

**RANCANG BANGUN APLIKASI PEMESANAN TIKET  
DI KOLAM RENANG FAILDA SUKANAGARA  
CIANJUR SELATAN**

**TUGAS AKHIR**

Disusun sebagai salah satu syarat untuk kelulusan Program Strata 1  
di Program Studi Teknik Informatika Universitas Pasundan Bandung

oleh:

Muhamad Faisal Ilhami Akbar  
NRP : 15.304.0044



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS PASUNDAN BANDUNG  
JULI 2019**



**LEMBAR PENGESAHAN  
LAPORAN TUGAS AKHIR**

Telah diujikan dan dipertahankan dalam Sidang Sarjana Program Studi Teknik Informatika Universitas Pasundan Bandung, pada hari dan tanggal sidang sesuai berita acara sidang, tugas akhir dari:

Nama : Muhamad Faisal Ilhami Akbar  
Nrp : 15.304.0044

Dengan judul:

**“RANCANG BANGUN APLIKASI PEMESANAN TIKET  
DI KOLAM RENANG FAILDA SUKANAGARA  
CIANJUR SELATAN”**

Bandung, 11 Juli 2019

Menyetujui,

Pembimbing Utama,

Pembimbing Pendamping,

(R. Sandhika Galih A., S.T., M.T.)

(Wanda Gusdy, S.T., M.T.)



## **LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR**

Saya menyatakan dengan sesungguhnya bahwa :

1. Tugas akhir ini adalah benar-benar asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Universitas Pasundan Bandung maupun di Perguruan Tinggi lainnya
2. Tugas akhir ini merupakan gagasan, rumusan dan penelitian saya sendiri, tanpa bantuan pihak lain kecuali arahan dari tim Dosen Pembimbing
3. Dalam tugas akhir ini tidak terdapat karya atau pendapat orang lain, kecuali bagian-bagian tertentu dalam penulisan laporan Tugas Akhir yang saya kutip dari hasil karya orang lain telah dituliskan dalam sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan karya ilmiah, serta disebutkan dalam Daftar Pustaka pada tugas akhir ini
4. Kakas, perangkat lunak, dan alat bantu kerja lainnya yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab saya, bukan tanggung jawab Universitas Pasundan Bandung

Apabila di kemudian hari ditemukan seluruh atau sebagian laporan tugas akhir ini bukan hasil karya saya sendiri atau adanya plagiasi dalam bagian-bagian tertentu, saya bersedia menerima sanksi akademik, termasuk pencabutan gelar akademik yang saya sandang sesuai dengan norma yang berlaku di Universitas Pasundan, serta perundang-undangan lainnya.

Bandung, 11 Juli 2019

Yang membuat pernyataan

Materai  
6000,-

**( Muhamad Faisal Ilhami Akbar )**

NRP. 15.304.0044



## **ABSTRAK**

Kemajuan teknologi berkembang pesat dari tahun ke tahun dengan membawa banyak dampak perubahan contohnya dalam hal pemesanan tiket. Dengan adanya dampak tersebut banyak yang memanfaat teknologi internet untuk melakukan pemesanan tiket secara online, salah satunya adalah Kolam renang Failda. Saat ini, untuk melakukan pemesanan tiket customer harus datang langsung ke tempat untuk memesan tiket. Selanjutnya, tiket tersebut harus ditarik kembali oleh petugas untuk keperluan rekapitulasi data tiket. Cara yang dilakukan oleh pihak pengelola kolam renang Failda saat ini, mengakibatkan seringnya ketidak sesuaian antara tiket yang dibeli pengunjung dan uang yang masuk, yang berdampak sulitnya melakukan rekapitulasi data tiket karena tiket bisa rusak ataupun hilang.

Munculah gagasan untuk membuat aplikasi pemesanan tiket berbasis *mobile* dan aplikasi pengelolaan data tiket berbasis web. Metode yang digunakan dalam pembuatan aplikasi ini adalah *Waterfall*. Bahasa pemrograman yang digunakan untuk membuat web service dan aplikasi pengelolaan tiket berbasis web adalah PHP dengan menggunakan *framework* Laravel. *Mobile app* menggunakan platform Android dengan native Java. Database menggunakan DBMS MySQL.

Berdasarkan hasil implementasi yang dilakukan, aplikasi pemesanan tiket berbasis *mobile* dan aplikasi pengelolaan data tiket berbasis web telah sesuai dengan kebutuhan, karena dibangun berdasarkan identifikasi kebutuhan dan melakukan pemeriksaan terhadap kebutuhan tersebut kepada narasumber yang dilakukan wawancara.

Kata kunci : Aplikasi Pemesanan Tiket, Aplikasi Pengelolaan Data Tiket, *Mobile App*, *Web App*, Pemesanan.



## **ABSTRACT**

Technological progress is growing rapidly from year to year by bringing many changes to the example of ticket bookings. With this impact, many are utilizing internet technology to order tickets online, one of which is the Failda Swimming Pool. Currently, the customer to order tickets should come directly to the point of booking the ticket. Furthermore, the ticket must be withdrawn by the officer for the purpose of recapitulating ticket data. The way in which the pool manager Failda is currently doing, results in frequent discrepancies between tickets purchased by visitors and money coming in, which has the effect of making it difficult to recapitulate ticket data because tickets can be damaged or lost.

The idea was to create a mobile-based ticket booking application and a web-based ticket data management application. The method used in making this application is Waterfall. The programming language used to create web services and web-based ticket management applications is PHP using the Laravel framework. The mobile app uses the Android platform with native Java. The database uses the MySQL DBMS.

Based on the results of the implementation is done, the application of mobile-based ticket reservation and ticket applications web-based data management in accordance with the needs, because it was built based on the identification of needs and conduct an examination of the need for trainers who conducted the interview.

**Keywords:** Ticket Booking Application, Ticket Data Management Application, Mobile App, Web App, Booking.



## **KATA PENGANTAR**

Ucapan dan rasa syukur penulis layangkan ke hadirat Ilahi Robbi, yang telah berkenan menguatkan penulis untuk membuat Laporan Tugas Akhir dengan judul “Rancang Bangun Aplikasi Pemesanan Tiket Di Kolam Renang Failda Sukanagara Cianjur Selatan”.

Adapun penulisan laporan ini bertujuan untuk memenuhi salah satu syarat kelulusan Program Strata 1, di Program Studi Teknik Informatika Universitas Pasundan.

Penulis menyadari laporan ini dapat terwujud berkat bantuan dan dorongan dari berbagai pihak. Maka pada kesempatan ini penulis sampaikan terima kasih yang sebesar-besarnya atas segala bantuan yang penulis terima baik secara moril maupun materil, sehingga penulis dapat menyelesaikan laporan ini kepada :

1. Kepada Orang Tua tersayang, dan keluarga yang selalu memberikan motivasi serta do'anya dalam pembuatan tugas akhir ini.
2. Kepada pembimbing utama, R. Sandhika Galih Amalga, S.T, M.T
3. Kepada pembimbing pendamping, Wanda Gusdy, S.T, M.T
4. Koordinator Tugas Akhir dan Ketua Kelompok Keilmuan serta seluruh civitas akademika Teknik Informatika di UNIVERSITAS PASUNDAN BANDUNG, yang telah memberikan bekal ilmu selama penulis menimba ilmu.
5. Kepada Annysa Eka Yusilawati yang telah membantu dan memberi dukungan untuk menyelesaikan laporan ini.
6. Kepada teman-teman seperjuangan Universitas Pasundan Bandung yang tidak bisa semua penulis sebutkan.

Tiada gading yang tak retak, tiada gelombang tanpa ombak, segala kesalahan merupakan kelemahan dan kekurangan penulis. oleh karena itu, penulis harapkan kritik dan saran dari semua pihak demi perbaikan di masa yang akan datang.

