();
68 $stokHariIni = $stok - $data->total;
69 echo $stokHariIni;
70 return $stokHariIni;
71 }
72
73 public function pemesanan($id = NULL)
74 {
75 $dataInput = $this->session->datainput;
76 $dataInput['id_user'] = $this->session->id;
77 $this->session->set_userdata('datainput', $dataInput);
78 $this->tiket->insertPesanan($this->session->datainput);
79 $data['header']['title'] = "Pembayaran";
$data['body'] = $this->tiket->getDetailPemesanan($this->session-
80 >id);
81 $dataTiket = $this->tiket->getStok() - $data['body']->jumlah_tiket;
82 $this->tiket->updateStok($dataTiket);
83 if (!isset($data['body'])) {
84 redirect(base_url('tiket'), 'refresh');
```

Tabel 5.6 Kode Program Pemesanan Tiket (Lanjutan)

UserController.php

```
85  }
82  $this->session->unset_userdata('tanggal');
83  $data['order']=$this->tiket->getTiketone($data['body']->id_tiket);
84  $this->session->unset_userdata('tanggal');
85  // Set your Merchant Server Key
86  \Midtrans\Config::$serverKey = '';
87  \Midtrans\Config::$isProduction = false;
88  \Midtrans\Config::$isSanitized = true;
89  \Midtrans\Config::$is3ds = true;
90  $transaction_details = array(
91  'order_id' => $data['body']->id,
92  'gross_amount' => $data['body']->harga,
93  );
94
95  $item1_details = array(
96  'id' => $data['body']->id_tiket,
97  'price' => $data['body']->harga,
98  'quantity' => $data['body']->jumlah_tiket,
99  'name' => $data['order']->nama
100 );
101
102 $item_details = array ($item1_details);
103 $transaction = array(
104 'transaction_details' => $transaction_details,
105 'item_details' => $item_details,
106 );
107
108 $snapToken = \Midtrans\Snap::getSnapToken($transaction);
109 $data['snapToken']=$snapToken;
110 $this->load->view('test/test', $data);
111 }
112
113 function etiket(){
114         $idUser = $this->session->id;
115         $data['header']['title'] = "My Tickets";
116         $data['body'] = $this->tiket->getUserTiket($idUser);
117         return $this->load->view('user/dashboard/tiketUser', $data);
118     }
119     function cetakTiket(){
120         $data['header']['title'] = "Cetak Tiket";
121         $ticketData = array(
```

Tabel 5.6 Kode Program Pemesanan Tiket (Lanjutan)

UserController.php

```
122             'id' => $this->input->post('idTiket'),
123             'nama' => $this->input->post('namaTiket'),
124             'tanggal' => $this->input->post('tanggalTiket'),
125             'jumlah' => $this->input->post('jumlahTiket'),
126             'pemesan'          =>          strtoupper($this->input->post('pemesanTiket')),
127         );
128         $data['body'] = $ticketData;
129         //var_dump($data);
130         return $this->load->view('user/tiket/cetakTiket', $data);
131     }
132 }
```

Sumber: (Penulis, 2019)

Penjelasan kode program pada Tabel 5.6 adalah sebagai berikut:

1. Baris 16 – 28 merupakan baris kode yang berfungsi untuk menjankan fitur memilih kategori tiket dan mengetahui kegiatan dan fasilitas paket sesui kategori tiket. Untuk menjalankan fitur tersebut terdapat method *infoTiket().* Pada method *infoTiket()* berfungsi untuk melakukan pengecekan pada variabel input. Jika variabel input bernilai NULL, maka akan menampilkan pilihan kategori tiket. Jika variabel input bernilai TRUE, maka akan menampikan kegiatan dan fasilitas paket sesuai kategori tiket.
2. Baris 30 – 71 merupakan baris kode yang berfungsi untuk menjalankan fitur pembelian tiket. Untuk menjalankan fitur pembelian tiket terdapat method *detailTiket(id)* dan *cekStok(tgl)*. Pada method *detailTiket(id)* berfungsi untuk mengekseskusi pembelian tiket. Method *cekStok(tgl)* dijalankan ketika wisatawan memilih kategori tiket reservasi untuk mengetahui ketersediaan tiket maka wisatawan harus melakukan cek tanggal terlebih dahulu.
3. Baris 85 – 93 merupakan baris kode yang berfungsi untuk mengkonfigurasikan *servey key* agar dapat mengambil data dari midtrans
4. Baris 73 – 112 merupakan baris kode yang berfungsi untuk menjalankan fitur pembayaran tiket. Untuk menjalankan fitur pembayaran tiket terdapat

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

method pemesanan(id = NULL). Method pemesanan(id = NULL) berfungsi untuk melakukan pembayaran melalui midtrans.

5. Baris 113 – 118 merupakan baris kode yang berfungsi untuk menjalankan fitur melihat status pembayaran tiket pada menu tiket. Untuk menjalankan fitur tersebut terdapat method etiket().
6. Baris 119 – 131 merupakan baris kode yang berfungsi untuk menjalankan fitur mencetak laporan. Untuk menjalankan fitur pembayaran tiket terdapat method cetakTiket()

Tabel 5.7 Kode Program Pada Halaman Admin

AdminController.php

```
1  <?php
2  defined('BASEPATH') or exit('No direct script access allowed');
3
4  use PhpOffice\PhpSpreadsheet\Spreadsheet;
5  use PhpOffice\PhpSpreadsheet\Writer\Xlsx;
6
7  class AdminController extends CI_Controller
8  {
9  public function __construct()
10 {
11 parent::__construct();
12 $this->load->model('KegiatanModel', 'kegiatan');
13 $this->load->model('TiketModel', 'tiket');
14 $this->load->model('PenggunaModel', 'pengguna');
15 $this->load->helper('Khan');
16 if ($this->session->role != 1 || $this->session->username == null)
17 {
18 redirect(base_url('logout'), 'refresh');
19 }
20 }
21 public function index()
22 {
23 $tahun = (isset($_GET['thn']) && $_GET['thn'] != "") ?
24 $_GET['thn'] : date("Y"); //default tahun saat ini
25 $data['header']['title'] = "Pendapatan Penjualan Tiket";
26 $data['body']['tahun'] = [];
27 for ($i=1; $i <= 12; $i++) { // perulangan bulan 1-12
28 $data['body']['tahun'][$i][] = $this->tiket->laporanTahun($i,
29 $tahun); // isi data pemesanan di tanggal i
30 $data['body']['thn'][$i][] = date("F", mktime(0, 0, 0, $i, 10));
31 }
```

Tabel 5.7 Kode Program Pada Halaman Admin (Lanjutan)

AdminController.php

```
30  return $this->load->view('admin/dashboard/Dashboard', $data);
31  }
32
33  //tiket
34  public function tiket()
35  {
36  $data['header']['title'] = "Tiket";
37  $data['body'] = $this->tiket->getTiket();
38  return $this->load->view('admin/tiket/Tiket', $data);
39  }
40
41  public function dellTiket($id)
42  {
43  $this->tiket->dellTiket($id);
44  redirect(base_url('admin/tiket'), 'refresh');
45  }
46
47  public function editTiket($id)
48  {
49  $data = $this->input->post(NULL, TRUE);
50  if ($data) {
51  $this->tiket->editTiket($id, $data);
52  redirect(base_url('admin/tiket'), 'refresh');
53  } else {
54  redirect(base_url(), 'refresh');
55  }
56  }
57
58  public function addTiket()
59  {
60  $data = $this->input->post(NULL, TRUE);
61  if ($data) {
62  $text = pathinfo($_FILES['foto']['name'], PATHINFO_EXTENSION);
63  $config['file_name'] = time() . ".$ext";
64  $config['upload_path'] = './assets/img/tiket/';
65  $config['allowed_types'] = 'gif|jpg|png';
66  $this->upload->initialize($config);
67  $this->load->library('upload', $config);
68  if ($this->upload->do_upload('foto')) {
69  $data = array_merge($data, array("img" => $config['file_name']));
70  $data = $this->tiket->addTiket($data);
71  redirect(base_url('admin/tiket'), 'refresh');
```

Tabel 5.7 Kode Program Pada Halaman Admin (Lanjutan)

AdminController.php

```
72  } else {
73  $error = array('message' => $this->upload->display_errors());
74  var_dump($error);
75  }
76  } else {
77  redirect(base_url(), 'refresh');
78  }
79  }
80
81  //kegiatan
82  public function kegiatan()
83  {
84  $data['header']['title'] = "Kegiatan";
85  $data['body'] = $this->kegiatan->getKegiatan();
86  return $this->load->view('admin/kegiatan/Kegiatan', $data);
87  }
88
89  public function deleteKegiatan($id)
90  {
91  $this->kegiatan->dellKegiatan($id);
92  redirect(base_url('admin/kegiatan'), 'refresh');
93  }
94
95  public function editKegiatan($id)
96  {
97  $data = $this->input->post(NULL, TRUE);
98  if ($data) {
99  $this->kegiatan->editKegiatan($id, $data);
100 redirect(base_url('admin/kegiatan'), 'refresh');
101 } else {
102 redirect(base_url(), 'refresh');
103 }
104 }
105
106 public function tambahKegiatan()
107 {
108 $data = $this->input->post(NULL, TRUE);
109 if ($data) {
110 $ext = pathinfo($_FILES['foto']['name'], PATHINFO_EXTENSION);
111 $config['file_name'] = time() . ".$ext";
112 $config['upload_path'] = './assets/img/kegiatan/';
113 $config['allowed_types'] = 'gif|jpg|png';
```

Tabel 5.7 Kode Program Pada Halaman Admin (Lanjutan)

AdminController.php

```
114 $this->upload->initialize($config);
115 $this->load->library('upload', $config);
116 if ($this->upload->do_upload('foto')) {
117 $data = array_merge($data, array("foto" => $config['file_name']));
118 $data = $this->kegiatan->addKegiatan($data);
119 redirect(base_url('admin/kegiatan'), 'refresh');
120 } else {
121 $error = array('message' => $this->upload->display_errors());
122 var_dump($error);
123 }
124 } else {
125 redirect(base_url(), 'refresh');
126 }
127 }
128
129 //Pengguna
130 public function pengguna()
131 {
132 $data['header']['title'] = "Pengguna";
133 $data['body'] = $this->pengguna->getPengguna();
134 return $this->load->view('admin/pengguna/Pengguna', $data);
135 }
136
137 public function deletePengguna($id)
138 {
139 $this->pengguna->dellPengguna($id);
140 redirect(base_url('admin/pengguna'), 'refresh');
141 }
142
143 //Pemesanan
144 public function pemesanan()
145 {
146 $data['header']['title'] = "Pemesanan";
147 $data['body'] = $this->tiket->getPemesanan();
148
149 \Midtrans\Config::$serverKey = '';
150 \Midtrans\Config::$isProduction = false;
151 \Midtrans\Config::$isSanitized = true;
152 \Midtrans\Config::$is3ds = true;
153
154 for ($i = 0; $i < count($data['body']); $i++){
155 $status = \Midtrans\Transaction::status($data['body'][$i]->id);
```

Tabel 5.7 Kode Program Pada Halaman Admin (Lanjutan)

AdminController.php

```
156 $data['status'][] = array(
157 'id' => $data['body'][$i]->id,
158 'status' => $status->transaction_status,
159 'type' => $status->payment_type
160 );
161 }
162 return $this->load->view('admin/pemesanan/Pemesanan', $data);
163 }
164
165 //Cetak Laporan Pada Excel
166 public function cetaklaporan($thn)
167 {
168 $data = $this->tiket->testLaporanTahun($thn);
169 $spreadsheet = new Spreadsheet();
170 $sheet = $spreadsheet->getActiveSheet()->setTitle("Tahun");
171 $sheet->setCellValue('A1', 'No');
172 $sheet->setCellValue('B1', 'Bulan');
173 $sheet->setCellValue('C1', 'Tipe Paket');
174 $sheet->setCellValue('D1', 'Nama Paket');
175 $sheet->setCellValue('E1', 'Total Tiket');
176 $sheet->setCellValue('F1', 'Harga Total Tiket');
177 $no = 2;
178 $jumlahHargaTotal = 0;
179 foreach($data as $dt){
180 $sheet->setCellValue('A'.$no, $no-1);
181 $sheet->setCellValue('B'.$no, date("F", mktime(0, 0, 0, $dt['bulanPemesanan'], 10)));
182 $sheet->setCellValue('C'.$no, $dt['nama']);
183 $sheet->setCellValue('D'.$no, $dt['namaPaket']);
184 $sheet->setCellValue('E'.$no, $dt['total']);
185 $sheet->setCellValue('F'.$no, rupiah($dt['totalHarga']));
186 $jumlahHargaTotal = $jumlahHargaTotal + $dt['totalHarga'];
187 $no++;
188 }
189
190 $sheet->setCellValue('E'.($no+2), "Jumlah");
191 $sheet->setCellValue('F'.($no+2), rupiah($jumlahHargaTotal));
192 for($i = 1; $i <=12; $i++){
193 $sheet = $spreadsheet->createSheet()->setTitle(date("F", mktime(0, 0, 0, $i, 10)));
194 $data = $this->tiket->laporanTahun($i, $thn);
195 $sheet->setCellValue('A1', 'No');
196 $sheet->setCellValue('B1', 'Tanggal Pemesanan');
```

Tabel 5.7 Kode Program Pada Halaman Admin (Lanjutan)

AdminController.php

```
197 $sheet->setCellValue('C1', 'Tipe Paket');
198 $sheet->setCellValue('D1', 'Nama Paket');
199 $sheet->setCellValue('E1', 'Total Tiket');
200 $sheet->setCellValue('F1', 'Harga Total Tiket');
201 $no = 2;
202 $jumlahHargaTotal = 0;
203 foreach($data as $dt){
204 $sheet->setCellValue('A'.$no, $no-1);
205 $sheet->setCellValue('B'.$no, $dt['tanggalPemesanan']);
206 $sheet->setCellValue('C'.$no, $dt['nama']);
207 $sheet->setCellValue('D'.$no, $dt['namaPaket']);
208 $sheet->setCellValue('E'.$no, $dt['total']);
209 $sheet->setCellValue('F'.$no, rupiah($dt['totalHarga']));
210 $jumlahHargaTotal = $jumlahHargaTotal + $dt['totalHarga'];
211 $no++;
212 }
213 }
214 $writer = new Xlsx($spreadsheet);
215 $filename = 'Laporan Tahun '.$thn;
216 header('Content-Type: application/vnd.ms-excel');
217 header('Content-Disposition: attachment;filename="' . $filename . '.xlsx"');
218 header('Cache-Control: max-age=0');
219 $writer->save('php://output');
220 }
221
222 public function lapkeuangan()
223 {
224 $tahun = (isset($_GET['thn']) && $_GET['thn'] != "") ? $_GET['thn'] : date("Y"); //default tahun saat ini
225 $bulan = (isset($_GET['bln']) && $_GET['bln'] != "") ? $_GET['bln'] : date("m"); //default bulan saat ini
226 $data['bulan'] = date("F", mktime(0, 0, 0, $bulan, 10));
227 $data['bln'] = $bulan;
228 $data['thn'] = $tahun;
229 $data['header']['title'] = "Laporan Keuangan";
230 $data['body']['bulan'] = [];
231 $data['body']['tgl'] = [];
232 $data['body']['tahun'] = [];
233 for ($i=1; $i <= 31; $i++) { // perulangan tanggal 1-31
234 $tgl = $tahun . "-$bulan-$i"; // bikin format tanggal untuk jadi argument ke model
235 $data['body']['bulan'][$i][] = $this->tiket->laporanBulan($tgl); // isi data pemesanan di tanggal i
236 $data['body']['tgl'][$i][] = date("Y") . "-$bulan-$i";
```

Tabel 5.7 Kode Program Pada Halaman Admin (Lanjutan)

AdminController.php	
237	}
238	for ($i=1; $i <= 12; $i++) { // perulangan bulan 1-12
239	$data['body']['tahun'][$i][] = $this->tiket->laporanTahun($i, $tahun); // isi data pemesanan di tanggal i
240	$data['body']['thn'][$i][] = date("F", mktime(0, 0, 0, $i, 10));
241	}
242	return $this->load->view('admin/lapkeuangan/LapKeuangan', $data);
243	}
244	}

Sumber: (Penulis, 2019)

Penjelasan kode program pada Tabel 5.7 adalah sebagai berikut:

1. Baris 21 – 31 merupakan baris kode yang berfungsi untuk menjalankan fitur melihat grafik pendapatan penjualan tiket.
2. Baris 34 – 79 merupakan baris kode yang berfungsi untuk menjalankan fitur *update* pada tiket. Untuk menjalankan fitur mengelola data pada halaman tiket terdapat method *tiket(), dellTiket(id), editTiket(id),* dan *addTiket()*. Method *tiket()* berfungsi untuk menampilkan data tiket. Method *dellTiket(id)* berfungsi untuk menghapus data tiket. Method *editTiket(id)* berfungsi untuk mengubah data tiket. Dan Method *addTiket()* berfungsi untuk menambah data tiket.