Akhir kata, semoga penulisan laporan ini dapat bermanfaat bagi penulis dan bagi perkembangan ilmu Teknologi dimasa yang akan datang.

Bandung, 11 Juli 2019

Penulis



## DAFTAR ISI

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR .....	i
ABSTRAK .....	ii
ABSTRACT .....	iii
KATA PENGANTAR .....	v
DAFTAR ISI.....	vii
DAFTAR ISTILAH .....	xi
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR .....	xv
DAFTAR LAMPIRAN .....	xix
DAFTAR SIMBOL.....	xxi
BAB 1 PENDAHULUAN.....	1-1
1.1 Latar Belakang Tugas Akhir .....	1-1
1.2 Identifikasi Masalah .....	1-2
1.3 Tujuan Tugas Akhir .....	1-2
1.4 Lingkup Tugas Akhir .....	1-2
1.5 Metodologi Penyelesaian Tugas Akhir .....	1-2
1.6 Sistematika Penulisan Tugas Akhir.....	1-4
BAB 2 LANDASAN TEORI .....	2-1
2.1 Teori yang digunakan.....	2-1
2.1.1 Tiket .....	2-1
2.1.2 Tiket Online ( <i>E-Ticket</i> ) .....	2-1
2.1.2.1 Manfaat E-Ticket.....	2-1
2.1.2.2 Penerapan E-Ticketing .....	2-2
2.1.2.3 Contoh Aplikasi Pemesanan Tiket .....	2-2
2.1.3 <i>Waterfall</i> .....	2-3
2.1.4 <i>Web</i> .....	2-4
2.1.5 <i>Model View Controller (MVC)</i> .....	2-5
2.1.5.1 <i>Model</i> .....	2-5
2.1.5.2 <i>View</i> .....	2-5
2.1.5.3 <i>Controller</i> .....	2-5
2.1.6 <i>Web Framework Application</i> .....	2-6
2.1.6.1 Kelebihan <i>Web Application Framework</i> .....	2-6
2.1.6.2 Contoh <i>Web Application Framework</i> .....	2-6
2.1.7 <i>Web Service</i> .....	2-6

2.1.8	<i>Restfull</i> .....	2-7
2.1.9	<i>Application Programming Interface (API)</i> .....	2-8
2.1.10	<i>Mobile Application</i> .....	2-8
2.2	Penelitian Terdahulu.....	2-8
BAB 3	<b>SKEMA PENELITIAN</b> .....	3-1
3.1	Rancangan Penelitian .....	3-1
3.2	Analisis Masalah Dan Solusi TA.....	3-4
3.3	Kerangka Pemikiran Teoritis.....	3-5
3.3.1	Rancangan Analisis .....	3-5
3.3.2	Kerangka Pemikiran Teoritis.....	3-7
3.4	Analisis Teknologi.....	3-7
3.4.1	Asynchronous Javascript and XML (AJAX).....	3-7
3.4.2	Laravel.....	3-8
3.4.2.1	Fitur-Fitur Laravel .....	3-8
3.4.2.2	Kelebihan Laravel.....	3-8
3.4.3	Android.....	3-9
3.4.3.1	Android SDK.....	3-9
3.4.3.2	Arsitektur Aplikasi Berbasis Android .....	3-10
3.4.4	Retrofit.....	3-11
3.4.5	Guzzle.....	3-11
3.4.6	JSON.....	3-12
3.4.7	QR Code .....	3-13
3.5	Analisis Penggunaan Konsep .....	3-14
3.5.1	<i>Web Service</i> .....	3-14
3.5.2	Aplikasi Web.....	3-14
3.5.3	Aplikasi <i>Mobile</i> .....	3-15
3.6	Analisis Aplikasi Serupa .....	3-15
3.7	Profile Penelitian .....	3-16
3.7.1	Profil Objek Penelitian .....	3-16
3.7.2	Profil Tempat Penelitian.....	3-17
3.7.2.1	Sejarah Singkat .....	3-17
3.7.2.2	Struktur Organisasi .....	3-17
BAB 4	<b>PERENCANAAN DAN PEMODELAN</b> .....	4-1
4.1	<i>Current System</i> (Sistem saat ini) .....	4-1
4.2	Analisis Kebutuhan .....	4-1

4.2.1	Menentukan Kebutuhan Dasar .....	4-1
4.2.1.1	Analisis Stakeholder.....	4-1
4.2.1.2	Analisis Fungsional .....	4-2
4.2.2	Analisis Pengguna .....	4-2
4.2.3	Analisis Ranah.....	4-3
4.2.4	Pendekatan Untuk Pemodelan Spesifikasi Kebutuhan.....	4-3
4.2.4.1	UML ( <i>Unified Modelling Process</i> ) .....	4-4
4.2.4.2	Elemen-Elemen Model Analisis.....	4-4
4.2.5	Pemodelan Berbasis Skenario .....	4-5
4.2.5.1	Deskripsi <i>Use Case</i> .....	4-6
4.2.5.2	Skenario <i>Use Case</i> .....	4-6
4.2.5.3	Aliran Aktivitas .....	4-11
4.2.6	Pemodelan Data.....	4-20
4.2.6.1	Objek Data.....	4-20
4.2.6.2	Atribut Data.....	4-20
4.2.6.3	Relasi.....	4-22
4.2.7	Pemodelan Berbasis Kelas .....	4-23
4.2.7.1	Mengidentifikasi Kelas-Kelas Analisis .....	4-24
4.2.7.2	Menentukan Atribut-Atribut.....	4-24
4.2.7.3	Mendefinisikan Operasi-Operasi .....	4-28
4.2.7.4	Perancangan Kelas .....	4-30
4.2.8	Pemodelan Berbasis Perilaku .....	4-31
4.3	Perancangan Perangkat Lunak .....	4-39
4.3.1	Perancangan Data.....	4-39
4.3.2	Perancangan Arsitektural .....	4-41
4.3.2.1	Gaya Arsitektural .....	4-41
4.3.2.2	Arsitektur Teknologi .....	4-42
4.3.3	Perancangan Antarmuka Pengguna.....	4-42
BAB 5	KONSTRUKSI DAN DEPLOYMENT .....	5-1
5.1	Konstruksi Perangkat Lunak .....	5-1
5.1.1	Kebutuhan Perangkat dan Kakas Untuk Pengembangan .....	5-1
5.1.2	Pengkodean .....	5-2
5.1.2.1	Tahap Pengkodean Aplikasi Web dan Web Service .....	5-2
5.1.3	Implementasi Antarmuka Pengguna .....	5-16
5.1.4	Pengujian Perangkat Lunak.....	5-24
5.1.4.1	Pengujian Aplikasi Web.....	5-24
5.1.4.2	Pengujian Web Service .....	5-25

5.1.4.3 Pengujian Aplikasi <i>Mobile</i> .....	5-26
5.2 Deployment .....	5-27
5.2.1 Spesifikasi Server .....	5-27
5.2.2 Konfigurasi .....	5-27
BAB 6 KESIMPULAN DAN SARAN .....	6-1
6.1 Kesimpulan.....	6-1
6.2 Saran.....	6-1
6.3 Rekomendasi .....	6-1

## DAFTAR PUSTAKA

## DAFTAR ISTILAH

Table 1 Daftar Istilah

No	Istilah	Definisi
1.	<i>Web</i>	<i>Web</i> adalah kumpulan informasi, text, gambar, video serta animasi yang dihubungkan dengan jaringan-jaringan ( <i>hyperlink</i> ).
2.	<i>Mobile</i>	<i>Mobile</i> adalah kata sifat yang berarti dapat bergerak atau dapat digerakkan dengan bebas dan mudah.
3.	<i>Framework</i>	<i>Framework</i> adalah sebuah software untuk memudahkan para programmer membuat aplikasi atau web yang isinya adalah berbagai fungsi, plugin, dan konsep sehingga membentuk suatu sistem tertentu.
4.	<i>Requirement</i>	<i>Requirement</i> adalah sebuah pendokumentasian dokumen fisik dan kebutuhan fungsional yang didesain secara tertentu, produk atau proses yang harus dilakukan.
5.	<i>Server</i>	<i>Server</i> adalah sebuah sistem komputer yang menyediakan jenis layanan (service) tertentu dalam sebuah jaringan computer.
6.	<i>Client</i>	<i>Client</i> adalah komputer dalam jaringan yang menggunakan sumber daya yang disediakan oleh server.



## DAFTAR TABEL

Table 1 Daftar Istilah .....	xi
Table 1 Simbol Diagram Swimlane .....	xxi
Table 2 Simbol Diagram Use Case .....	xxi
Table 3 Simbol Diagram Kelas .....	xxi
Table 4 Simbol Diagram Sekuen .....	xxi
Table 5 Simbol Entity Relationship Diagram .....	xxii
Table 2.1 Penelitian Terdahulu .....	2-9
Table 3.1 Alur Penyelesaian Tugas Akhir .....	3-1
Table 3.2 Analisis Manfaat Aplikasi Pemesanan Tiket di Kolam Renang Failda .....	3-5
Table 3.3 Rancangan Analisis .....	3-5
Table 3.4 Penjelasan Rancangan Analisis .....	3-6
Table 4.1 Daftar Stakeholder .....	4-1
Table 4.2 Analisis Fungsional .....	4-2
Table 4.3 Daftar Pengguna .....	4-3
Table 4.4 Deskripsi Hasil Output Analisis Ranah .....	4-3
Table 4.5 Deskripsi Use Case .....	4-6
Table 4.6 Skenario Use Case Mengelola Pemesanan Tiket .....	4-7
Table 4.7 Skenario Use Case Mengupdate Pemesanan Tiket .....	4-7
Table 4.8 Skenario Use Case Mencetak Tiket .....	4-7
Table 4.9 Skenario Use Case Mengelola Konfirmasi Pembayaran .....	4-8
Table 4.10 Skenario Use Case Menampilkan Notifikasi Pemesanan Tiket .....	4-8
Table 4.11 Skenario Use Case Memverifikasi Pemesanan Tiket .....	4-9
Table 4.12 Skenario Use Case Form Konfirmasi Pembayaran .....	4-9
Table 4.13 Skenario Use Case Mengelola Akun Karyawan .....	4-10
Table 4.14 Skenario Use Case Menginput Akun Customer .....	4-10
Table 4.15 Skenario Use Case Mengelola Content Berita .....	4-10
Table 4.16 Skenario Use Case Menampilkan Rekap Data Penjualan Tiket .....	4-11
Table 4.17 Objek Data .....	4-20
Table 4.18 Atribut Data .....	4-21
Table 4.19 Relasi Antar Objek .....	4-23
Table 4.20 Kelas-Kelas Analisis .....	4-24
Table 4.21 Atribut-Atribut Kelas .....	4-25
Table 4.22 Perilaku Pada Setiap Kelas .....	4-28
Table 4.23 Perancangan Data .....	4-39
Table 5.1 Spesifikasi Laptop .....	5-1

Table 5.2 Software yang digunakan dalam pengembangan .....	5-1
Table 5.3 Daftar Controller Aplikasi Web .....	5-3
Table 5.4 Daftar Controller Web Service.....	5-6
Table 5.5 Daftar Model Web Service dan Aplikasi Web .....	5-9
Table 5.6 Daftar Presenter Aplikasi Mobile .....	5-14
Table 5.7 Spesifikasi Server .....	5-27

## DAFTAR GAMBAR

Gambar 1.1 Metodologi Penyelesaian Tugas Akhir .....	1-4
Gambar 2.1 Pemesanan tiket pesawat terbang di tiket.com [GTN19] .....	2-2
Gambar 2.2 Pemesanan tiket wahana rekreasi di aplikasi Digitiket [DIG19].....	2-3
Gambar 2.3 Pemesanan tiket bioskop di aplikasi tix id [NES19] .....	2-3
Gambar 2.4 Model Proses Waterfall [PRE10] .....	2-4
Gambar 2.5 Model View Controller (MVC) [PHI17].....	2-5
Gambar 3.1 Kerangka Permasalahan .....	3-4
Gambar 3.2 Kerangka Pemikiran Teoritis.....	3-7
Gambar 3.3 Arsitektur Android [ZPJ16].....	3-11
Gambar 3.4 Bentuk Object Pada JSON [JSO00] .....	3-12
Gambar 3.5 Bentuk Array Pada JSON [JSON00].....	3-13
Gambar 3.6 Bentuk Value Pada JSON [JSO00] .....	3-13
Gambar 3.7 QR Code [IBG16] .....	3-14
Gambar 3.8 Logo Perusahaan .....	3-17
Gambar 3.9 Struktur Organisasi.....	3-18
Gambar 4.1 Analisis Ranah [PRE10].....	4-3
Gambar 4.2 Elemen Model Analisis [PRE10] .....	4-4
Gambar 4.3 Diagram <i>Use Case</i> Pengelolaan Tiket.....	4-5
Gambar 4.4 Use Case Pemesanan Tiket Berbasis Mobile .....	4-5
Gambar 4.5 Diagram <i>Swimlane</i> pada use case UC-01 untuk aktor kasir .....	4-12
Gambar 4.6 Diagram <i>Swimlane</i> pada use case UC-01 untuk aktor customer .....	4-13
Gambar 4.7 Diagram <i>Swimlane</i> pada use case UC-02 aktor admin .....	4-13
Gambar 4.8 Diagram <i>Swimlane</i> pada use case UC-03 .....	4-14
Gambar 4.9 Diagram <i>Swimlane</i> pada use case UC-04 .....	4-14
Gambar 4.10 Diagram <i>Swimlane</i> pada use case UC-05 .....	4-15
Gambar 4.11 Diagram <i>Swimlane</i> pada use case UC-06.....	4-15
Gambar 4.12 Diagram <i>Swimlane</i> pada use case UC-07 .....	4-16
Gambar 4.13 Diagram <i>Swimlane</i> pada use case UC-08 .....	4-17
Gambar 4.14 Diagram <i>Swimlane</i> pada use case UC-09 .....	4-18
Gambar 4.15 Diagram <i>Swimlane</i> pada use case UC-10 .....	4-19
Gambar 4.16 Diagram <i>Swimlane</i> pada use case UC-11 .....	4-19
Gambar 4.17 Diagram Kelas .....	4-30
Gambar 4.18 Diagram Sekuen menginput pemesanan tiket aktor kasir .....	4-31
Gambar 4.19 Diagram Sekuen menginput pemesanan tiket aktor customer.....	4-31
Gambar 4.20 Diagram Sekuen mengupdate pemesanan tiket.....	4-32