3. Baris 82 – 127 merupakan baris kode yang berfungsi untuk menjalankan fitur mengelola data pada kegiatan. Untuk menjalankan fitur mengelola data pada kegiatan terdapat method *kegiatan(), deleteKegiatan(id), editKegiatan(id),* dan *tambahKegiatan()*. Method *kegiatan()* berfungsi untuk menampilkan data kegiatan. Method *deleteKegiatan(id)* berfungsi untuk menghapus data kegiatan. Method *editKegiatan()* berfungsi untuk mengubah data kegiatan. Method *tambahKegiatan()* berfungsi untuk menambah data kegiatan.
4. Baris 130 – 141 merupakan baris kode yang berfungsi untuk menjalankan fitur *update* pada pengguna. Untuk menjalankan fitur update pada pengguna terdapat method *pengguna()* dan method *deletePengguna(id).*

Method *pengguna()* berfungsi untuk menampilkan data pengguna. Method *deletePengguna(id)* berfungsi untuk menghapus data pengguna.

5. Baris 149 – 152 merupakan baris kode yang berfungsi untuk mengkonfigurasikan *servey key* agar dapat mengambil data dari midtrans.
6. Baris 144 – 163 merupakan baris kode yang berfungi untuk menjalankan fitur melihat daftar pemesanan. Untuk menjalankan fitur melihat daftar pemesanan terdapat method *pemesanan()*.
7. Baris 166 – 220 merupakan baris kode yang berfungsi untuk menjalankan fitur menu cetak laporan keuangan. Untuk menjalankan fitur cetak laporan terdapat method *cetaklaporan(thn)* yang berfungsi untuk menguduh laporan keuangn berdasarkan tahun.
8. Baris 222 – 243 merupakan baris kode yang berfungsi untuk menjalankan fungsi melihat laporan keuangan. Untuk menjalankan fitur cetak laporan terdapat method *lapkeuangan()*. Method *lapkeuangan()* berfungsi untuk menampilkan data laporan keuangan berdasarkan bulan dan tahun.

Tabel 5.8 Kode Program Cek Status Pembayaran Tiket

CornController.php

```
1   <?php
2   defined('BASEPATH') or exit('No direct script access allowed');
3
4   class cron extends CI_Controller
5   {
6   public function __construct()
7   {
8   parent::__construct();
9   $this->load->model('TiketModel', 'tiket');
10  $this->load->helper('Khan');
11  date_default_timezone_set('Asia/Jakarta');
12  }
13
14  public function cekBayar()
15  {
16  $data['header']['title'] = "Pemesanan";
17  $data['body'] = $this->tiket->getPemesanan();
18
19  \Midtrans\Config::$serverKey = '';
```

Tabel 5.8 Kode Program Cek Status Pembayaran Tiket (Lanjutan)

CornController.php	
20	`\Midtrans\Config::$isProduction = false;`
21	`\Midtrans\Config::$isSanitized = true;`
22	`\Midtrans\Config::$is3ds = true;`
23	
24	`for ($i = 0; $i < count($data['body']); $i++){`
25	`$status = \Midtrans\Transaction::status($data['body'][$i]->id);`
26	`$paymentId = $data['body'][$i]->id;`
27	`$paymentStatus = $status->transaction_status;`
28	`$this->tiket->setStatusBayar($paymentId, $paymentStatus);`
29	`}`
30	`}`
31	`}`

Sumber: (Penulis, 2019)

Penjelasan kode program pada Tabel 5.8 adalah sebagai berikut:

1. Baris 14 – 29 merupakan baris kode yang berfungsi untuk mengecek status pembayaran pemesanan tiket.

5.5 Implementasi Antarmuka

Implementasi antarmuka ini menjelaskan tentang implementasi pengguna sistem dalam berinteraksi dengan penggunanya. Berikut ini adalah implementasi antarmuka dari aplikasi *e-commerce* Wisata Kampung Sapi *Adventure*:

5.5.1 Implementasi Antarmuka Halaman Kegiatan

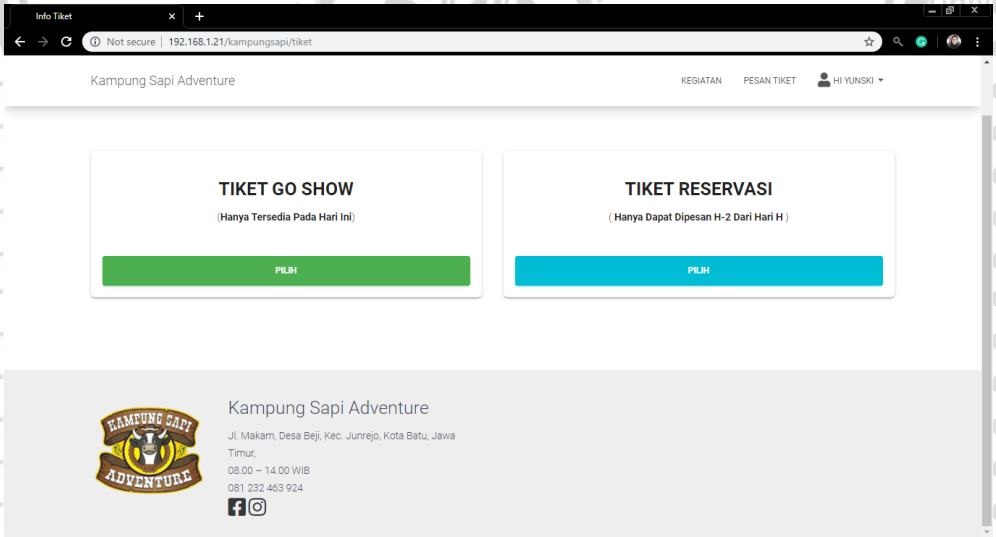
Pada Gambar 5.4 merupakan implementasi antarmuka halaman kegiatan terdapat beberapa nama kegiatan yang dilengkapi deskripsi dan gambar sesuai nama kegiatan. Implementasi antarmuka ini terdapat pada *view* Kegiatan.php.

Gambar 5.4 Implementasi Antarmuka Halaman Kegiatan Aplikasi *E-commerce* Wisata Kampung Sapi *Adventure*

Sumber: Penulis (2019)

5.5.2 Implementasi Antarmuka Halaman Pesan Tiket

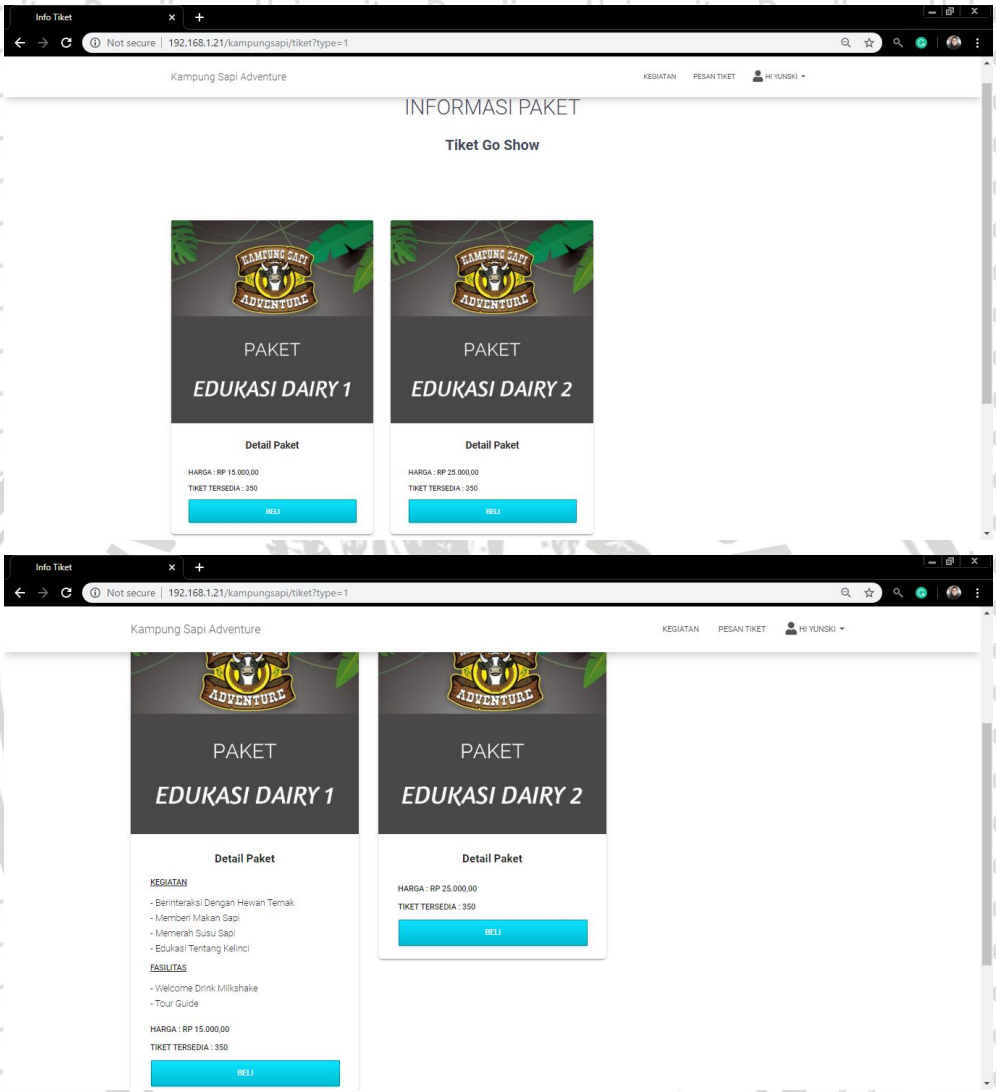
Pada implementasi antarmuka pesan tiket terdapat beberapa halaman antarmuka di dalamnya, yaitu halaman antamuka memilih kategori tiket, halaman antarmuka informasi paket tiket, halaman antarmuka form pemesanan tiket, halaman antarmuka detail tiket, dan halaman antarmuka pembayaran melalui midtrans. Pada Gambar 5.5 merupakan implementasi antarmuka memilih kategori tiket. Pada halaman ini wisatawan dapat melakukan pemesanan tiket dengan memilih kategori tiket. Implementasi antarmuka ini terdapat pada *view* Tiket.php.



Gambar 5.5 Implementasi Antamuka Halaman Pesan Tiket Aplikasi *E-commerce* Wisata Kampung Sapi *Adventure*

Sumber: (Penulis, 2019)

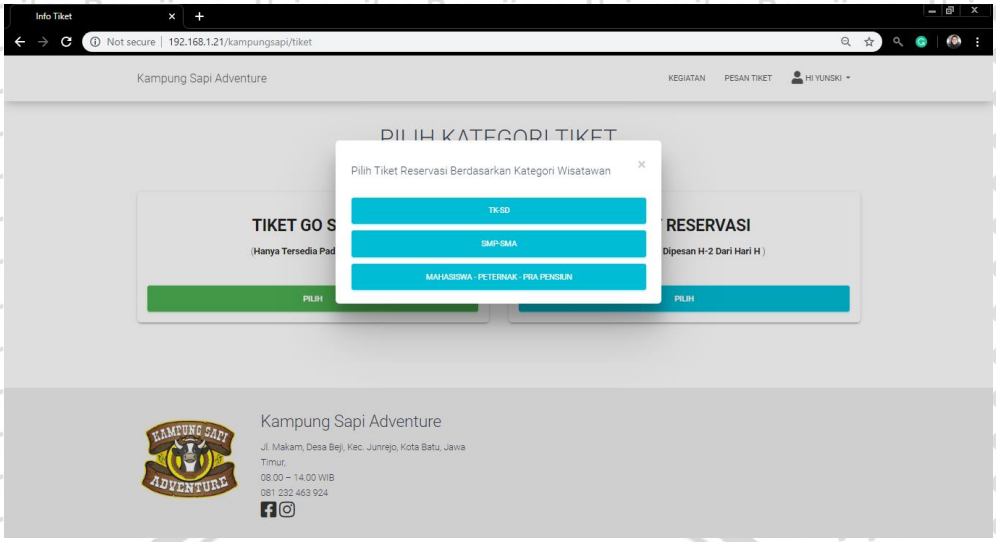
Pada Gambar 5.6 merupakan implementasi antarmuka halaman informasi paket pada kategori tiket go show. Pada halaman ini wisatawan dapat mengetahui kegiatan dan fasilitas paket, harga tiket, tiket tersedia, dan melakukan pembelian tiket. Implementasi antarmuka ini terdapat pada *view* InfoTiket.php.



Gambar 5.6 Implementasi Antarmuka Halaman Informasi Paket Tiket Go Show

Sumber: (Penulis, 2019)

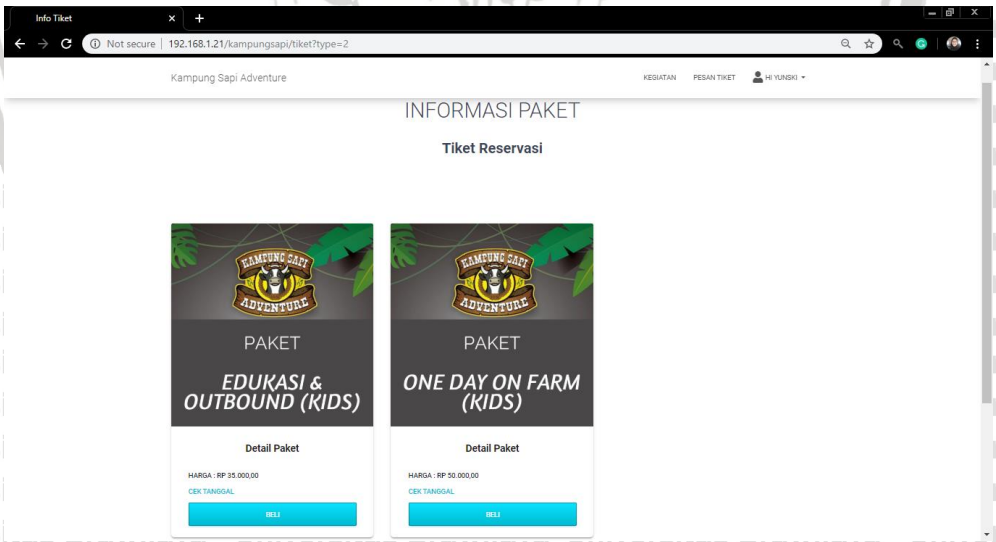
Pada Gambar 5.7 merupakan implementasi antarmuka halaman memilih kategori tiket reservasi memiliki penjelasan yang sama dengan penjelasan perancangan halaman tiket reservasi pada perancangan antarmuka. Implementasi antarmuka ini terdapat pada *view* Tiket.php.