Gambar 4.21 Diagram Sekuen mencetak tiket .....	4-32
Gambar 4.22 Diagram Sekuen mengelola konfirmasi pembayaran (update) .....	4-33
Gambar 4.23 Diagram Sekuen mengelola konfirmasi pembayaran (delete) .....	4-33
Gambar 4.24 Diagram Sekuen menampilkan notifikasi pemesanan .....	4-34
Gambar 4.25 Diagram Sekuen memverifikasi pemesanan tiket.....	4-34
Gambar 4.26 Diagram Sekuen menginput konfirmasi pembayaran.....	4-35
Gambar 4.27 Mengelola akun karyawan (tambah).....	4-35
Gambar 4.28 Mengelola akun karyawan (update).....	4-36
Gambar 4.29 Mengelola akun karyawan (delete).....	4-36
Gambar 4.30 Menginput akun customer .....	4-37
Gambar 4.31 Mengelola content berita (tambah) .....	4-37
Gambar 4.32 Mengelola content berita (update) .....	4-38
Gambar 4.33 Mengelola content berita (delete) .....	4-38
Gambar 4.34 Menampilkan data rekap penjualan tiket .....	4-39
Gambar 4.35 Gaya Arsitektural Berpusat Pada Data .....	4-41
Gambar 4.36 Arsitektur Teknologi Web .....	4-41
Gambar 4.37 Arsitektur Teknologi Mobile .....	4-41
Gambar 4.38 Arsitektur Teknologi.....	4-42
Gambar 4.39 Antarmuka pemesanan tiket berbasis web.....	4-42
Gambar 4.40 Antarmuka tambah dan update data pemesanan tiket berbasis web .....	4-43
Gambar 4.41 Antarmuka konfirmasi pembayaran berbasis web .....	4-43
Gambar 4.42 Antarmuka update status pembayaran tiket berbasis web .....	4-44
Gambar 4.43 Antarmuka notifikasi pembayaran tiket berbasis web .....	4-44
Gambar 4.44 Antarmuka kelola pengguna berbasis web .....	4-45
Gambar 4.45 Antarmuka tambah dan update kelola pengguna berbasis web .....	4-45
Gambar 4.46 Antarmuka halaman cetak tiket berbasis web.....	4-46
Gambar 4.47 Antarmuka halaman scan QR Code berbasis mobile.....	4-46
Gambar 4.48 Antarmuka halaman registrasi aplikasi <i>mobile</i> .....	4-47
Gambar 4.49 Antarmuka halaman awal aplikasi berbasis mobile.....	4-47
Gambar 4.50 Antarmuka sidebar aplikasi berbasis mobile .....	4-48
Gambar 4.51 Antarmuka halaman pemesanan tiket aplikasi mobile.....	4-48
Gambar 4.52 Antarmuka halaman konfirmasi pembayaran aplikasi mobile.....	4-49
Gambar 4.53 Antarmuka halaman bukti pembayaran .....	4-49
Gambar 4.54 Antarmuka halaman list pemesanan tiket .....	4-50
Gambar 5.1 Struktur Directori Aplikasi Web dan Web Service .....	5-2
Gambar 5.2 Controller UserController.....	5-3
Gambar 5.3 Controller KonfirmasiPembayaranController.....	5-4

Gambar 5.4 Controller PemesananController .....	5-5
Gambar 5.5 Controller KonfigurasiMobileController.....	5-6
Gambar 5.6 Controller Web Service PemesananController.....	5-7
Gambar 5.7 Controller Web Service PembayaranController .....	5-7
Gambar 5.8 Controller Web Service PemesananController.....	5-8
Gambar 5.9 Controller Web Service KonfigurasiMobileController .....	5-8
Gambar 5.10 Model UserModel.....	5-9
Gambar 5.11 Model KaryawanModel.....	5-10
Gambar 5.12 Model KonfirmasiPembayaranModel .....	5-10
Gambar 5.13 Model PemesananModel .....	5-11
Gambar 5.14 Model CustomerModel.....	5-11
Gambar 5.15 List View Aplikasi Web .....	5-12
Gambar 5.16 List Route Aplikasi Web .....	5-12
Gambar 5.17 Struktur Directori Aplikasi Mobile .....	5-13
Gambar 5.18 Koneksi ke Web Service .....	5-13
Gambar 5.19 Presenter DetailPemesananActivity .....	5-14
Gambar 5.20 Presenter PembayaranActivity .....	5-15
Gambar 5.21 Presenter RegisterActivity .....	5-15
Gambar 5.22 PresenterScanActivity .....	5-16
Gambar 5.23 Implementasi antarmuka pemesanan tiket berbasis web .....	5-16
Gambar 5.24 Implementasi Antarmuka tambah dan update pemesanan tiket berbasis web.....	5-17
Gambar 5.25 Implementasi Antarmuka konfirmasi pembayaran berbasis web.....	5-17
Gambar 5.26 Implementasi Antarmuka update status pembayaran tiket berbasis web .....	5-18
Gambar 5.27 Implementasi Antarmuka menampilkan notifikasi pembayaran tiket berbasis web ....	5-18
Gambar 5.28 Implementasi Antarmuka kelola pengguna berbasis web .....	5-19
Gambar 5.29 Implementasi Antarmuka tambah dan update kelola pengguna berbasis web .....	5-19
Gambar 5.30 Implementasi Antarmuka cetak tiket berbasis web .....	5-20
Gambar 5.31 Implementasi Antarmuka halaman scan QR Code mobile.....	5-20
Gambar 5.32 Implementasi Antarmuka halaman registrasi aplikasi mobile.....	5-21
Gambar 5.33 Implementasi Antarmuka halaman awal aplikasi berbasis mobile.....	5-21
Gambar 5.34 Implemetnasi Antarmuka sidebar aplikasi berbasis mobile .....	5-22
Gambar 5.35 Implementasi Antarmuka halaman pemesanan tiket aplikasi mobile.....	5-22
Gambar 5.36 Implementasi Antarmuka halaman konfirmasi pembayaran aplikasi mobile.....	5-23
Gambar 5.37 Implementasi Antarmuka halaman bukti pembayaran .....	5-23
Gambar 5.38 Implementasi Antarmuka halaman list pemesanan tiket .....	5-24



## **DAFTAR LAMPIRAN**

LAMPIRAN 1 BERITA ACARA WAWANCARA .....	1-1
LAMPIRAN 2 LINK REPOSITORY APLIKASI .....	B-1
LAMPIRAN 3 TABLE PENGUJIAN PERANGKAT LUNAK .....	C-1



## DAFTAR SIMBOL

Table 0.1 Simbol Diagram *Swimlane*

No.	Simbol	Nama Simbol	Deskripsi
1.		Start State	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Activity	Suatu himpunan aliran kerja dalam organisasi yang mempunyai nilai terhadap business actor.
3.		Decision	Asosiasi pencabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
5.		End State	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

Table 0.2 Simbol Diagram *Use Case*

No.	Simbol	Nama Simbol	Deskripsi
1.		Use Case	Use case merepresentasikan spesifik penggunaan sistem oleh aktor. Use case menspesifikasi perilaku sistem atau bagian sistem dan deskripsi sekumpulan aksi termasuk varian - varian yang dilakukan sistem untuk menghasilkan nilai atau informasi.
2.		Aktor	Mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan use case.
3.		Association	Relasi yang menjelaskan bahwa aktor melakukan interaksi dua arah dimana aktor memberikan sesuatu kepada use case dan menerima sesuatu dari business use case, begitupun sebaliknya.

Table 0.3 Simbol Diagram Kelas

No.	Simbol	Nama Simbol	Deskripsi
1.		Class	Menggambarkan class yang akan dibangun di dalam sistem
2.		Association	Penghubung antara kelas satu dengan kelas yang lainnya.

Table 0.4 Simbol Diagram Sekuen

No.	Simbol	Nama Simbol	Deskripsi
1.		Boundary	Menggambarkan sebuah penggambaran dari form.

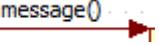
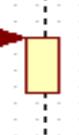
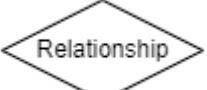
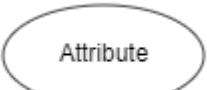
No.	Simbol	Nama Simbol	Deskripsi
2.		Aktor	Mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan use case.
3.		Control	Menggambarkan penghubung antar boundary dan entity
4.		Entity	Menggambarkan hubungan kegiatan yang akan dilakukan.
5.		Message	Menggambarkan pengiriman pesan.
6.		Focus of Control and Life Line	Menggambarkan tempat mulai dan berakhirnya sebuah message.

Table 0.5 Simbol *Entity Relationship Diagram*

No.	Simbol	Nama Simbol	Deskripsi
1.		Entitas	Entitas digunakan sebagai objek yang dapat diidentifikasi dalam lingkungan pemakai.
2.		Relasi	Relasi digunakan untuk menunjukkan adanya hubungan antar entitas.
3.		Atribut	Atribut menunjukkan field yang dimiliki oleh setiap entitas.