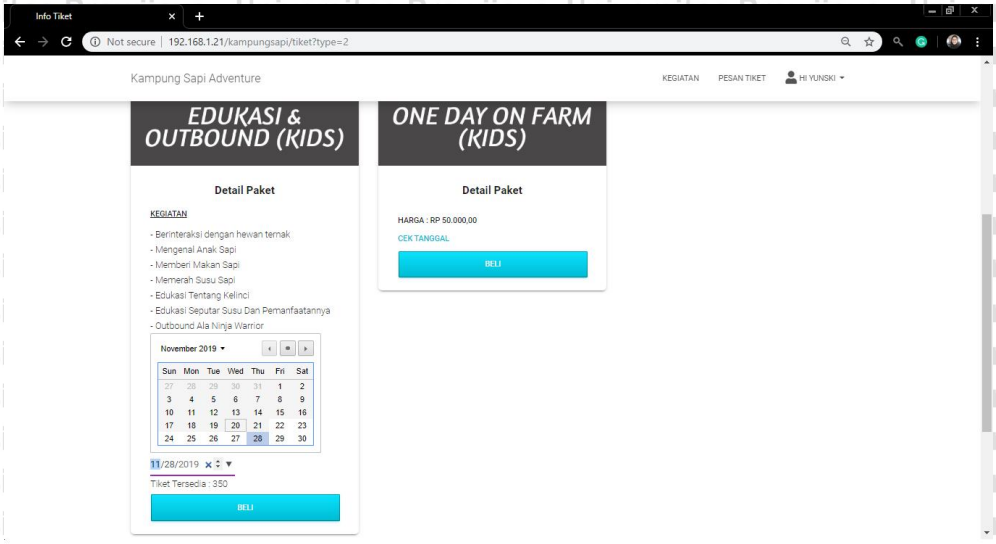


Gambar 5.7 Implementasi Antarmuka Halaman Kategori Tiket Reservasi

Sumber: (Penulis, 2019)

Pada Gambar 5.8 merupakan implementasi antarmuka halaman informasi paket tiket reservasi wisatawan dapat mengetahui fasilitas dan kegiatan, harga tiket, tiket tersedia, dan melakukan pembelian. Pada halaman ini wisatawan untuk melakukan pembelian diharuskan untuk melihat tiket yang tersedia dengan melakukan pengecekan tanggal terlebih dahulu. Implementasi antarmuka ini terdapat pada *view* InfoTiket.php.

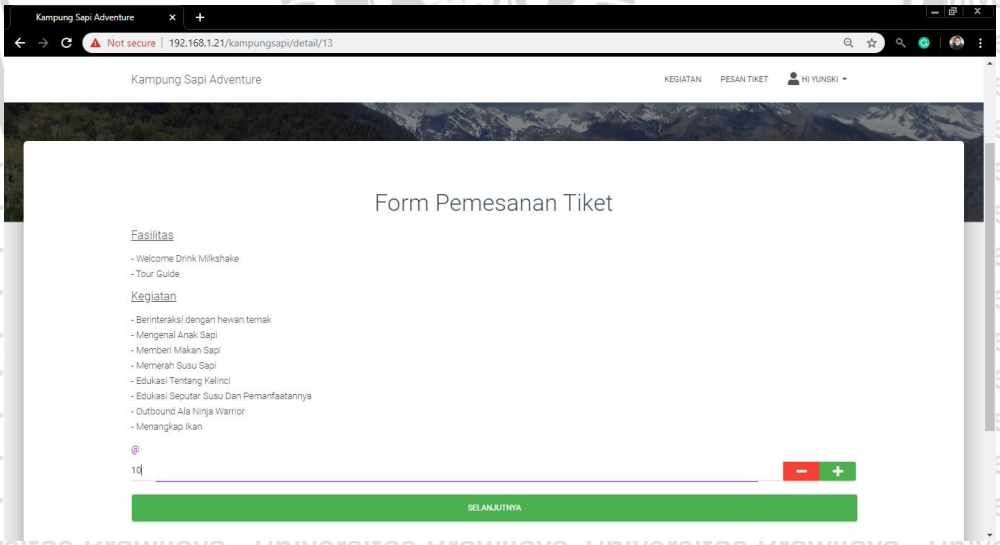




Gambar 5.8 Implementasi Antarmuka Informasi Paket Tiket Reservasi

Sumber: (Penulis, 2019)

Pada Gambar 5.9 merupakan implementasi antarmuka halaman form pemesanan tiket. Pada halaman form pemesanan tiket, wisatawan diminta untuk menginputkan jumlah tiket yang akan dibeli. Implementasi antarmuka ini terdapat pada *view* DetailTiket.php.

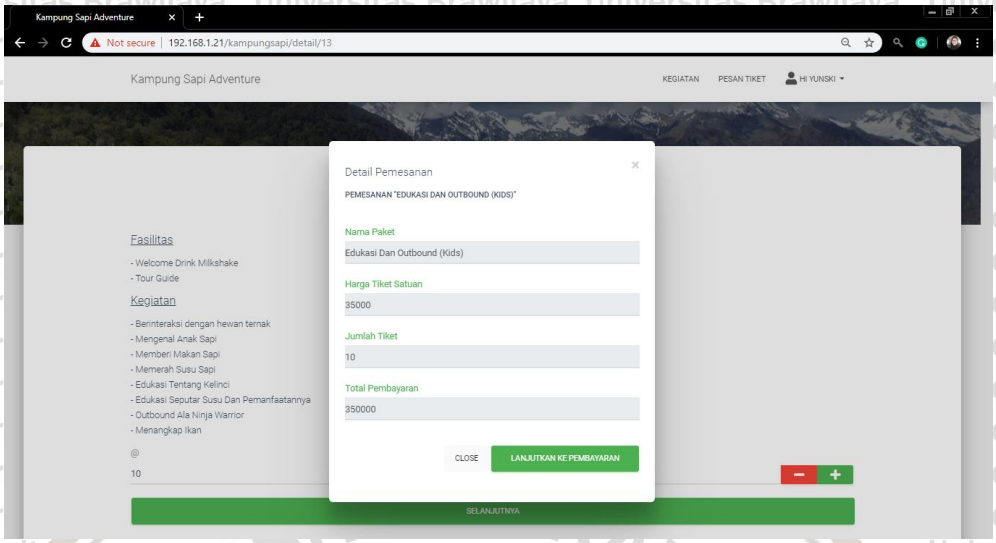


Gambar 5.9 Implementasi Antarmuka Halaman Form Pemesanan Tiket

Sumber: (Penulis, 2019)

Pada Gambar 5.10 merupakan implementasi antarmuka halaman detail tiket. Pada halaman ini wisatawan dapat mengetahui detail informasi pemesanan tiket. Jika pemesanan telah sesuai, maka wisatawan dapat menekan tombol

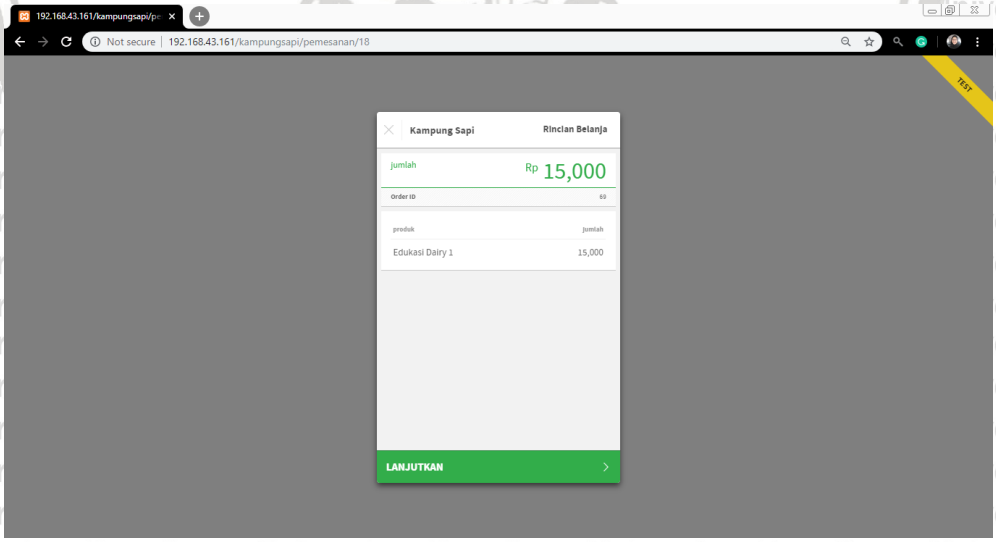
"Lanjutkan Ke Pembayaran". Implementasi antarmuka ini terdapat pada *view* DetailTiket.php.



Gambar 5.10 Implementasi Antarmuka Halaman Detail Tiket

Sumber: (Penulis, 2019)

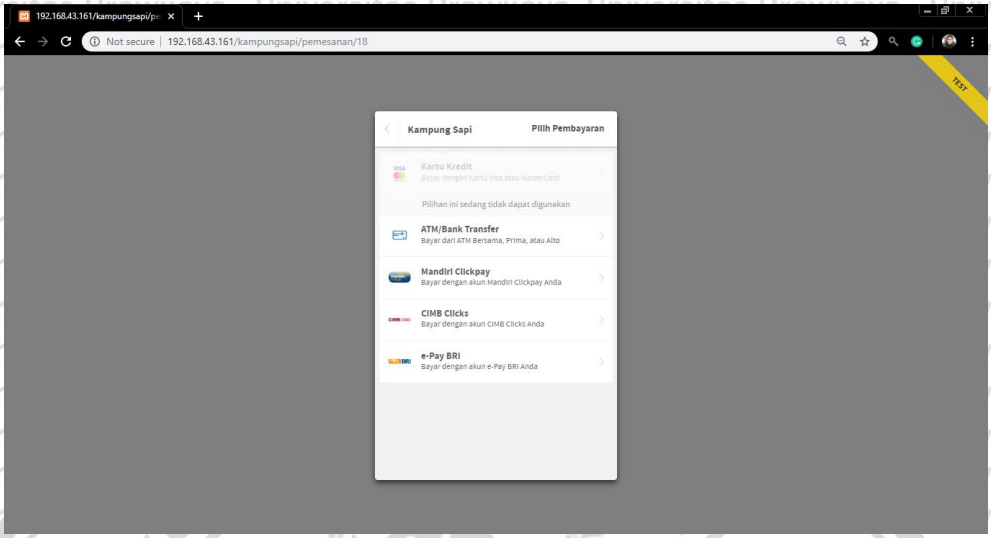
Pada Gambar 5.11 merupakan implementasi antarmuka halaman pembayaran pada midtrans. Pada halaman ini wisatawan dapat melihat rincian belanja pembelian tiket. Implementasi antarmuka ini terdapat pada *view* pemesanan.php.



Gambar 5.11 Implementasi Antarmuka Halaman Pembayaran Pada Midtrans

Sumber: (Penulis, 2019)

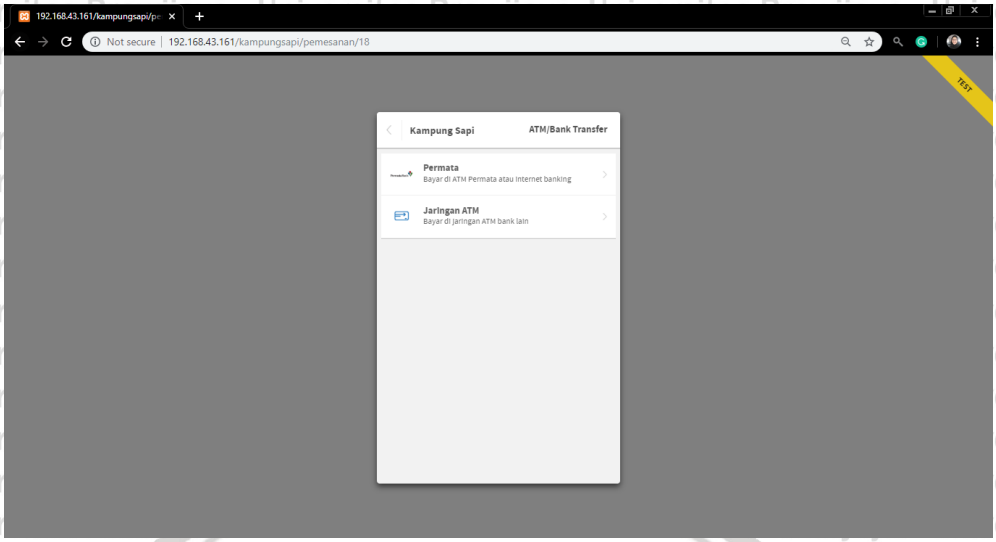
Pada Gambar 5.12 merupakan implementasi antarmuka halaman memilih metode pembayaran pada midtrans. Pada halaman ini wisatawan dapat memilih metode pembayaran. Implementasi antarmuka ini terdapat pada *view* pemesanan.php.



Gambar 5.12 Implementasi Antarmuka Halaman Memilih Metode Pembayaran Pada Midtrans

Sumber: (Penulis, 2019)

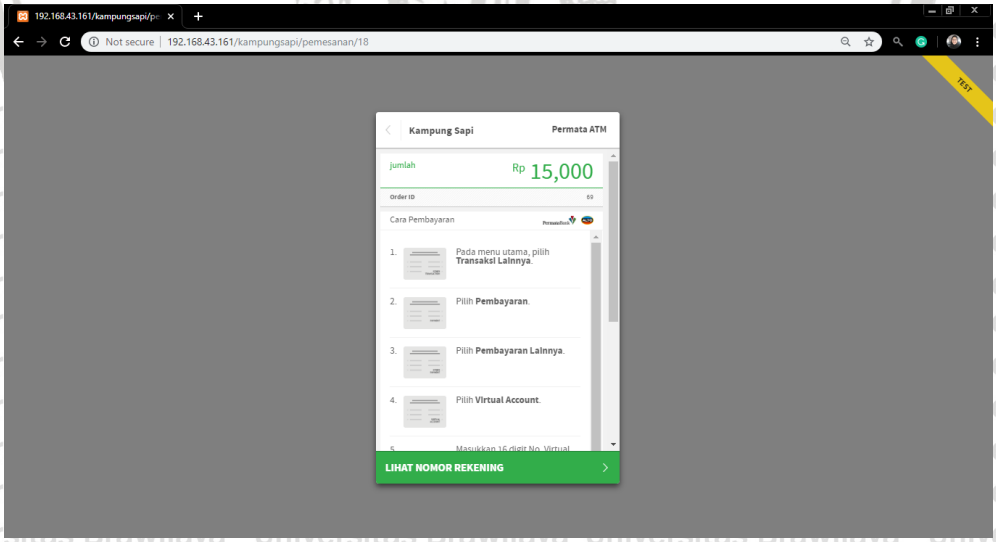
Pada Gambar 5.13 merupakan implementasi antarmuka halaman metode pembayaran menggunakan ATM/Bank Transfer pada midtrans. Pada halaman tersebut wisatawan dapat melakukan pembayaran melalui Permata atau Jaringan ATM lain. Implementasi antarmuka ini terdapat pada *view* pemesanan.php.