## **BAB 1**

### **PENDAHULUAN**

Bab ini berisi penjelasan umum mengenai usulan penelitian yang dilakukan dalam pengerjaan tugas akhir. Di dalamnya berisi latar belakang tugas akhir, identifikasi masalah, tujuan tugas akhir, lingkup tugas akhir, metodologi yang digunakan dalam pengerjaan tugas akhir, dan sistematika penulisan laporan tugas akhir.

#### **1.1 Latar Belakang Tugas Akhir**

Kolam renang Failda merupakan kolam renang yang terletak di Sukanagara kabupaten Cianjur Provinsi Jawa Barat yang penulis pilih untuk melakukan penelitian. Kolam renang ini sudah terkenal hampir di seluruh daerah sekitar Cianjur Selatan. Dimana kolam renang ini dijadikan sebagai objek wisata untuk berlibur, berekreasi ataupun untuk berolahraga. Saat ini kolam renang Failda bekerja sama dengan sekolah-sekolah yang ada disana mulai dari SD – SMA untuk para guru olahraga melakukan praktik renang untuk siswa-siswinya.

Berdasarkan wawancara dengan pihak pengelola kolam renang Failda. Saat ini, proses pemesanan tiket di kolam renang Failda masih menggunakan tiket kertas yang dibeli oleh pengunjung. Tiket tersebut harus ditarik kembali oleh petugas untuk keperluan rekapitulasi data tiket yang dibeli pengunjung dan uang yang masuk setiap harinya. Cara yang dilakukan oleh pihak pengelola kolam renang Failda saat ini, mengakibatkan seringnya ketidak sesuaian antara tiket yang dibeli pengunjung dan uang yang masuk, yang berdampak sulitnya melakukan rekapitulasi data tiket karena tiket bisa rusak ataupun hilang. Oleh karena itu, dibutuhkan solusi teknologi untuk mengurangi dampak dari masalah-masalah tersebut. [Lampiran 1]

Seiring dengan semakin pesatnya kemajuan dalam dunia teknologi saat ini, kebutuhan akan kemudahan dalam mengakses dan memanfaatkan teknologi informasi dalam memenuhi kebutuhan dalam masyarakat semakin meningkat, salah satunya adalah pemanfaatan media internet (online) yang digunakan untuk memperoleh informasi. Dengan adanya peningkatan jumlah pengguna internet di Indonesia yang cukup signifikan dari tahun ke tahun membuat pelaku bisnis memanfaatkan internet dalam upaya meningkatkan pelayanannya [HMJ17]. Salah satu contohnya adalah melakukan usaha dibidang pemesanan tiket secara online, saat ini sudah banyak aplikasi yang digunakan untuk memesan tiket secara online.

Untuk menyelesaikan masalah yang ada di kolam renang Failda, dibutuhkan beberapa aplikasi yang harus diterapkan. Aplikasi yang akan diterapkan yaitu aplikasi berbasis *mobile*, dimana aplikasi ini digunakan untuk pemesanan tiket secara online yang nantinya bisa diunduh di play store. Penggunaan aplikasi ini diharapkan untuk mempermudah pengunjung dalam melakukan pemesanan tiket kolam renang. Sedangkan untuk pengelolaan data tiket akan dibuat menggunakan aplikasi berbasis *web* yang

dibangun menggunakan *framework* Laravel, dimana aplikasi ini akan mengelola semua data tiket yang telah dibeli pengunjung baik secara langsung maupun online. Selanjutnya untuk pertukaran data antara aplikasi *mobile* akan menggunakan teknologi *web service*.

Berdasarkan dari latar belakang yang telah diuraikan diatas, maka penulis tertarik untuk membuat tugas akhir dengan judul “Rancang Bangun Aplikasi Pemesanan Tiket Di Kolam Renang Failda Sukanagara Cianjur Selatan”.

## 1.2 Identifikasi Masalah

Berdasarkan latarbelakang yang telah dipaparkan sebelumnya, maka permasalahan yang dimunculkan pada tugas akhir ini adalah :

1. Bagaimana membuat aplikasi mobile yang dapat melakukan pemesanan tiket ?
2. Bagaimana membuat aplikasi web untuk mengelola data pemesanan tiket dan melakukan rekapitulasi data tiket ?
3. Bagaimana menerapkan teknologi web service sebagai teknologi pertukaran data pada aplikasi *mobile* ?

## 1.3 Tujuan Tugas Akhir

Tujuan tugas akhir ini adalah :

1. Menghasilkan aplikasi berbasis web yang memiliki kemampuan untuk mengelola data pemesanan tiket di kolam renang Failda.
2. Menghasilkan aplikasi berbasis mobile yang digunakan untuk pemesanan tiket secara online.
3. Membuat *Web Service* sebagai teknologi pertukaran data aplikasi *mobile*.

## 1.4 Lingkup Tugas Akhir

Penyelesaian Tugas Akhir dibatasi sebagai berikut :

1. Bahasa pemrograman yang di gunakan dalam pembangunan aplikasi adalah PHP dengan memakai *framework* Laravel untuk membuat web service dan aplikasi berbasis web serta Java untuk aplikasi berbasis Android.
2. Aplikasi mobile dibuat hanya untuk perangkat Android.

## 1.5 Metodologi Penyelesaian Tugas Akhir

Metodologi tugas akhir ini merupakan serangkaian kegiatan yang dilakukan dalam upaya menyelesaikan tugas akhir ini. Berikut ini merupakan penjelasan mengenai langkah penyelesaian tugas akhir :

## 1. Identifikasi Masalah

Pada tahap ini dilakukan identifikasi masalah untuk mengetahui masalah apa saja yang ada di tempat penelitian.

## 2. Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data yang relevan secara teoritis dari sumber yang jelas.

### a. Studi Literatur

Pada tahap ini dilakukan pencarian dan perbandingan referensi yang didapat dari buku, jurnal ilmiah dalam bentuk cetak maupun e-book di internet untuk mendapatkan teori yang relevan dengan masalah yang sudah diidentifikasi serta dengan tujuan dari tugas akhir ini.

### b. Wawancara

Pada tahap ini dilakukan wawancara secara langsung terhadap permasalahan yang terdapat pada kolam renang Failda.

## 3. Pembangunan Perangkat Lunak

### a. Komunikasi

Dalam tahap ini dilakukan komunikasi untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut.

### b. Perancangan

Perancangan merupakan kegiatan untuk menentukan jadwal pembuatan perangkat lunak dan memperkirakan harga dari perangkat lunak serta perkiraan-perkiraan lainnya.

### c. Pemodelan

Pemodelan merupakan kegiatan untuk merepresentasikan hasil dari komunikasi ke dalam bentuk diagram-digram untuk keperluan perancangan perangkat lunak yang akan dibangun.

### d. Konstruksi

Dalam tahap ini dilakukan pembangunan aplikasi dan pengujian berdasarkan hasil tahapan pemodelan yang sebelumnya telah dilakukan.

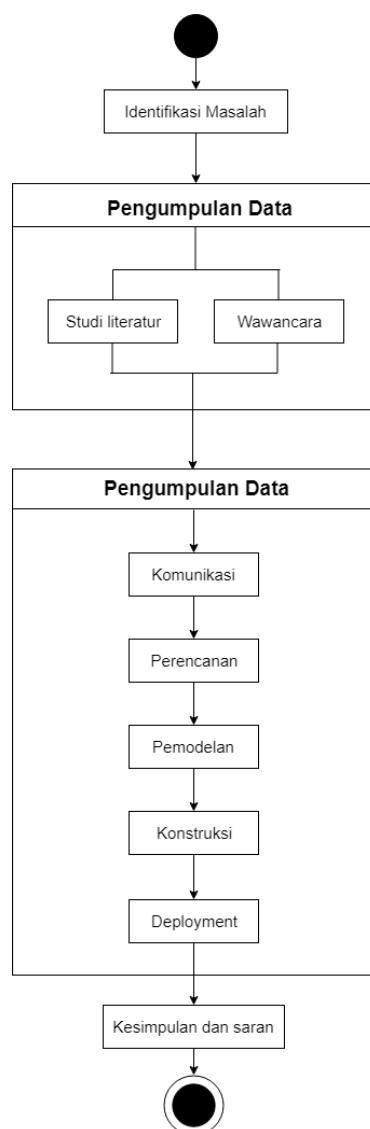
### e. Deployment

Dalam tahap ini perangkat lunak diserahkan kepada pengguna.