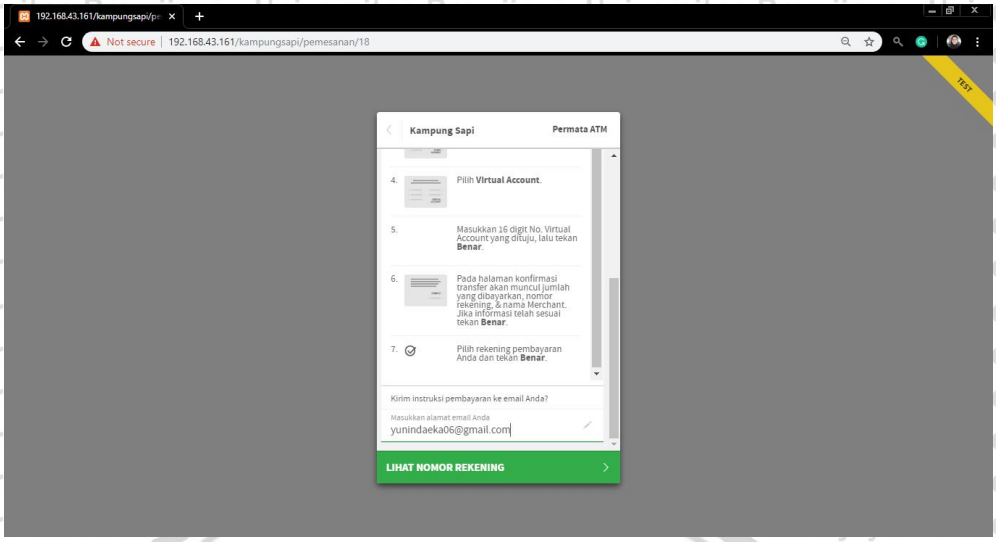


Gambar 5.13 Implementasi Antarmuka Halaman Metode Pembayaran Menggunakan ATM/Bank Transfer Pada Midtrans

Sumber: (Penulis, 2019)

Pada Gambar 5.14 merupakan implementasi antarmuka halaman cara pembayaran melalui Permatan pada midtrans. Pada halaman ini wisatawan dapat mengetahui cara pembayaran menggunakan Permata. Wisatawan dapat menerima instruksi pembayaran melalui email dengan menginputkan email pada *text field* yang tertera pada halaman ini. Implementasi antarmuka ini terdapat pada *view* pemesanan.php.

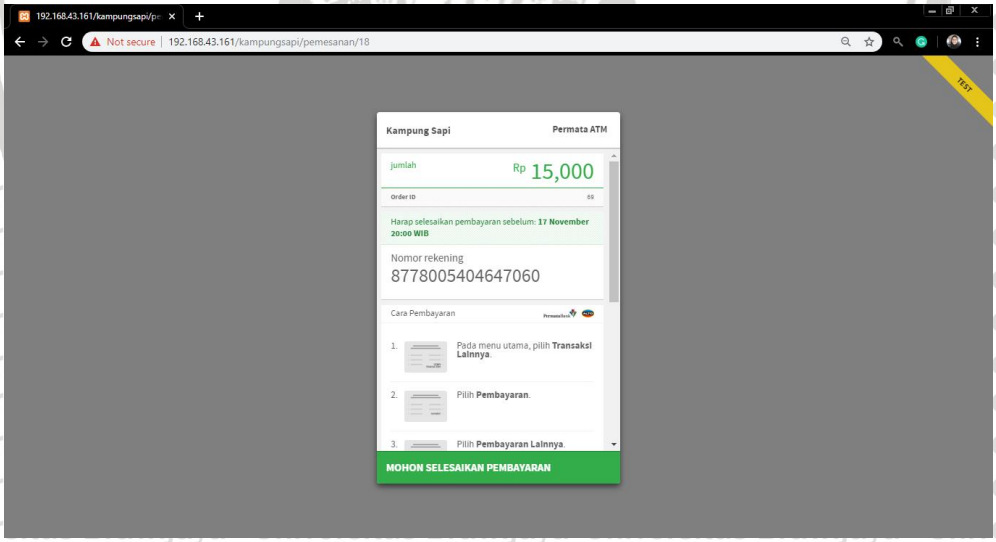




Gambar 5.14 Implementasi Antarmuka Halaman Cara Pembayaran Melalui Permata Pada Midtrans

Sumber: (Penulis, 2019)

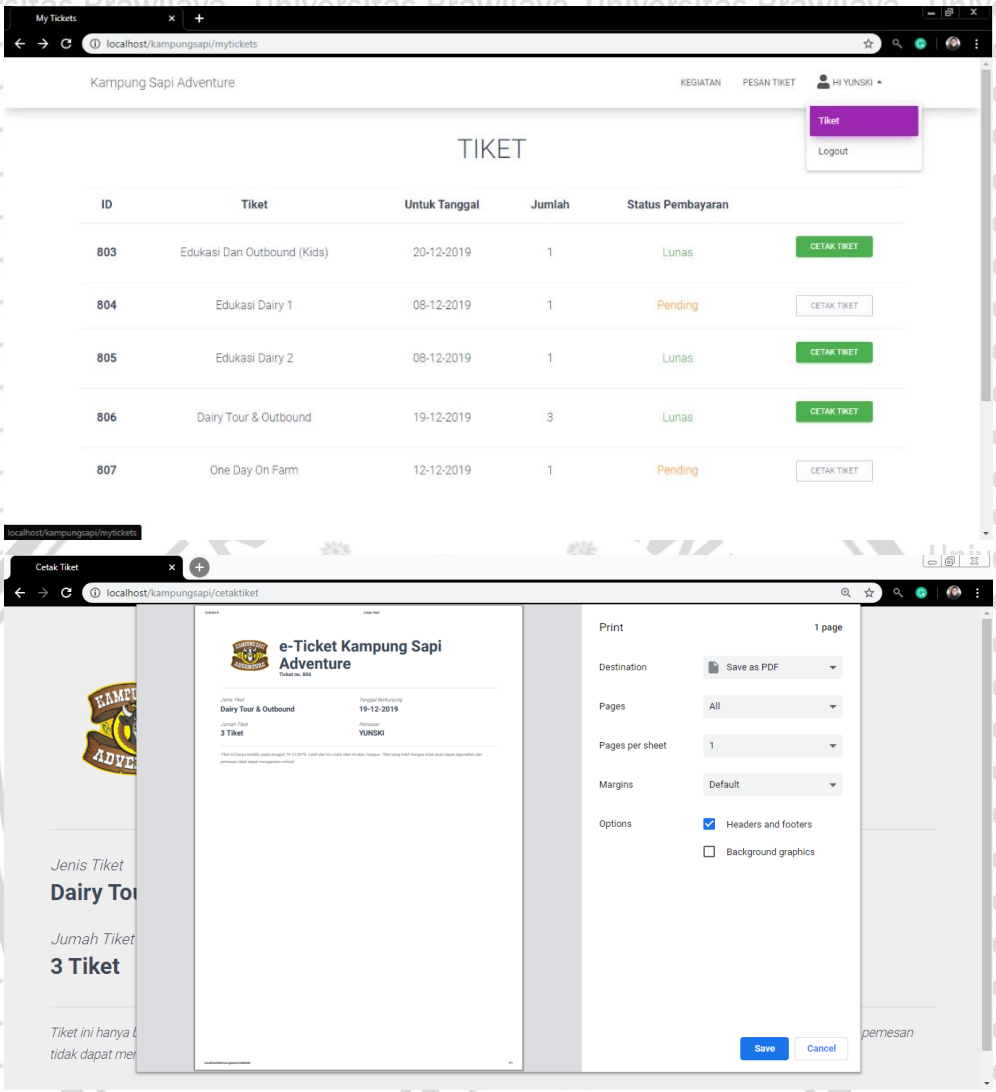
Pada Gambar 5.15 merupakan implementasi antarmuka halaman metode pembayaran melalui Permata Bank. Halaman ini berisi informasi nomor rekening yang nantinya digunakan wisatawan untuk melakukan pembayaran. Implementasi antarmuka ini terdapat pada *view* pemesanan.php.



Gambar 5.15 Implementasi Antarmuka Halaman Metode Pembayaran Melalui Permata Bank

Sumber: (Penulis, 2019)

Pada Gambar 5.16 merupakan implementasi antarmuka halaman pada menu tiket. Halaman ini berisi informasi mengenai . Implementasi antarmuka ini terdapat pada *view* pemesanan.php.

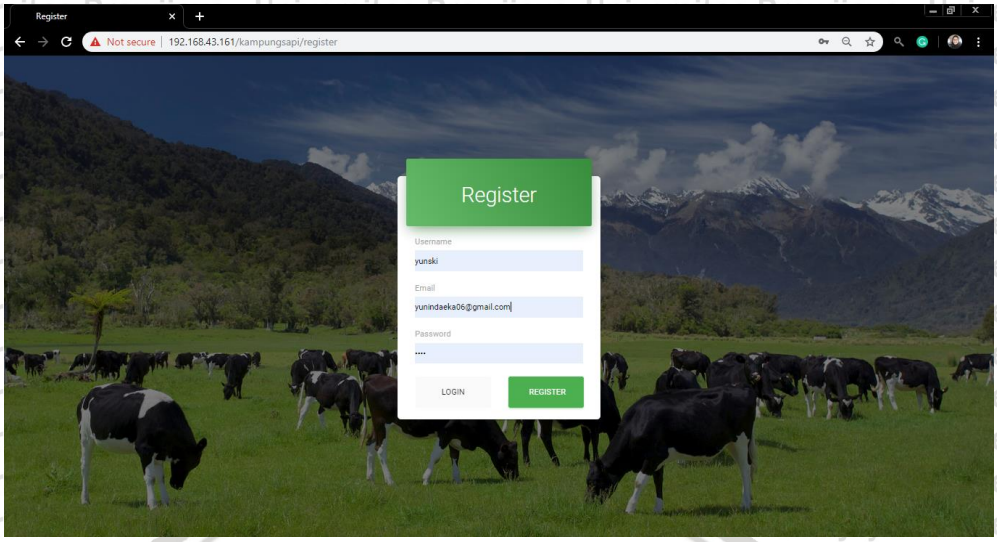


Gambar 5.16 Implementasi Antarmuka Halaman Pada Menu Tiket

Sumber: (Penulis, 2019)

5.5.3 Implementasi Antarmuka Halaman Register

Pada Gambar 5.17 merupakan implementasi halaman register. Pada halaman register wisatawan diminta untuk menginputkan *username, password,* dan email. Implementasi antarmuka ini terdapat pada *view* register.php.

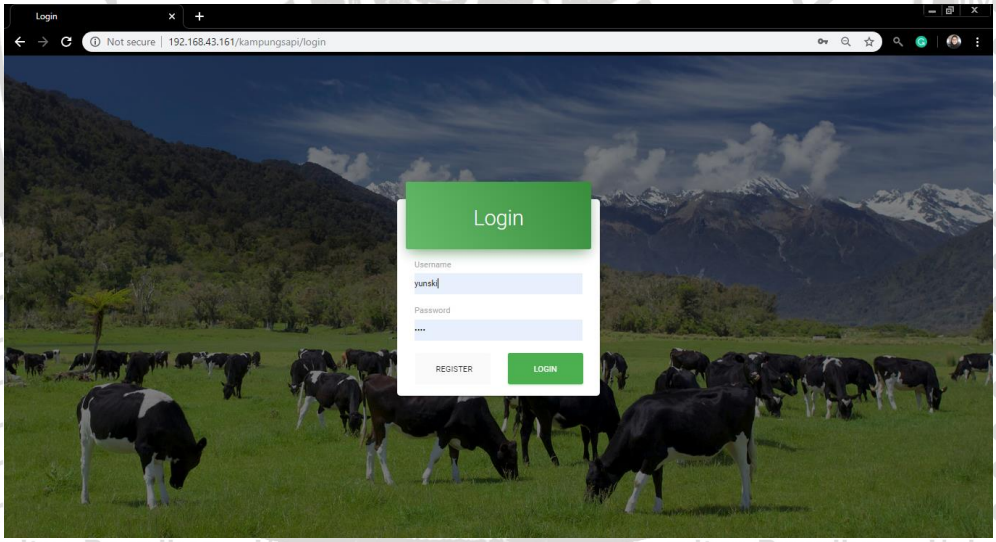


Gambar 5.17 Implementasi Antarmuka Halaman Register

Sumber: (Penulis, 2019)

5.5.4 Implementasi Antarmuka Halaman *Login*

Pada Gambar 5.18 merupakan implementasi halaman *login*. Pada halaman *login* wisatawan dan admin diminta untuk menginputkan *username* dan *password*. Implementasi antarmuka ini terdapat pada *view login*.php.

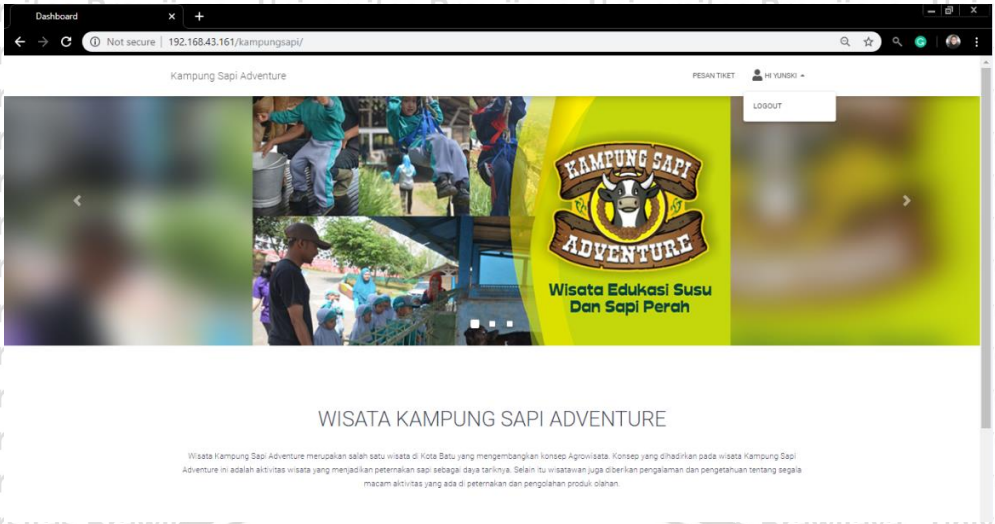


Gambar 5.18 Implementasi Antarmuka Halaman *Login*

Sumber: (Penulis, 2019)

5.5.5 Implementasi Antarmuka Halaman *Logout*

Pada Gambar 5.19 merupakan implementasi halaman *logout*. Pada halaman *login* wisatawan dan admin dapat melakukan *logout*.

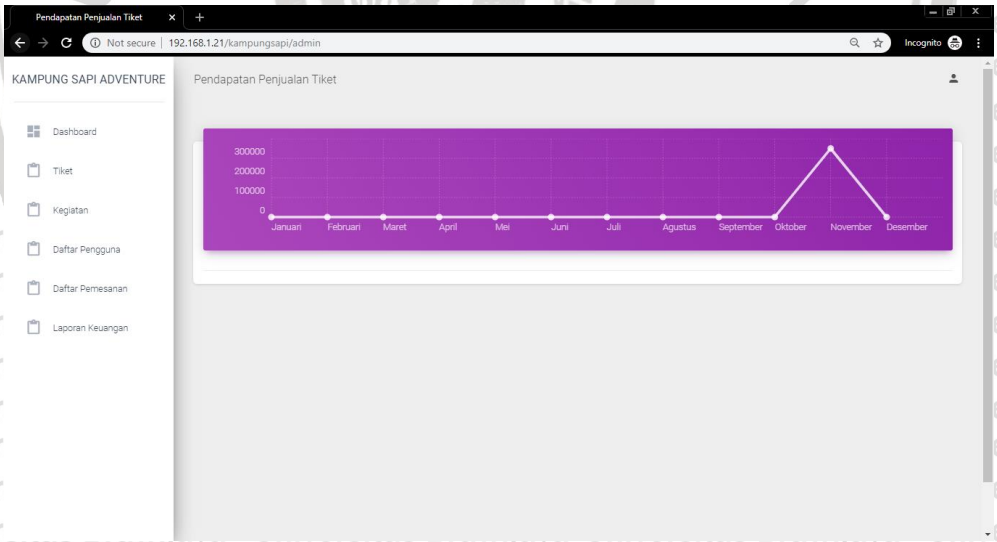


Gambar 5.19 Implementasi Antarmuka Halaman *Logout*

Sumber: (Penulis, 2019)

5.5.6 Implementasi Antarmuka Halaman *Dashboard* Admin

Pada Gambar 5.20 merupakan implementasi antarmuka halaman *dashboard* admin. Pada halaman ini admin dapat melihat pendapatan penjualan tiket berdasarkan bulan melalui line chart. Implementasi antarmuka ini terdapat pada *view dashboard*.php.

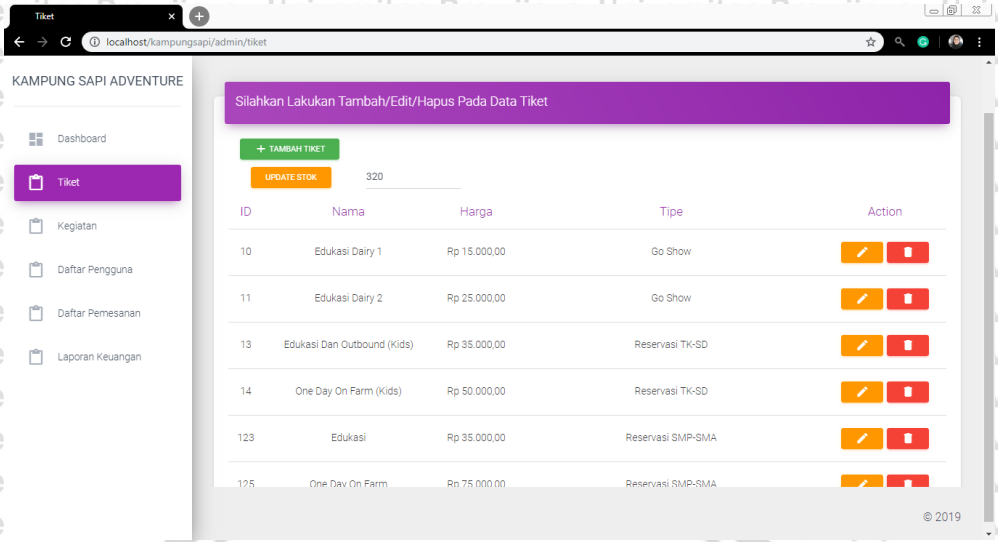


Gambar 5.20 Implementasi Antarmuka Halaman *Dashboard* Admin

Sumber: (Penulis, 2019)

5.5.7 Implementasi Antarmuka Halaman Menu Tiket

Pada Gambar 5.21 merupakan implementasi antarmuka halaman menu tiket. Admin pada halaman menu tiket dapat menambah, mengubah, dan menghapus data tiket. Implementasi antarmuka ini terdapat pada *view* Tiket.php.

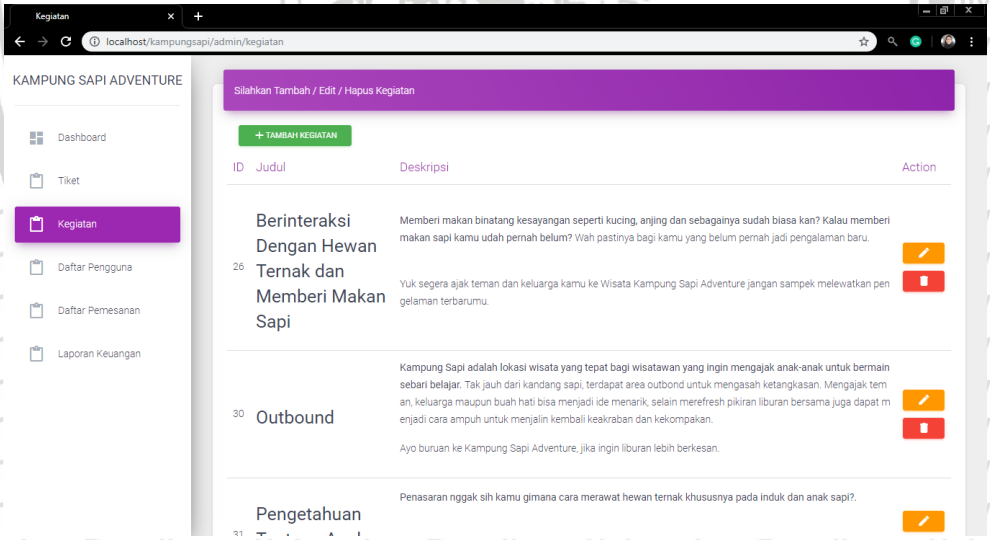


Gambar 5.21 Implementasi Antarmuka Halaman Menu Tiket

Sumber: (Penulis, 2019)

5.5.8 Implementasi Antarmuka Halaman Menu Kegiatan

Pada Gambar 5.22 merupakan implementasi antarmuka halaman menu kegiatan. Admin pada halaman menu kegiatan dapat menambah, mengubah, dan menghapus data tiket. Implementasi antarmuka ini terdapat pada *view* Kegiatan.php.

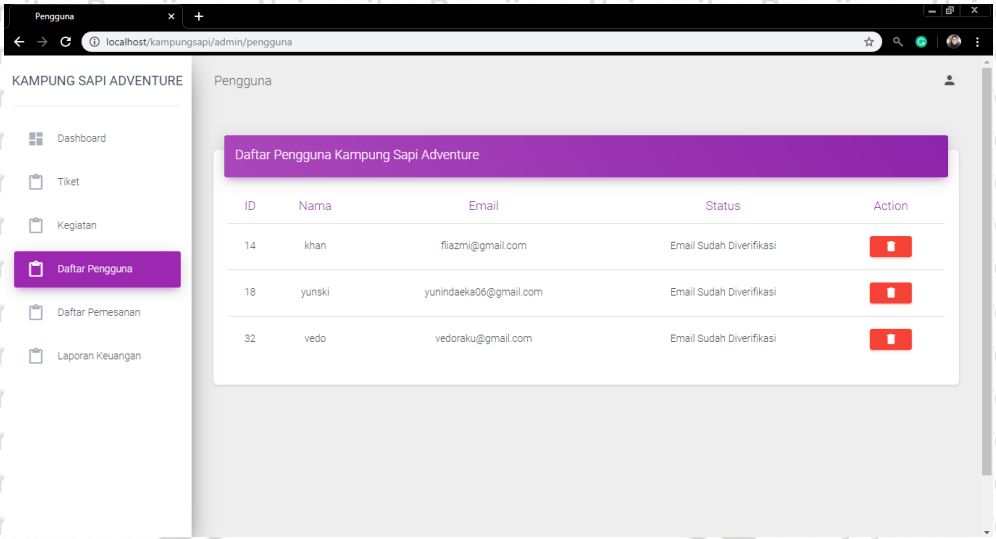


Gambar 5.22 Implementasi Antarmuka Halaman Menu Kegiatan

Sumber: (Penulis, 2019)

5.5.9 Implementasi Antarmuka Halaman Menu Daftar Pengguna

Pada Gambar 5.23 merupakan implementasi antarmuka halaman daftar pengguna. Admin pada halaman ini dapat mengetahui wisatawan yang telah melakukan register dan menghapus daftar pengguna. Implementasi antarmuka ini terdapat pada *view* Pengguna.php.

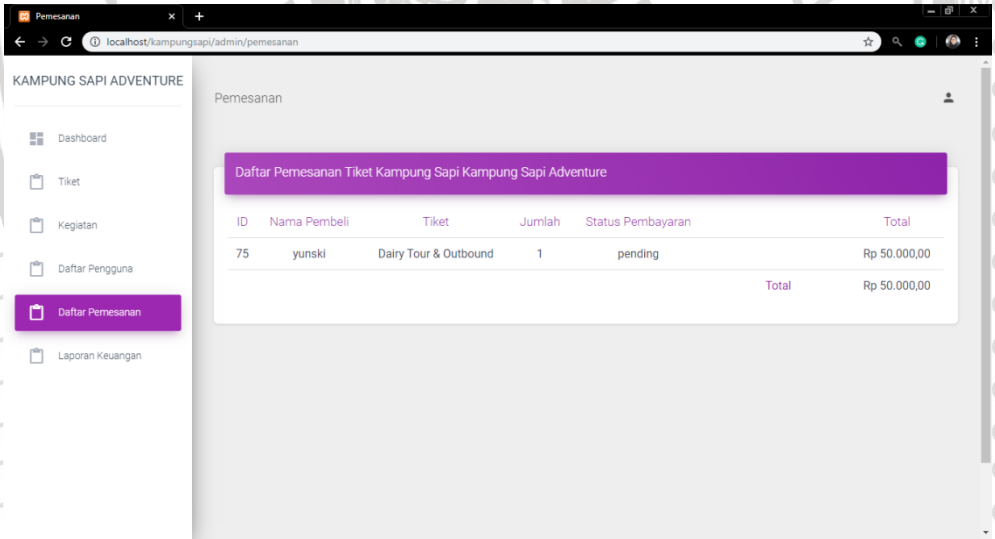


Gambar 5.23 Implementasi Antarmuka Halaman Menu Daftar Pengguna

Sumber: (Penulis, 2019)

5.5.10 Implementasi Antarmuka Halaman Menu Daftar Pemesanan

Pada Gambar 5.24 merupakan implementasi antarmuka halaman daftar pemesanan tiket. Admin pada halaman ini dapat melihat data pemesanan tiket. Implementasi antarmuka ini terdapat pada *view* Pemesanan.php.

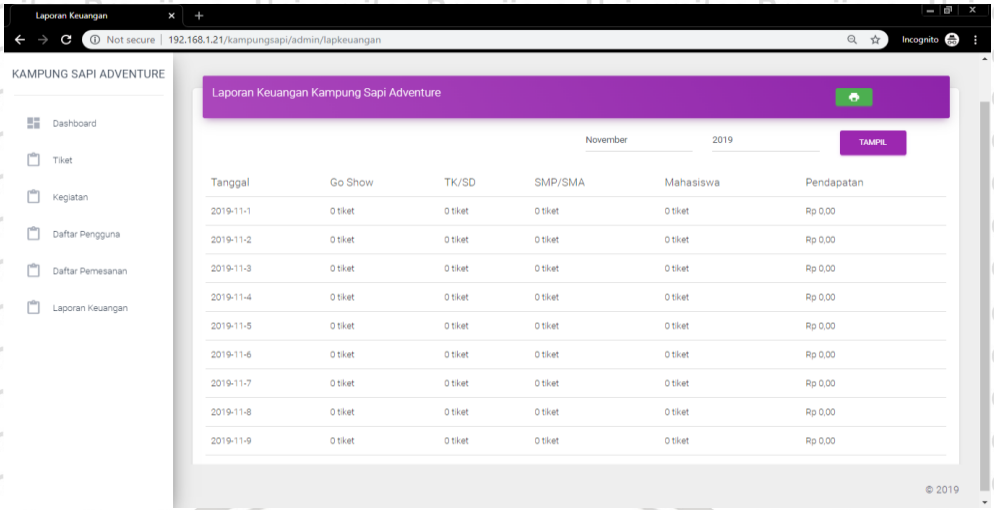


Gambar 5.24 Implementasi Antarmuka Halaman Menu Daftar Pemesanan

Sumber: Penulis (2019)

5.5.11 Implementasi Antarmuka Halaman Menu Laporan Keuangan

Pada Gambar 5.25 merupakan implementasi antarmuka halaman menu laporan keuangan. Admin pada halaman ini dapat melihat laporan keuangan serta mencetak laporan keuangan berdasarkan bulan dan tahun. Implementasi antarmuka ini terdapat pada *view* lapkeuangan.php.



Gambar 5.25 Implementasi Antarmuka Halaman Menu Laporan Keuangan

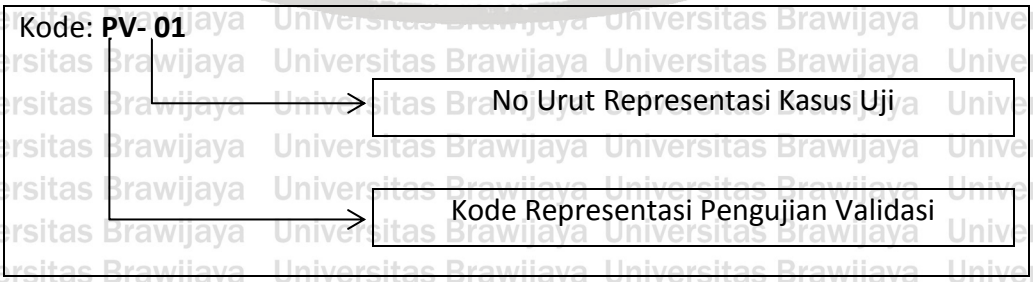
Sumber: (Penulis, 2019)

5.6 Pengujian

Pada bab ini akan dilakukan pengujian terhadap aplikasi e-*commerce* Wisata Kampung Sapi *Adventure*. Pengujian dilakukan untuk memastikan hasil rancangan yang dibuat telah sesuai dengan spesifikasi persyaratan yang telah didefinisikan. Pengujian yang dilakukan dalam penelitian ini adalah dengan melakukan pengujian *black-box* yang akan dilakukan untuk menguji kebutuhan fungsional pada sistem dan *compability testing* dilakukan untuk menguji kebutuhan non – fungsional pada sistem.

5.6.1 Pengujian Fungsional

Pada pengujian *black – box* ini menggunakan uji validasi yang dilakukan untuk memastikan bahwa sistem dapat berjalan sesuai dengan persyaratan sistem yang telah didefinisikan dan dijelaskan sebelumnya. Pengujian kebutuhan fungsional pada aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* dilakukan dengan mendefinisikan kasus uji berdasarkan *use case specification* yang telah didefinisikan pada bagian analisis kebutuhan. Kemudian setiap hasil pengujian yang dilakukan akan dibandingkan dengan hasil yang telah didefinisikan sebelumnya pada *use case specification.*



Gambar 5.26 Aturan Penomoran Pengujian *Black – Box*

Sumber: (Penulis, 2019)

Pada Gambar 5.26 terdapat kode PV-01 yang merupakan aturan penomoran pada pengujian *black - box*.

Pada tabel dibawah ini akan ditunjukkan rencana pengujian validasi dengan kode PV-01 yang dilakukan untuk memastikan bahwa sistem dapat digunakan oleh aktor wisatawan untuk melakukan register ke dalam sistem.

Tabel 5.9 Kasus Uji Validasi Melakukan Register

Kode Pengujian	PV – 01
Nama Kasus Uji	Kasus uji melakukan register
Kode Persyaratan	KF – 01
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan register.
Prosedur Uji	1. Penguji teridentifikasi sebagai wisatawan 2. Penguji memilih menu "register" 3. Penguji memasukkan *username*: "yunski", email: "yuninda06@gmail.com", password: "a123" 4. Penguji menekan tombol register 5. Penguji melakukan verifikasi email
Hasil yang Diharapkan	Sistem menampilkan halaman *login*
Keluaran	Sistem menampilkan halaman *login*
Status Uji	Valid

Sumber: (Penulis, 2019)

Pada tabel dibawah ini akan ditunjukkan rencana pengujian validasi dengan kode PV-02 yang dilakukan untuk memastikan bahwa sistem dapat digunakan oleh aktor wisatawan untuk melakukan *login* ke dalam sistem.

Tabel 5.10 Kasus Uji Validasi Melakukan *Login* Pada Wisatawan

Kode Pengujian	PV – 02
Nama Kasus Uji	Kasus uji melakukan *login*
Kode Persyaratan	KF – 02
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan *login*.

Tabel 5.10 Kasus Uji Validasi Melakukan *Login* Pada Wisatawan (Lanjutan)

Kode Pengujian	PV – 02
Prosedur Uji	1. Penguji teridentifikasi sebagai wisatawan 2. Penguji memilih menu "*login*" 3. Penguji memasukkan username: "yunski" dan password: "a123" 4. Penguji menekan tombol *login*
Hasil yang Diharapkan	Sistem menampilkan halaman utama aplikasi
Keluaran	Sistem menampilkan halaman utama aplikasi
Status Uji	Valid

Sumber: (Penulis, 2019)

Pada tabel dibawah ini akan ditunjukkan rencana pengujian validasi dengan kode PV-03 yang dilakukan untuk memastikan bahwa sistem dapat digunakan oleh aktor wisatawan untuk melakukan *logout*.