## 4. Kesimpulan Dan Saran

Dalam tahap ini dilakukan penyimpulan dari penelitian yang telah dilakukan terkait dengan masalah yang telah diidentifikasi, serta saran sebagai prospek penelitian selanjutnya.

Berikut ini adalah gambar untuk metodologi penyelesaian tugas akhir, seperti yang dapat dilihat pada Gambar 1.1.



Gambar 1.1 Metodologi Penyelesaian Tugas Akhir

## 1.6 Sistematika Penulisan Tugas Akhir

Sistematika penulisan tugas akhir ini disusun untuk memberikan gambaran umum tentang penelitian yang dilakukan. Sistematika penulisan tugas akhir ini antara lain :

### BAB 1 PENDAHULUAN

Bab ini berisi penjelasan umum mengenai usulan penelitian yang dilakukan dalam pengerjaan tugas akhir. Di dalamnya berisi latar belakang masalah, identifikasi masalah, rumusan masalah, tujuan tugas akhir, lingkup tugas akhir, metodologi pengerjaan tugas akhir, dan sistematika penulisan tugas akhir.

### BAB 2 LANDASAN TEORI

Bab ini berisi definisi-definisi, teori-teori, serta konsep-konsep dasar yang diperlukan untuk menganalisis situasi yang diteliti yang didapat dari buku-buku teks ataupun makalah-makalah di jurnal-jurnal ilmiah yang terkait yang relevan sebagai referensi pengerjaan tugas akhir ini. Di dalam bab ini dikemukakan hasil-hasil penelitian terdahulu.

**BAB 3 SKEMA PENELITIAN**

Bab ini berisi kerangka pelaksanaan tugas akhir, rancangan penelitian, analisis masalah dan manfaat tugas akhir, kerangka pemikiran teoritis, analisis penggunaan konsep, dan profil perusahaan.

**BAB 4 PERENCANAAN DAN PEMODELAN**

Bab ini berisi mengenai analisis kebutuhan aplikasi dan perancangan aplikasi pemesanan tiket di kolam renang Failda berdasarkan kebutuhan aplikasi yang telah dipaparkan.

**BAB 5 KONSTRUKSI DAN DEPLOYMENT**

Bab ini berisi cara pengimplementasian hasil analisis dan perancangan pada bab sebelumnya dengan proses pengkodean aplikasi dan pengujian aplikasi yang akan dibuat serta menjelaskan cara deploy aplikasi ke server dan ke *client*.

**BAB 6 KESIMPULAN DAN SARAN**

Bab ini berisi mengenai hasil penelitian serta pernyataan yang didapat berdasarkan identifikasi masalah yang dikemukakan, serta keterkaitan dari semua tahap yang dilakukan dalam penelitian. Di dalamnya terdapat pula saran yang diusulkan untuk penelitian selanjutnya.



## **BAB 2**

### **LANDASAN TEORI**

Dalam bab ini membahas beberapa teori yang mendasari penulisan dari tugas akhir ini. Teori umum yang digunakan meliputi teori mengenai konsep pemesanan tiket, teori pengembangan perangkat lunak, dan beberapa teori lainnya yang mendasari penyusunan tugas akhir ini.

#### **2.1 Teori yang digunakan**

##### **2.1.1 Tiket**

Tiket merupakan suatu voucher untuk menunjukkan bahwa telah membayar pintu masuk suatu teater, taman hiburan, kebun binatang, konser dan bioskop [ADA17].

Tiket adalah suatu dokumen perjalanan yang dikeluarkan oleh suatu perusahaan yang berisi rute, tanggal, harga, data penumpang yang digunakan untuk melakukan suatu perjalanan [TAN08].

Selembar kertas atau kartu kecil yang diberikan kepada seseorang, biasanya untuk menunjukkan bahwa mereka telah membayar untuk suatu acara, perjalanan, atau kegiatan [CAM19]:

1. Tiket konser
2. Tiket kereta / bus / pesawat
3. Tiket lotre / undian
4. Kantor tiket
5. Pengumpul tiket

##### **2.1.2 Tiket Online (*E-Ticket*)**

*E-Ticketing* atau electronic ticketing adalah suatu cara untuk mendokumentasikan proses penjualan dari aktifitas perjalanan pelanggan tanpa harus mengeluarkan dokumen berharga secara fisik ataupun paper ticket [GNG06].

*E-Ticketing* (ET) adalah peluang untuk meminimalkan biaya dan mengoptimalkan kenyamanan penumpang. *E-Ticketing* mengurangi biaya proses tiket, menghilangkan fomulir kertas dan meningkatkan fleksibilitas penumpang dan agen perjalanan dalam membuat perubahan perubahan dalam jadwal perjalanan [NKY08].

##### **2.1.2.1 Manfaat E-Ticket**

Saat ini hampir semua maskapai penerbangan menjual tiket dengan *E-ticketing*. *E-Ticketing* menyediakan banyak manfaat diantaranya [NKY08] :

1. Mengurangi biaya yang terkait dengan pencetakan dan mailing tiket ke pembeli. Menghilangkan atau mengurangi memerlukan tiket untuk stok, amplop dan pos.
2. Mengurangi tenaga kerja yang terkait dengan pencetakan dan mailing tiket.

3. Keamanan terjamin, karena barcode validasi menghilangkan kemungkinan data palsu dan duplikat tiket.
4. Pemesanan *e-ticketing* oleh konsumen berarti mengetahui berapa banyak konsumen perusahaan, karena perusahaan menyimpan data konsumen di database perusahaan.
5. Memberikan informasi tambahan seperti jam keberangkatan, tempat berkumpul dan informasi lain yang pelanggan mungkin dianggap perlu tahu.
6. Menyediakan kemampuan unik periklanan. Meningkatkan pendapatan perusahaan dengan menawarkan ruang iklan pada web tiket maskapai penerbangan.

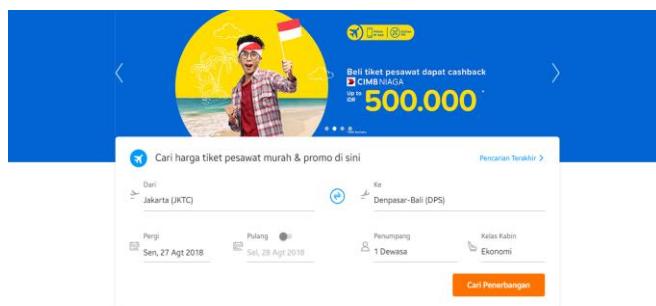
### 2.1.2.2 Penerapan E-Ticketing

Dalam proses *E-Ticketing* terdapat beberapa langkah yang dapat dilakukan guna mempermudah kita dalam pemesanan tiket secara online dimana dengan *E-Ticketing* tidak perlu repot dalam memesan tiket. Langkah-langkah *E-Ticketing* sangat praktis, reservasi adalah yang paling utama. Sebagai pengguna *mobile*, ada beberapa pilihan yang tersedia bagi konsumen. Menelepon *Call Center* maskapai penerbangan yang dipilih. Langkah berikutnya adalah pembayaran. Dengan mengutamakan kemudahan, dapat melakukannya dengan ATM, serta *credit card*. *Call Center* yang menjamin keamanan saat memasukan nomor *credit card*. Terjamin semua informasi akan disimpan dengan rapih dan aman. Jika tertinggal tanda terima perjalanan, dapat meminta duplikatnya di seluruh kantor perjualan tiket maskapai penerbangan (airport) [NKY08].