Tabel 5.11 Kasus Uji Validasi Melakukan *Logout* Pada Wisatawan

Kode Pengujian	PV – 03
Nama Kasus Uji	Kasus uji melakukan *logout*
Kode Persyaratan	KF – 03
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan *logout*.
Prosedur Uji	1. Penguji teridentifikasi sebagai wisatawan 2. Penguji memilih menu "*logout*"
Hasil yang Diharapkan	Sistem menampilkan halaman utama aplikasi
Keluaran	Sistem menampilkan halaman utama aplikasi
Status Uji	Valid

Sumber: (Penulis, 2019)

Pada tabel dibawah ini akan ditunjukkan rencana pengujian validasi dengan kode PV-04 yang dilakukan untuk memastikan bahwa sistem dapat digunakan oleh aktor wisatawan untuk melihat kegiatan.

Tabel 5.12 Kasus Uji Validasi Melihat Kegiatan Pada Wisatawan

Kode Pengujian	PV – 04
Nama Kasus Uji	Kasus uji melihat kegiatan
Kode Persyaratan	KF – 04
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melihat kegiatan.
Prosedur Uji	1. Penguji teridentifikasi sebagai wisatawan 2. Penguji memilih menu kegiatan
Hasil yang Diharapkan	Sistem menampilkan halaman kegiatan
Keluaran	Sistem menampilkan halaman kegiatan
Status Uji	Valid

Sumber: (Penulis, 2019)

Pada tabel dibawah ini akan ditunjukkan rencana pengujian validasi dengan kode PV-05 yang dilakukan untuk memastikan bahwa sistem dapat digunakan oleh aktor wisatawan untuk melakukan pemesanan tiket pada menu pesan tiket yang terdapat pada sistem.

Tabel 5.13 Kasus Uji Validasi Melakukan Pemesanan Tiket

Kode Pengujian	PV – 05
Nama Kasus Uji	Kasus uji melakukan pemesanan tiket
Kode Persyaratan	KF – 05
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan pemesanan tiket.
Prosedur Uji	1. Penguji teridentifikasi sebagai wisatawan 2. Penguji memilih menu “pesan tiket” 3. Penguji memilih kategori tiket go show 4. Penguji mengetahui fasilitas dan kegiatan paket sesuai kategori tiket 5. Penguji mengetahui harga paket sesuai kategori tiket 6. Penguji mengetahui ketersediaan tiket 7. Penguji memilih kategori tiket reservasi 8. Penguji mengetahui fasilitas dan kegiatan paket sesuai kategori tiket.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 5.13 Kasus Uji Validasi Melakukan Pemesanan Tiket (Lanjutan)

Kode Pengujian	PV – 05
Prosedur Uji	9. Penguji mengetahui harga paket sesuai kategori tiket 10. Penguji melakukan cek tanggal 11. Penguji mengetahui ketersediaan tiket 12. Penguji melakukan pembelian tiket 13. Penguji menginputkan jumlah tiket yang akan dibeli 14. Penguji mengetahui detail pembelian tiket 15. Penguji melakukan pembayaran melalui midtrans 16. Penguji memilih metode pembayaran transfer pada permata bank 17. Penguji mengetahui no rekening yang dituju dan cara pembayaran transfer pada permata bank 18. Penguji melakukan pembayaran menggunakan simulator pada midtrans 19. Penguji berhasil melakukan pembayaran 20. Penguji mendapatkan elektronik tiket
Hasil yang Diharapkan	Sistem memberikan elektronik tiket kepada wisatawan
Keluaran	Sistem memberikan elektronik tiket kepada wisatawan
Status Uji	Valid

Pada tabel dibawah ini akan ditunjukkan rencana pengujian validasi dengan kode PV-06 yang dilakukan untuk memastikan bahwa sistem dapat digunakan oleh aktor admin untuk melakukan *login* ke dalam sistem.

Tabel 5.14 Kasus Uji Validasi *Login* Pada Admin

Kode Pengujian	PV – 06
Nama Kasus Uji	Kasus uji melakukan *login*
Kode Persyaratan	KF – 01
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan *login*.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 5.14 Kasus Uji Validasi *Login* Pada Admin (Lanjutan)

Kode Pengujian	PV – 06
Prosedur Uji	1. Penguji teridentifikasi sebagai admin 2. Penguji memilih menu "*login*" 3. Penguji memasukkan *username*: "admin" dan *password*: "admin" 4. Penguji menekan tombol *login*
Hasil yang Diharapkan	Sistem menampilkan *dashboard* admin
Keluaran	Sistem menampilkan *dashboard* admin
Status Uji	Valid

Sumber: (Penulis, 2019)

Pada tabel dibawah ini akan ditunjukkan rencana pengujian validasi dengan kode PV-07 yang dilakukan untuk memastikan bahwa sistem dapat digunakan oleh aktor admin untuk melakukan *logout*.

Tabel 5.15 Kasus Uji Validasi *Logout* Pada Admin

Kode Pengujian	PV – 07
Nama Kasus Uji	Kasus uji melakukan *logout*
Kode Persyaratan	KF – 02
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan *logout*.
Prosedur Uji	1. Penguji teridentifikasi sebagai admin 2. Penguji memilih menu "*logout*"
Hasil yang Diharapkan	Sistem menampilkan halaman *dashboard* aplikasi
Keluaran	Sistem menampilkan halaman *dashboard* aplikasi
Status Uji	Valid

Pada tabel dibawah ini akan ditunjukkan rencana pengujian validasi dengan kode PV-08 yang dilakukan untuk memastikan bahwa sistem dapat digunakan oleh aktor admin untuk melihat halaman *dashboard* admin yang terdapat pada sistem.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 5.16 Kasus Uji Melihat Halaman *Dashboard* Admin

Kode Pengujian	PV – 08
Nama Kasus Uji	Kasus uji melihat halaman *dashboard* admin
Kode Persyaratan	KF – 03
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melihat grafik pendapatan penjualan tiket.
Prosedur Uji	1. Penguji teridentifikasi sebagai admin 2. Penguji memilih menu "*Dashboard*"
Hasil yang Diharapkan	Sistem menampilkan grafik pendapatan penjualan tiket
Keluaran	Sistem menampilkan grafik pendapatan penjualan tiket
Status Uji	Valid

Sumber: (Penulis, 2019)

Pada tabel dibawah ini akan ditunjukkan rencana pengujian validasi dengan kode PV-09 yang dilakukan untuk memastikan bahwa sistem dapat digunakan oleh aktor admin untuk mengelola data tiket pada menu tiket yang terdapat pada sistem.

Tabel 5.17 Kasus Uji Mengelola Data Tiket

Kode Pengujian	PV – 09
Nama Kasus Uji	Kasus uji mengelola data tiket
Kode Persyaratan	KF – 04
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk mengelola data tiket.
Prosedur Uji	1. Penguji teridentifikasi sebagai admin 2. Penguji memilih menu "Tiket" 3. Penguji menekan tombol tambah untuk menambah data tiket 4. Penguji menekan tombol edit untuk melakukan perubahan pada data tiket 5. Penguji menekan tombol hapus untuk menghapus data tiket

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 5.17 Kasus Uji Mengelola Data Tiket (Lanjutan)

Kode Pengujian	PV – 09
Hasil yang Diharapkan	Sistem menjalankan aktivitas mengelola data tiket yang dilakukan admin pada halaman tiket
Keluaran	Sistem menjalankan aktivitas mengelola data tiket yang dilakukan admin pada halaman tiket
Status Uji	Valid

Sumber: (Penulis, 2019)

Pada tabel dibawah ini akan ditunjukkan rencana pengujian validasi dengan kode PV-10 yang dilakukan untuk memastikan bahwa sistem dapat digunakan oleh aktor admin untuk mengelola data kegiatan pada menu kegiatan yang terdapat pada sistem.

Tabel 5.18 Kasus Uji Mengelola Data Kegiatan

Kode Pengujian	PV – 10
Nama Kasus Uji	Kasus uji mengelola data kegiatan
Kode Persyaratan	KF – 05
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk mengelola data kegiatan.
Prosedur Uji	1. Penguji teridentifikasi sebagai admin 2. Penguji memilih menu “Kegiatan”
Prosedur Uji	3. Penguji menekan tombol tambah untuk menambah data kegiatan 4. Penguji menekan tombol edit untuk melakukan perubahan pada data kegiatan 5. Penguji menekan tombol hapus untuk menghapus data kegiatan
Hasil yang Diharapkan	Sistem menjalankan aktivitas mengelola data kegiatan yang dilakukan admin pada halaman kegiatan
Keluaran	Sistem menjalankan aktivitas mengelola data tiket yang dilakukan admin pada halaman kegiatan
Status Uji	Valid

Sumber: (Penulis, 2019)

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Pada tabel dibawah ini akan ditunjukkan rencana pengujian validasi dengan kode PV-11 yang dilakukan untuk memastikan bahwa sistem dapat digunakan oleh aktor admin untuk melihat daftar pemesanan tiket pada menu daftar pemesanan yang terdapat pada sistem.

Tabel 5.19 Kasus Uji Melihat Daftar Pemesanan Tiket

Kode Pengujian	PV – 11
Nama Kasus Uji	Kasus uji melihat daftar pemesanan tiket
Kode Persyaratan	KF – 06
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melihat daftar pemesanan tiket.
Prosedur Uji	1. Penguji teridentifikasi sebagai admin 2. Penguji memilih menu "Daftar Pemesanan"
Hasil yang Diharapkan	Sistem menampilkan halaman daftar pemesanan tiket
Keluaran	Sistem menampilkan halaman daftar pemesanan tiket
Status Uji	Valid

Sumber: (Penulis, 2019)

Pada tabel dibawah ini akan ditunjukkan rencana pengujian validasi dengan kode PV-12 yang dilakukan untuk memastikan bahwa sistem dapat digunakan oleh aktor admin untuk mengelola data pada menu daftar pengguna yang terdapat pada sistem.

Tabel 5.20 Kasus Uji Mengelola Data Daftar Pengguna

Kode Pengujian	PV – 12
Nama Kasus Uji	Kasus uji mengelola data daftar pengguna
Kode Persyaratan	KF – 07
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk mengelola data daftar pengguna .
Prosedur Uji	1. Penguji teridentifikasi sebagai admin 2. Penguji memilih menu "Daftar Pengguna" 3. Penguji menekan tombol hapus untuk menghapus pengguna
Hasil yang Diharapkan	Sistem menjalankan aktivitas mengelola data daftar pengguna yang dilakukan admin pada halaman daftar pengguna

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 5.20 Kasus Uji Mengelola Data Daftar Pengguna

Kode Pengujian	PV – 12
Keluaran	Sistem menjalankan aktivitas mengelola data daftar pengguna yang dilakukan admin pada halaman daftar pengguna
Status Uji	Valid

Sumber: (Penulis, 2019)

Pada tabel dibawah ini akan ditunjukkan rencana pengujian validasi dengan kode PV-13 yang dilakukan untuk memastikan bahwa sistem dapat digunakan oleh aktor admin untuk melihat laporan keuangan pada menu laporan keuangan yang terdapat pada sistem.

Tabel 5.21 Kasus Uji Melihat Laporan Keuangan

Kode Pengujian	PV – 13
Nama Kasus Uji	Kasus uji melihat laporan keuangan
Kode Persyaratan	KF – 08
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melihat laporan keuaangan
Prosedur Uji	1. Penguji teridentifikasi sebagai admin 2. Penguji memilih menu “Laporan Keuangan”
Hasil yang Diharapkan	Sistem menampilkan halaman laporan keuangan berdasarkan bulan dan tahun
Keluaran	Sistem menampilkan halaman laporan keuangan
Status Uji	Valid

Sumber: (Penulis, 2019)

Pada tabel dibawah ini akan ditunjukkan rencana pengujian validasi dengan kode PV-14 yang dilakukan untuk memastikan bahwa sistem dapat digunakan oleh aktor admin untuk mencetak laporan keuangan pada menu laporan keuangan yang terdapat pada sistem.

Tabel 5.22 Kasus Uji Mencetak Laporan Keuangan

Kode Pengujian	PV – 14
Nama Kasus Uji	Kasus uji mencetak laporan keuangan
Kode Persyaratan	KF – 09

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 5.23 Kasus Uji Mencetak Laporan Keuangan

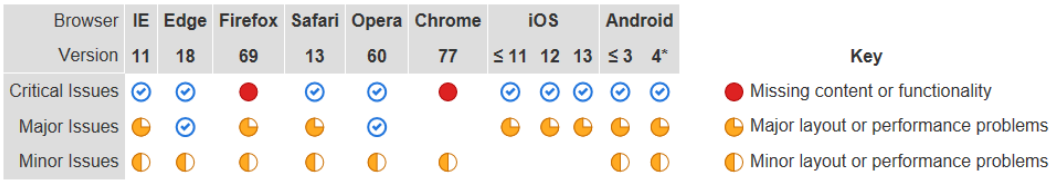
Kode Pengujian	PV – 14
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk mencetak laporan keuangan
Prosedur Uji	1. Penguji teridentifikasi sebagai admin 2. Penguji memilih menu "Laporan Keuangan" 3. Penguji menekan tombol print 4. Laporan keuangan terunduh
Hasil yang Diharapkan	Sistem menjalankan fungsi mencetak laporan keuangan
Keluaran	Sistem menjalankan fungsi mencetak laporan keuangan
Status Uji	Valid

Sumber: (Penulis, 2019)

Berdasarkan pengujian kebutuhan fungsional yang telah dilakukan pada waisatawan dan admin terbukti valid yang artinya aplikasi ini menunjukkan adanya keselarasan dengan rancangan sistem.

5.6.2 Pengujian Non-Fungsional

Pengujian validasi kebutuhan non-fungsional pada aplikasi e-commerce Wisata Kampung Sapi *Adventure* menggunakan uji *compatibility*. Uji *compatibility* digunakan untuk membuktikan bahwa aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* dapat diakses pada browser yang berbeda – beda. Pada pengujian *compatibility* ini menggunakan *software sortsite*. Hasil uji validasi kebutuhan non-fungsional pada aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* dapat dilihat pada Gambar 5.27.



Gambar 5.27 Hasil Uji *Compatibility* Menggunakan *Software Sortsite*

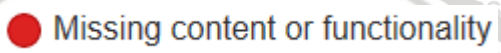
Sumber: (Penulis, 2019)

Hasil pengujian *compability* yang telah dilakukan menggunakan *software sortsite* pada Gambar 5.27 menunjukkan bahwa aplikasi tidak kompatibel pada browser *firefox* versi 69 dan *chrome* versi 77. Permasalahan tersebut terjadi karena pada browser *firefox* dan *chrome* merupakan *browser* yang mendukung tingkat keamanan sebuah website. Jika sebuah *website* tidak menggunakan protokol HTTPS, maka website tersebut dianggap sebagai *critical issues. Critical*

issues terjadi pada halaman *login* dan register, ini terjadi karena dalam menjalankan aplikasi menggunakan *localhost*. Terdapat masalah pada *major issues* ketika aplikasi diakses menggunakan *internet explorer, firefox, safari, iOS,* dan *android*. Sedangkan masalah pada *minor issues* terjadi ketika aplikasi diakses menggunakan *internet explorer, edge, firefox, safari, opera, chrome,* dan *android*. Permasalahan yang tejadi pada *major issues dan minor issues* tersebut terjadi karena *browser* tersebut tidak mendukung *bootstrap.*

Berdasarkan hasil pengujian non-fungsional yang telah dilakukan dengan menggunakan *software sortsite* dapat disimpulkan bahwa aplikasi dapat dijalankan pada beberapa *browser*.