### 2.1.2.3 Contoh Aplikasi Pemesanan Tiket

Berikut ini adalah beberapa contoh aplikasi pemesanan tiket.

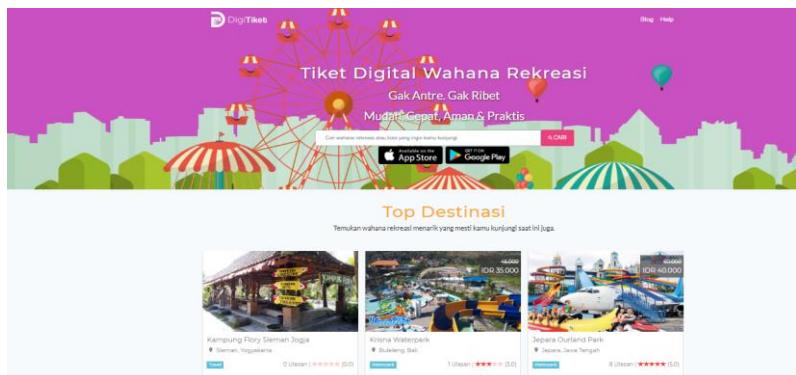
1. Aplikasi pemesanan tiket pesawat terbang di tiket.com



Gambar 2.1 Pemesanan tiket pesawat terbang di tiket.com [GTN19]

Pemesanan tiket pesawat terbang di tiket.com bisa melalui website atau melalui aplikasi mobile. Dimana pengguna bisa memilih rute penerbangan yang akan tuju, kemudian memilih maskapai penerbangan. Selanjutnya pengguna akan menerima rincian pemesanan untuk dilanjutkan ke pembayaran. Setelah melakukan pembayaran, pengguna akan menerima e-tiket yang dipesan melalui SMS dan Email.

## 2. Aplikasi pemesanan tiket wahana rekreasi di aplikasi Digitiket



Gambar 2.2 Pemesanan tiket wahana rekreasi di aplikasi Digitiket [DIG19]

Pemesanan wahana rekreasi ini bisa melalui aplikasi Digitiket. Dimana pengguna bisa mencari wahana rekreasi yang akan dituju. Kemudian memilih tanggal untuk datang ke wahana tersebut dan menentukan jumlah tiket yang dipesan. Selanjutnya pengguna akan menerima rincian pemesanan dan diarahkan ke menu pembayaran. Setelah melakukan pembayaran, pengguna akan menerima e-tiket yang dikirim ke aplikasi Digitiket.

## 3. Aplikasi pemesanan tiket bioskop di aplikasi tix id



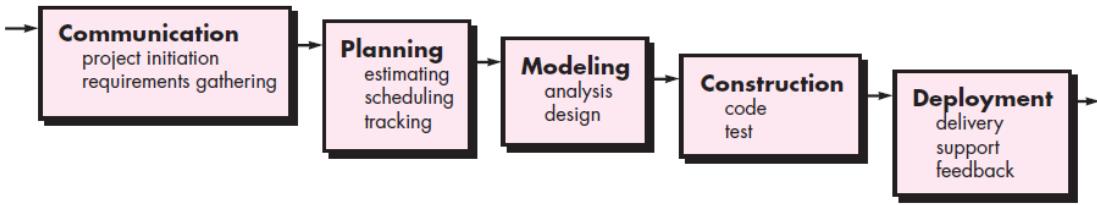
Gambar 2.3 Pemesanan tiket bioskop di aplikasi tix id [NES19]

Pemesanan tiket bioskop xxi bisa melalui aplikasi Tix id. Dimana pengguna bisa memilih kota, kemudian memilih film yang akan ditonton. Selanjutnya pengguna memilih tempat bioskop dan jadwal film yang tersedia. Kemudian memilih tempat duduk dan setelah itu akan muncul rincian pembelian tiket dan pengguna harus segera membayar tiket tersebut. Setelah melakukan pembayaran, pengguna akan menerima e-tiket yang dipesan aplikasi Tix id.

### 2.1.3 *Waterfall*

Model *waterfall* yaitu pendekatan yang sistematis dan berurutan pada pengembangan perangkat lunak, yang dimulai dengan spesifikasi kebutuhan pengguna dan berlanjut melalui tahapan-tahapan perencanaan (*planning*), pemodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem/

perangkat lunak ke para pelanggan/pengguna (*deployment*), yang diakhiri dengan dukungan berkelanjutan pada perangkat lunak lengkap yang dihasilkan [PRE10].



Gambar 2.4 Model Proses Waterfall [PRE10]

Tahap dari skema model waterfall adalah [PRE10]:

#### 1. Communication

Langkah ini merupakan analisis terhadap kebutuhan software, dan tahap untuk mengadakan pengumpulan data dengan melakukan pertemuan dengan customer, maupun mengumpulkan data-data tambahan baik yang ada di jurnal, artikel, maupun dari internet.

#### 2. Planning

Proses planning merupakan lanjutan dari proses communication (analysis requirement). Tahapan ini akan menghasilkan dokumen user requirement atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan software, termasuk rencana yang akan dilakukan.

#### 3. Modeling

Proses modeling ini akan menerjemahkan syarat kebutuhan ke sebuah perancangan software yang dapat diperkirakan sebelum dibuat coding. Proses ini berfokus pada rancangan struktur data, arsitektur software, representasi interface, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut software requirement.

#### 4. Construction

Construction merupakan proses membuat kode. Coding atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. Programmer akan menerjemahkan transaksi yang diminta oleh user. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu software, artinya penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan menemukan kesalahan-kesalahan terhadap sistem untuk kemudian bisa diperbaiki.

#### 5. Deployment

Tahapan ini bisa dikatakan final dalam pembuatan sebuah software atau sistem. Setelah melakukan analisis, desain dan pengkodean maka sistem yang sudah jadi akan digunakan oleh user. Kemudian software yang telah dibuat harus dilakukan pemeliharaan secara berkala.

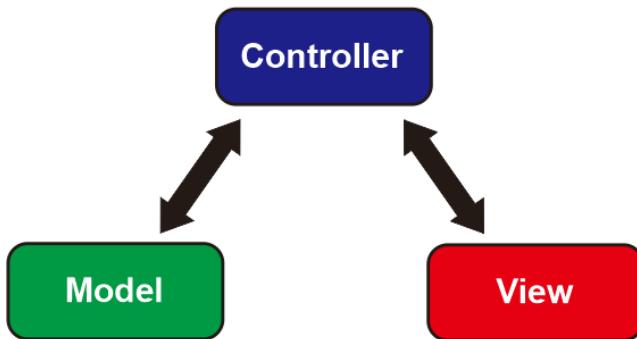
### 2.1.4 Web

*Web* dapat diartikan sebagai kumpulan halaman halaman yang digunakan untuk menampilkan informasi, teks, gambar diam atau bergerak, animasi, suara, dan atau gabungan dari semuanya itu, baik

yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling berkait dimana masing masing dihubungkan dengan jaringan jaringan halaman (*hyperlink*) [SSU04].

### 2.1.5 *Model View Controller* (MVC)

*Model View Controller* (MVC) adalah suatu metode yang memisahkan data logic (*Model*) dari presentation logic (*View*) dan process logic (*Controller*) atau secara sederhana adalah memisahkan antara desain interface, data, dan process [PHI17].



Gambar 2.5 Model View Controller (MVC) [PHI17]

#### 2.1.5.1 *Model*

*Model* mengelola basis data (RDBMS) seperti MySQL ataupun yang lainnya. Model menghubungkan dengan database sehingga biasanya dalam model akan berisi class ataupun fungsi untuk membuat (create), melakukan pembaharuan (update), menghapus data (delete), mencari data (search), dan mengambil data (select) pada database [PHI17].