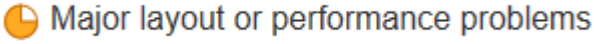
Pada Gambar 5.28 sampai Gambar 2.30 terdapat keterangan terkait masalah yang terdapat pada *critical issues, major issues,* dan *minor issues*.



Gambar 5.28 Petunjuk Adanya Masalah Pada *Critical Issues*

Sumber: (Penulis, 2019)

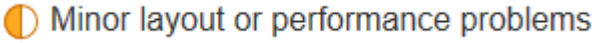
Gambar 5.28 merupakan simbol yang menunjukkan bahwa adanya masalah kehilangan konten atau fungsionalitas ketika aplikasi dijalankan pada sebuah *browser*.



Gambar 5.29 Petunjuk Adanya Masalah Pada *Mayor Issues*

Sumber: (Penulis, 2019)

Gambar 5.29 merupakan simbol yang menunjukkan bahwa adanya masalah pada tata letak atau kinerja berat, dan gambar ketika aplikasi dijalankan pada sebuah *browser*.



Gambar 5.30 Petunjuk Adanya Masalah *Minor Issues*

Sumber: (Penulis, 2019)

Gambar 5.30 merupakan simbol yang menunjukkan bahwa adanya masalah pada tata letak atau kinerja ringan ketika aplikasi dijalankn pada sebuah *browser*.

BAB 6 PENUTUP

6.1 Kesimpulan

Berdasarkan hasil analisi dan perancangan, implementasi, dan pengujian yang telah dilakukan, maka diambil kesimpulan sebagai berikut:

1. Proses analisis aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* dilakukan dengan melakukan studi literatur melalui wawancara mengenai proses bisnis yang telah berlangsung pada Wisata Kampung Sapi *Adventure* yang nantinya peneliti dapat melakukan perbaikan dengan membuat proses bisnis usulan serta mendeskripsikan kebutuhan fungsional maupun non-fungsional. Pada proses bisnis saat ini, transaksi jual beli tiket yang dibeli di hari yang sama atau disebut dengan tiket *go show* maupun reservasi masih dilakukan secara manual yaitu wisatawan mengirimkan foto bukti pembayaran transfer. Kemudian petugas wisata akan menginputkan data transaksi pembelian tiket ke dalam aplikasi pemrosesan *spreadsheet*. Pada proses bisnis yang diusulkan peneliti, ada beberapa perubahan aktivitas dalam proses bisnis yaitu, transaksi pembelian tiket menggunakan *payment gateway* midtrans dan pendataan transaksi dengan menggunakan aplikasi. Hasil dari analisis wawancara yang telah dilakukan sebelumnya didapat 5 kebutuhan fungsional pada aktor wisatawan dan 9 kebutuhan fungsional pada aktor admin dan 1 kebutuhan non-fungsional. Sedangkan pada penelitian ini perancangan aplikasi *e-commerce* Wisata Kampung Sapi *Adventure* menggunakan bahasa pemodelan *Unifed Modelling Language* (UML) yang meliputi 12 *use case diagram,* 12 *activity diagram*, dan 12 *sequence diagram*.

2. Proses implementasi dilakukan dengan integrasi *payment gateway* dengan memanfaatkan layanan API *sandbox* dari midtrans menggunakan web service untuk mengambil data yang diperlukan. Pengimplementasian menggunakan bahasa pemrograman PHP, CSS, HTML, Javascript dan menggunakan *framework CodeIgniter*.

3. Pengujian dilakukan dengan menguji setiap kebutuhan fungsional dan non-fungsional yang ada pada wisatawan dan admin. Hasil pengujian dari 14 pengujian yang diujikan terbukti valid yang artinya sistem dikembangkan sesuai dengan persyaratan sistem yang telah didefinisikan sebelumnya. Pada pengujian non-fungsional yang telah dilakukan dengan *software* sortsite dinyatakan bahwa aplikasi sudah dapat dijalankan pada browser yang memungkinkan digunakan oleh pengguna. Namun, masih terdapat *critical issues* yang terjadi pada fungsi *login* pada browser *firefox* versi 69 dan *chrome* versi 77.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

6.2 Saran

Saran yang dapat diberikan setelah menyelesaikan penelitian skripsi ini adalah sebagai berikut:

1. Aplikasi dapat dikembangkan dengan meninjau kembali pada kebutuhan fungsional dan non-fungsional untuk dapat dilakukan penambahan sebuah fitur pada aplikasi, serta membuat rancangan antarmuka agar lebih *user friendly*.

DAFTAR REFERENSI

A.S., R. & M.Shalahuddin, 2016. *Sistem Informasi Manajemen konsep dan pengembangannya.* Bandung: Informatika Bandung.

Aditama, R., 2017. *Web Service Pembayaran Uang Keliah Online Dengan PHP dan SOAP WSDL.* Yogyakarta: Lokomedia.

Aribowo, 2019. *Penjelasan Mengenai Wisata Kampung Sapi Adventure* [Interview] (20 Agustus 2019).

Aribowo, 2019. *Proses Bisnis Wisata Kampung Sapi Adventure* [Interview] (20 August 2019).

Booch, G., Rumbaugh, J. & Jacobson, I., 2005. *The Unified Modeling Language User Guide.* 2nd ed. Canada: Addison Wesley Professional.

Fielding, R. T., 2000. *Architectural Styles and the Design of Network-based Software Architectures.* s.l., University Of California.

Griffiths, A., 2010. *CodeIgniter 1.7 Professional Development.* Birmingham: Packt Publishing.

Jeffrey Whitten, L. B., 2007. *Systems Analysis & Design Methods.* 7nd ed. California: McGraw-Hill Education.

Kendall, K. & Kendall, J., 2011. *System Analysis and Design.* 8nd ed. New Jersey: Person Education.

Marthasari, G., Aminudin & Munarko, Y., 2010. Implementasi Web Service Untuk Mendukung Interoperabilitas Pada Aplikasi E-commerce. *EEPIS.*

Midtrans (PT Midtrans), 2019. *Sejarah Midtrans.* [Online] Available at: https://midtrans.com/about [Accessed 27 August 2019].

Mukherjee, M. & Roy, S., 2017. E-Commerce and Online Payment in the Modern Era. *International Journal of Advanced Research in Computer Science and Software Engineering,* 7(5), pp. 1-5.

Notation, B. P. M., 2019. *BPMN.* [Online] Available at: http://www.bpmn.org/ [Accessed 29 Juli 2019].

Pattnaik, S., Ghosh, P. & Bharti, A., 2005. The Working Model Of An E-Payment System. *Journal of Theoretical and Applied Information Technology,* pp. 10-13.

Perancangan Aplikasi E-Commerce Wisata Kampung Sapi Adventure (2019) Penulis.

Pressman, R., 2010. *Software Engineering: a Practitioner's Approach.* 2nd ed. Los Angeles: McGraw-Hill Education.

Qin, Z., 2009. *Introduction to E-commerce.* Beijing: Tsinghua University Press; Springer-Verlag GmbH Berlin Heidelberg.

Riyanto, Utami, E. & Amborowaty, A., 2013. Pemanfaatan Web Service Pada Integrasi Data Farmasi Di RSU Banyumas. *Seminar Nasional Informatika 2013 (SemnasIF 2013).*

Sarwono, J. & Prihartono, K., 2012. *Perdagangan Online: Cara Bisnis di Internet.* [e-book] Jakarta: PT Elex Media Komputindo. Tersedia di: Google Books <http://booksgoogle.com> [Diakses 29 Agustus 2019].

Sommerville, I., 2011. *Software Engineering: Ninth Edition.* Boston: Pearson.

Weske, M., 2007. *Business Process Management Concept, Languages, Architectures.* 2nd ed. Berlin: Springer.

LAMPIRAN A HASIL WAWANCARA

Topik : Wawancara terstruktur untuk mengetahui proses bisnis saat ini pada Wisata Kampung Sapi *Adventure*
Hari, Tanggal : Selasa, 20 Agustus 2019
Tempat : Wisata Kampung Sapi *Adventure*
Narasumber : Aribowo
(Pengelola Wisata Kampung Sapi *Adventure*)

Pewawancara : Konsep apa yang yang dihadirkan dalam Wisata Kampung Sapi *Adventure*?

Narasumber : Sebuah peternakan sapi yang dijadikan sebagai tempat objek wisata edukasi sekaligus mengembangkan konsep agrowisata. Wisatawan tidak hanya bermain namun juga mendapatkan pengetahuan dan pengalaman mengenai segala aktivitas yang ada di dipeternakan dan cara mengolah sebuah produk olahan seperti susu sapi, sabun dari susu sapi, yogurt dan masih banyak lagi.

Pewawancara : Melalui media apa pihak wisata mempromosikan Wisata Kampung Sapi *Adventure* kepada masyarakat?

Narasumber : Pempromosian kepada masyarakat hanya melalui media sosial Facebook, Intagram, dan Whatsapp. Dalam seminggu ada 1 hari beriklan berbayar pada Facebook maupun Instagram.

Pewawancara : Siapa saja target wisatawan pada Wisata Kampung Sapi *Adventure*?

Narasumber : Terdapat 3 model wisatawan yaitu, rombongan sekolah, umum, dan mencari inspirasi. Wisatawan dengan kategori grup terdiri dari kelompok bermain sampai SMP. Wisatawan dengan kategori umum tidak memiliki batasan dan tidak terpaut umur. Dan wisatawan dengan kategori mencari inspirasi terdiri dari pelajar SMA sampai para pensiunan.

Pewawancara : Dalam 1 hari Wisata Kampung Sapi *Adventure* dapat menampung berapa wisatawan?

Narasumber : Wisata Kampung Sapi *Adventure* dalam 1 hari mampu menampung 350 pengunjung.

Pewawancara : Ada berapa jenis tiket yang ditawarkan pada wisata ini?

Narasumber : Pada wisata ini terdapat 2 jenis kategori tiket yaitu tiket yang dapat dijual dan dibeli pada hari yang sama (*go show*) dan tiket reservasi. Tiket reservasi dimana wisatawan dapat melakukan pemesanan tiket dua hari sebelum hari H.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Pewawancara : Apakah dari kedua kategori tiket terdapat macam – macam paket tiket?

Narasumber : Iya. Pada kategori tiket go show dengan model wisatawan umum terdapat 2 paket tiket yaitu edukasi 1 dan edukasi 2. Kategori tiket reservasi dengan model wisatawan rombongan sekolah pada siswa TK dan SD terdapat 2 paket yaitu edukasi dan outbound (kids), one day on farm (kids). Kemudian kategori tiket reservasi dengan model wisatawan rombongan sekolah pada siswa SMP – SMA terdiri dari 3 paket yaitu paket edukasi, one day on farm, dan dairy tour dan outbound. Selanjutnya tiket reservasi dengan model wisatawan mencari inspirasi terdapat 3 paket inspirasi bisnis 1, inspirasi bisnis 2, dan inspirasi bisnis dan outbound. Masing paket pada kategori tiket memiliki fasilitas dan kegiatan serta harga yang berbeda – beda.

Pewawancara : Apakah sudah ada pembangunan sistem informasi pada wisata ini untuk media pempromosian maupun transaksi penjualan dan pembelian tiket?

Narasumber : Belum ada.

Pewawancara : Melalui media apa pihak wisata mempromosikan Wisata Kampung Sapi *Adventure* kepada masyarakat?

Narasumber : Pempromosian kepada masyarakat hanya melalui media sosial Facebook, Intagram, dan Whatsapp. Dalam seminggu ada 1 hari beriklan berbayar pada Facebook maupun Instagram.

Pewawancara : Apakah melalui ketiga media sosial Facebook, Instagram, dan Whatsapp dirasa sudah cukup efektif dalam hal mempromosikan wisata ini kepada masyarakat?

Narasumber : Belum, dikarenakan masih banyak masyarakat yang belum mempunyai salah satu dari ketiga media sosial seperti Facebook, Instagram dan kami (pihak wisata) tidak dapat secara bebas memuat sebuah konten pempromosian dengan menggunakan media sosial seperti terkadang sebuah foto maupun video yang telah diupload tidak jelas atau terpotong dikarenakan format yang diberikan pada platform tersebut terbatas, masalah tersebut dapat mengurangi daya tarik dari calon wisatawan.

Pewawancara : Bagaimana mekanisme transaksi penjualan dan pembelian tiket pada Wisata Kampung Sapi *Adventure*?

Narasumber : Transaksi penjualan tiket dan pembelian tiket dapat dilakukan di loket penjualan tiket maupun melalui media sosial seperti Instagram, Facebook, dan Whatsapp. Dari ketiga media sosial

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

tersebut wisatawan lebih sering melakukan pembelian maupun pemesanan tiket melalui Whatsapp.

Pewawancara : Bagaimana mekanisme transaksi pembelian tiket melalui Whatsapp?

Narasumber : Ketika wisatawan melakukan pembelian melalui whatsapp, maka pihak wisata kampung sapi segera membalas pesan tersebut dengan memberikan pilihan kategori dan paket tiket. Setelah wisatawan memilih kategori dan paket tiket, maka pihak wisata akan memberikan formulir mengenai detail pembelian tiket kepada wisatawan yang wajib diisi secara lengkap seperti nama pemesan, kategori dan paket tiket yang dipilih, tanggal berkunjung, jumlah tiket yang dibeli, email ataupun no telfon yang dapat dihubungi. Jika wisatawan telah mengirim formulir tersebut, maka pihak wisata memberikan informasi cara pembayaran melalui transfer bank. Setelah wisatawan melakukan pembayaran, maka wisatawan diharuskan mengirimkan bukti pembayaran. Selanjutnya pihak wisata memberikan informasi tiket yang telah dibeli dapat diambil.

Pewawancara : Bagaimana laporan keuangan untuk pendataan serta pendapatan penjualan dan pembelian tiket?

Narasumber : Laporan keuangan saat ini masih dilakukan secara manual dengan menginputkan data penjualan dan pembelian tiket ke dalam spreadsheet. Jadi jika ada wisatawan yang melakukan pembelian melalui media sosial maupun secara langsung diloket tiket maka akan dicatat pada spreadsheet. Untuk mengetahui jumlah pendapatan yang diperoleh maka data transaksi penjualan yang telah dicatat akan dijumlahkan berdasarkan bulan dan tahun.

Mengetahui,

Pengelola Wisata Kampung Sapi *Adventure*

Aribowo

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

LAMPIRAN B HASIL PENGUJIAN VALIDASI

Kuesioner Pengujian Validasi kepada Administrasi Wisata Kampung Sapi *Adventure*

Pengguna : Administrasi

Nama : ARIBOWO

Umur : 34 tahun.

Jabatan : SUPERVISOR KAMPUNG SAPI ADVENTURE

Petunjuk Pengisian

1. Penguji adalah administrasi Wisata Kampung Sapi *Adventure.*
2. Penguji menggunakan fungsi *login* dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "valid" apabila keluaran sesuai dengan harapan dan diisi dengan "tidak valid" apabila keluaran tidak sesuai harapan.

Kode Pengujian	PV – 05
Nomor Skenario	Skenario 1
Kode Persyaratan	KF – 01
Tujuan Penelitian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan *login*
Prosedur Uji	1. Penguji teridentifikasi sebagai administrasi Kampung Sapi *Adventure.* 2. Penguji memilih menu *login.* 3. Penguji memasukkan *username*: "admin" dan *password*: "admin". 4. Penguji menekan tombol *login*.
Hasil yang diharapkan	Sistem menampilkan halaman utama admin
Keluaran	1. Keluaran sesuai dengan hasil yang diharapkan 2. Keluaran tidak sesuai yang diharapkan
Status Uji	VALID

Batu, 7 Desember 2019

Aribowo

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Kode Pengujian	PV – 05
Nomor Skenario	Skenario 1.1
Kode Persyaratan	KF – 01
Tujuan Penelitian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan *login*
Prosedur Uji	1. Penguji teridentifikasi sebagai administrasi Kampung Sapi *Adventure.* 2. Penguji memilih menu *login.* 3. Penguji memasukkan *username*: "admin" dan *password*: "sayaadmin". 4. Penguji menggunakan menekan tombol *login.*
Hasil yang diharapkan	Sistem menampilkan username or password invalid
Keluaran	1. Keluaran sesuai dengan hasil yang diharapkan 2. Keluaran tidak sesuai yang diharapkan
Status Uji	VALID.

Petunjuk Pengisian

1. Penguji adalah administrasi Wisata Kampung Sapi *Adventure.*
2. Penguji menggunakan fungsi *logout* dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "valid" apabila keluaran sesuai dengan harapan dan diisi dengan "tidak valid" apabila keluaran tidak sesuai harapan.

Kode Pengujian	PV – 06
Nomor Skenario	Skenario 2
Kode Persyaratan	KF – 02
Tujuan Penelitian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan *logout*
Prosedur Uji	1. Penguji teridentifikasi sebagai administrasi Kampung Sapi *Adventure.* 2. Penguji telah berhasil masuk ke dalam sistem. 3. Penguji menekan tombol *logout.*
Hasil yang diharapkan	Sistem menampilkan halaman utama aplikasi
Keluaran	1. Keluaran sesuai dengan hasil yang diharapkan 2. Keluaran tidak sesuai yang diharapkan
Status Uji	VALID

Petunjuk Pengisian

1. Penguji adalah administrasi Wisata Kampung Sapi *Adventure*.
2. Penguji menggunakan fungsi *dashboard* dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "valid" apabila keluaran sesuai dengan harapan dan diisi dengan "tidak valid" apabila keluaran tidak sesuai harapan.

Kode Pengujian	PV – 07
Nomor Skenario	Skenario 3
Kode Persyaratan	KF – 03
Tujuan Penelitian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melihat halaman *dashboard*
Prosedur Uji	1. Penguji teridentifikasi sebagai administrasi Kampung Sapi *Adventure*. 2. Penguji memilih menu *"Dashboard"*.
Hasil yang diharapkan	Sistem menampilkan halaman *dashboard* berupa line chart data pendapatan per bulan
Keluaran	1. Keluaran sesuai dengan hasil yang diharapkan 2. Keluaran tidak sesuai yang diharapkan
Status Uji	VALID.

Petunjuk Pengisian

1. Penguji adalah administrasi Wisata Kampung Sapi *Adventure*.
2. Penguji menggunakan fungsi tiket dengan prosedur uji dan nilai masukan (*input*) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "valid" apabila keluaran sesuai dengan harapan dan diisi dengan "tidak valid" apabila keluaran tidak sesuai harapan.

Kode Pengujian	PV – 08
Nomor Skenario	Skenario 4
Kode Persyaratan	KF – 04
Tujuan Penelitian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk mengelola data tiket
Prosedur Uji	1. Penguji teridentifikasi sebagai administrasi Kampung Sapi *Adventure*. 2. Penguji memilih menu "Tiket". 3. Penguji menekan tombol tambah untuk menambah data tiket. 4. Penguji menekan tombol edit untuk melakukan perubahan pada data tiket 5. Penguji menekan tombol hapus untuk menghapus data tiket
Hasil yang diharapkan	Sistem menjalankan aktivitas mengelola data tiket yang dilakukan admin pada halaman tiket
Keluaran	1. Keluaran sesuai dengan hasil yang diharapkan 2. Keluaran tidak sesuai yang diharapkan
Status Uji	VALID.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Petunjuk Pengisian

1. Penguji adalah administrasi Wisata Kampung Sapi *Adventure*.
2. Penguji menggunakan fungsi kegiatan dengan prosedur uji dan nilai masukan (*input*) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "valid" apabila keluaran sesuai dengan harapan dan diisi dengan "tidak valid" apabila keluaran tidak sesuai harapan.

Kode Pengujian	PV – 09
Nomor Skenario	Skenario 4
Kode Persyaratan	KF – 04
Tujuan Penelitian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk mengelola data kegiatan
Prosedur Uji	1. Penguji teridentifikasi sebagai administrasi Kampung Sapi *Adventure*. 2. Penguji memilih menu "Kegiatan". 3. Penguji menekan tombol tambah untuk menambah data kegiatan 4. Penguji menekan tombol pensil untuk melakukan perubahan pada data kegiatan 5. Penguji menekan tombol hapus untuk menghapus data kegiatan
Hasil yang diharapkan	Sistem menjalankan aktivitas mengelola data kegiatan yang dilakukan admin pada halaman kegiatan
Keluaran	1. Keluaran sesuai dengan hasil yang diharapkan 2. Keluaran tidak sesuai yang diharapkan
Status Uji	VALID

Petunjuk Pengisian

1. Penguji adalah administrasi Wisata Kampung Sapi *Adventure*.
2. Penguji menggunakan fungsi daftar pemesanan dengan prosedur uji dan nilai masukan (*input*) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "valid" apabila keluaran sesuai dengan harapan dan diisi dengan "tidak valid" apabila keluaran tidak sesuai harapan.

Kode Pengujian	PV – 10
Nomor Skenario	Skenario 5
Kode Persyaratan	KF – 06
Tujuan Penelitian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melihat daftar pemesanan tiket
Prosedur Uji	1. Penguji teridentifikasi sebagai administrasi Kampung Sapi *Adventure*. 2. Penguji memilih menu "Daftar Pemesanan".
Hasil yang diharapkan	Sistem menampilkan halaman daftar pemesanan
Keluaran	1. Keluaran sesuai dengan hasil yang diharapkan 2. Keluaran tidak sesuai yang diharapkan
Status Uji	VALID

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Petunjuk Pengisian

1. Penguji adalah administrasi Wisata Kampung Sapi *Adventure.*
2. Penguji menggunakan fungsi daftar pengguna dengan prosedur uji dan nilai masukan (*input*) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "valid" apabila keluaran sesuai dengan harapan dan diisi dengan "tidak valid" apabila keluaran tidak sesuai harapan.

Kode Pengujian	PV – 11
Nomor Skenario	Skenario 6
Kode Persyaratan	KF – 07
Tujuan Penelitian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk mengelola daftar pengguna
Prosedur Uji	1. Penguji teridentifikasi sebagai administrasi Kampung Sapi *Adventure.* 2. Penguji memilih menu "Daftar Pengguna". 3. Penguji menekan tombol hapus untuk menghapus pengguna.
Hasil yang diharapkan	Sistem menjalankan aktivitas mengelola data daftar pengguna yang dilakukan admin pada halaman daftar pengguna
Keluaran	1. Keluaran sesuai dengan hasil yang diharapkan 2. Keluaran tidak sesuai yang diharapkan
Status Uji	VALID

Petunjuk Pengisian

1. Penguji adalah administrasi Wisata Kampung Sapi *Adventure.*
2. Penguji menggunakan fungsi laporan keuangan dengan prosedur uji dan nilai masukan (*input*) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "valid" apabila keluaran sesuai dengan harapan dan diisi dengan "tidak valid" apabila keluaran tidak sesuai harapan.

Kode Pengujian	PV – 12
Nomor Skenario	Skenario 7
Kode Persyaratan	KF – 08
Tujuan Penelitian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melihat laporan keuangan
Prosedur Uji	1. Penguji teridentifikasi sebagai administrasi Kampung Sapi *Adventure.* 2. Penguji memilih menu "Laporan Keuangan".
Hasil yang diharapkan	Sistem menampilkan halaman laporan keuangan
Keluaran	1. Keluaran sesuai dengan hasil yang diharapkan 2. Keluaran tidak sesuai yang diharapkan
Status Uji	VALID

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Petunjuk Pengisian

1. Penguji adalah administrasi Wisata Kampung Sapi *Adventure*.
2. Penguji menggunakan fungsi cetak laporan keuangan dengan prosedur uji dan nilai masukan (*input*) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "valid" apabila keluaran sesuai dengan harapan dan diisi dengan "tidak valid" apabila keluaran tidak sesuai harapan.

Kode Pengujian	PV – 12
Nomor Skenario	Skenario 7
Kode Persyaratan	KF – 09
Tujuan Penelitian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk mengunduh laporan keuangan
Prosedur Uji	1. Penguji teridentifikasi sebagai administrasi Kampung Sapi *Adventure*. 2. Penguji memilih menu "Laporan Keuangan". 3. Penguji menekan tombol print 4. Laporan keuangan terunduh
Hasil yang diharapkan	Sistem menjalankan fungsi mencetak laporan keuangan
Keluaran	1. Keluaran sesuai dengan hasil yang diharapkan 2. Keluaran tidak sesuai yang diharapkan
Status Uji	VALID.

Kuesioner Pengujian Validasi kepada Wisatawan Wisata Kampung Sapi *Adventure*

Pengguna : Wisatawan

Nama : FILDZA

Umur : 21

Pekerjaan : MAHASISWA

Petunjuk Pengisian

1. Penguji adalah wisatawan Wisata Kampung Sapi *Adventure*.
2. Penguji menggunakan fungsi register dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. *Untuk jawaban hasil pengujian diisi dengan "valid" apabila keluaran sesuai dengan* harapan dan diisi dengan "tidak valid" apabila keluaran tidak sesuai harapan.

Kode Pengujian	PV – 01
Nomor Skenario	Skenario 1
Kode Persyaratan	KF – 01
Tujuan Penelitian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan register
Prosedur Uji	1. Penguji teridentifikasi sebagai wisatawan Kampung Sapi *Adventure*. 2. Penguji memilih menu register. 3. Penguji memasukkan *username*: "vedo, email: "vedoraku@gmail.com", dan *password*: "a123". 4. Penguji menekan tombol register. 5. Penguji melakukan verifikasi email.
Hasil yang diharapkan	Sistem menampilkan halaman *login*
Keluaran	1. *Keluaran sesuai dengan hasil yang diharapkan* 2. Keluaran tidak sesuai yang diharapkan
Status Uji	VALID

Batu, 7 Desember 2019

Petunjuk Pengisian

1. Penguji adalah wisatawan Wisata Kampung Sapi *Adventure*.
2. Penguji menggunakan fungsi *login* dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "valid" apabila keluaran sesuai dengan harapan dan diisi dengan "tidak valid" apabila keluaran tidak sesuai harapan.

Kode Pengujian	PV – 02
Nomor Skenario	Skenario 2
Kode Persyaratan	KF – 02
Tujuan Penelitian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan *login*
Prosedur Uji	1. Penguji teridentifikasi sebagai wisatawan Kampung Sapi *Adventure*. 2. Penguji memilih menu *login*. 3. Penguji memasukkan *username*: "vedo" dan *password*: "a123". 4. Penguji menekan tombol *login*.
Hasil yang diharapkan	Sistem menampilkan halaman utama wisatawan
Keluaran	1. Keluaran sesuai dengan hasil yang diharapkan 2. Keluaran tidak sesuai yang diharapkan
Status Uji	VALID

Kode Pengujian	PV – 02
Nomor Skenario	Skenario 2.1
Kode Persyaratan	KF – 02
Tujuan Penelitian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan *login*
Prosedur Uji	1. Penguji teridentifikasi sebagai wisatawan Kampung Sapi *Adventure*. 2. Penguji memilih menu *login*. 3. Penguji memasukkan *username*: "vedo" dan *password*: "vedo". 4. Penguji menggunakan menekan tombol *login*.
Hasil yang diharapkan	Sistem menampilkan username or password invalid
Keluaran	1. Keluaran sesuai dengan hasil yang diharapkan 2. Keluaran tidak sesuai yang diharapkan
Status Uji	VALID

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Petunjuk Pengisian

1. Penguji adalah wisatawan Wisata Kampung Sapi *Adventure*.
2. Penguji menggunakan fungsi *logout* dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "valid" apabila keluaran sesuai dengan harapan dan diisi dengan "tidak valid" apabila keluaran tidak sesuai harapan.

Kode Pengujian	PV – 03
Nomor Skenario	Skenario 3
Kode Persyaratan	KF – 03
Tujuan Penelitian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan *logout*
Prosedur Uji	1. *Penguji teridentifikasi sebagai wisatawan Kampung Sapi Adventure.* 2. Penguji telah berhasil masuk ke dalam sistem. 3. Penguji menekan tombol *logout*.
Hasil yang diharapkan	Sistem menampilkan halaman utama aplikasi
Keluaran	1. Keluaran sesuai dengan hasil yang diharapkan 2. Keluaran tidak sesuai yang diharapkan
Status Uji	VALID

Petunjuk Pengisian

1. Penguji adalah wisatawan Wisata Kampung Sapi *Adventure*.
2. Penguji menggunakan fungsi kegiatan dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "valid" apabila keluaran sesuai dengan harapan dan diisi dengan "tidak valid" apabila keluaran tidak sesuai harapan.

Kode Pengujian	PV – 04
Nomor Skenario	Skenario 4
Kode Persyaratan	KF – 04
Tujuan Penelitian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melihat kegiatan
Prosedur Uji	1. *Penguji teridentifikasi sebagai wisatawan Kampung Sapi Adventure.* 2. Penguji menggunakan memilih menu kegiatan.
Hasil yang diharapkan	Sistem menampilkan data kegiatan
Keluaran	1. Keluaran sesuai dengan hasil yang diharapkan 2. Keluaran tidak sesuai yang diharapkan
Status Uji	VALID

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Petunjuk Pengisian

1. Penguji adalah wisatawan Wisata Kampung Sapi *Adventure.*
2. Penguji menggunakan fungsi pesan tiket dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "valid" apabila keluaran sesuai dengan harapan dan diisi dengan "tidak valid" apabila keluaran tidak sesuai harapan.

Kode Pengujian	PV – 05
Nomor Skenario	Skenario 5
Kode Persyaratan	KF – 05
Tujuan Penelitian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan pemesanan tiket
	1. Penguji teridentifikasi sebagai wisatawan Kampung Sapi *Adventure.* 2. Penguji memilih menu pesan. 3. Penguji memilih kategori tiket. 4. Penguji mengetahui fasilitas dan kegiatan paket sesuai kategori tiket. 5. Penguji mengetahui harga paket sesuai kategori tiket. 6. Penguji dapat langsung mengetahui ketersediaan tiket jika memilih kategori tiket go show. 7. Penguji melakukan cek tanggal pada kategori tiket reservasi ketika ingin melihat ketersediaan tiket. 8. Penguji melakukan pembelian tiket dengan menekan tombol beli. 9. Penguji menginputkan jumlah tiket yang akan dibeli. 10. Penguji menekan tombol selanjutnya. 11. Penguji mengetahui detail pembelian tiket. 12. Penguji menekan tombol lanjutkan pembayaran. 13. Penguji melakukan pembayaran pada midtrans. 14. Penguji memilih metode pembayaran pada midtrans. 15. Penguji memlilih metode pembayaran melalui transfer bank. 16. Penguji melihat jumlah yang akan dibayar beserta order id dan cara pembayaran 17. Penguji menekan tombol lihat no rekening 18. Penguji melakukan pembayaran ke no rekening tujuan 19. Penguji memilih menu tiket 20. Penguji dapat melihat status pebayaran tiket pada menu tiket 21. Penguji dapat melakukan cetak tiket ketika status pembayaran berhasil

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Kode Pengujian	PV – 05
Nomor Skenario	Skenario 5
Kode Persyaratan	KF – 05
Tujuan Penelitian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan pemesanan tiket
Hasil yang diharapkan	Sistem memberikan elektronik tiket
Keluaran	1. Keluaran sesuai dengan hasil yang diharapkan 2. Keluaran tidak sesuai yang diharapkan
Status Uji	VALID