#### 2.1.5.2 *View*

*View* adalah bagian user interface atau bagian yang nantinya merupakan tampilan untuk end user. *View* bisa berupa halaman html, css, rss, javascript, jquery, ajax, dll. Karena metode yang dipakai merupakan MVC ataupun pengaksesan yang berhubungan dengan database. Sehingga view hanya menampilkan data-data hasil dari model dan controller [PHI17].

#### 2.1.5.3 *Controller*

*Controller* adalah penghubung antara view dan model, maksudnya adalah karena model tidak dapat berhubungan langsung dengan view ataupun sebaliknya., controller inilah yang digunakan sebagai jembatan dikeduanya. Sehingga tugas controller adalah sebagai pemrosesan data atau alur logic program, menyediakan variable yang akan ditampilkan di view, pemanggilan model sehingga model dapat mengakses database, error handling validasi atau check terhadap suatu inputan [PHI17]..

### **2.1.6 *Web Framework Application***

Suatu kumpulan kode berupa pustaka (*library*) dan alat (*tool*) yang dipadukan sedemikian rupa menjadi satu kerangka kerja (*framework*) guna memudahkan dan mempercepat proses pengembangan aplikasi *web* [RAH15].

#### **2.1.6.1 *Kelebihan Web Application Framework***

Berikut ini adalah kelebihan dari *Web Application Framework*, diantaranya sebagai berikut :

1. Dapat membantu developer/programmer dalam membangun aplikasi
2. Penerapan Design Pattern memudahkan dalam rancangan, Pengembangan dan pemeliharaan sistem.
3. Stability dan Reliability aplikasi yang kita bangun lebih stabil dan handal.
4. Coding Style Consistence, memudahkan dalam hal membaca source code karena cara pengkodean yang sama/konsisten.
5. Security Concern Framework, mengantisipasi dan memasang perisai terhadap adanya berbagai masalah.
6. Dokumentasi, framework dapat mendisiplinkan kita menulis dokumentasi untuk sistem apa yang kita bangun.

#### **2.1.6.2 *Contoh Web Application Framework***

Dibawah ini adalah contoh *framework-framework* berbasis web [PHI17] :

1. Codeigniter
2. Yii
3. Zend Framework
4. Ruby on Rails
5. Laravel
6. CakePHP
7. Django
8. Symfony
9. FuelPHP
10. PhalconPHP

### **2.1.7 *Web Service***

*Web service* adalah sebuah antarmuka yang mendeskripsikan sekumpulan operasi yang dapat diakses dalam sebuah jaringan melalui pesan *XML* yang telah distandardkan [KRE01].

*Web service* merupakan komponen perangkat lunak yang *loosely coupled*, dapat digunakan ulang, membungkus fungsionalitas diskret, didistribusikan, dan dapat diakses secara pragmatik melalui protokol internet standar [THE01].

*Web service* adalah sebuah sistem perangkat lunak yang dirancang untuk mendukung inter operasi mesin ke mesin di sebuah jaringan [W3C04].

*Web service* adalah teknologi yang mengubah kemampuan internet dengan menambahkan kemampuan *transactional web*, yaitu kemampuan web untuk saling berkomunikasi dengan pola *program-to-program* (P2P). Fokus web selama ini didominasi oleh komunikasi *program-to-user* dengan interaksi *business-to-consumer* (B2C), sedangkan *transactional web* akan didominasi oleh *program-to-program* dengan interaksi *business-to-business* [HDE11].

elayanan web memiliki dua teknik popular, yaitu SOAP (Simple Object Access Protocol) dan RESTfull. Dalam layanan berbasis SOAP, klien mengirimkan pesan SOAP kelayanan dan layanan merespon dalam jenis pesan SOAP juga. RESTfull adalah salah satu yang menggunakan HTTP tidak hanya sebagai infrastruktur pertukaran informasi, tetapi juga sebagai sebuah pedoman untuk merancang permintaan layanan dan tanggapan dari layanan. Dalam layanan RESTfull, HTTP itu sendiri dapat diketahui sebagai suatu API. SOAP memiliki standar, toolkit, dan perpustakaan *software* yang melimpah. REST tidak memiliki standard resmi, relatif sedikit toolkit, dan pustaka *software* yang tidak merata antara bahasa pemrograman, namun ada dukungan yang terus ditingkatkan untuk layanan REST di seluruh bahasa pemograman [MKA13].

### **2.1.8 Restfull**

Representational State Transfer (REST) merupakan salah satu gaya arsitektur dari pengembangan API yang menggunakan Hypertext Transfer Protocol (HTTP) dalam melakukan komunikasi data [APB18].

Salah satu kriteria desain web services yang paling sering digunakan adalah restful web services, restful sendiri bekerja dengan cara resource-oriented. Pada restfull web services *client* (requester) mengakses services yang ditawarkan oleh web server, yaitu dengan cara mengakses url dari resource menggunakan method pada http [LRA13].

Dalam dunia web API, protokolnya adalah http. API *client* dapat berinteraksi dengan API dengan mengirimkan berbagai jenis pesan http. Standar http mendefinisikan delapan jenis pesan, yaitu :

1. GET

Method Get mengambil data dari web server dengan menentukan parameter di bagian url dari permintaan.

2. DELETE

Method Delete menghapus sumber daya

3. POST

Method Post memanfaatkan badan pesan untuk mengirim data ke server web.

4. PUT

Method Put mirip dengan post memanfaatkan badan pesan untuk mentransfer data.

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

*Application Programming Interface* merupakan sebuah *interface* yang diimplementasikan dengan menggunakan *software* sehingga *software* tersebut dapat berinteraksi dengan *software* lain, seperti halnya *user interface* yang memungkinkan *user* untuk berinteraksi dengan komputer [AEP18].

*Application Programming Interface* (API) adalah antarmuka yang dibangun oleh pengembang sistem sehingga beberapa atau seluruh fungsi sistem dapat diakses secara programatik [APB18].

### **2.1.10 Mobile Application**

*Mobile Application* adalah jenis perangkat lunak aplikasi yang dirancang untuk dijalankan di perangkat seluler, seperti komputer smartphone atau tablet. *Mobile application* sering berfungsi untuk menyediakan layanan serupa bagi pengguna yang diakses di PC [TIN18].

Berikut ini adalah karakteristik Mobile Application yaitu [ZPJ16] :

1. Ukuran yang kecil

Perangkat mobile memiliki ukuran yang kecil. Konsumen menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.

2. Memory yang terbatas

Perangkat mobile juga memiliki memory yang kecil, yaitu primary (RAM) dan secondary (disk).

3. Daya proses yang terbatas

Sistem mobile tidaklah setangguh rekan mereka yaitu desktop

4. Mengkonsumsi daya yang rendah

Perangkat mobile menghabiskan sedikit daya dibandingkan dengan mesin desktop

5. Kuat dan dapat diandalkan

Karena perangkat mobile selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapi benturan-benturan, gerakan, dan sesekali tetesan-tetesan air.

6. Konektivitas yang terbatas

Perangkat mobile memiliki bandwith rendah, beberapa dari mereka bahkan tidak tersambung.

7. Masa hidup yang pendek

Perangkat-perangkat konsumen ini menyala dalam hitungan detik kebanyakan dari mereka selalu menyala.

## **2.2 Penelitian Terdahulu**

Ada beberapa penelitian terdahulu yang terkait dengan tugas akhir ini yang dapat dilihat pada Table 2.1.

Table 2.1 Penelitian Terdahulu

No	Peneliti	Hasil	Alasan digunakan
1.	Saputra, Adit M.,2017 [MAS17]	Membangun aplikasi pemesanan tiket bus berbasis android untuk mempermudah pemesanan.	Karena menjelaskan tentang cara pembuatan aplikasi pemesanan tiket berbasis android.
2.	Mutmainah, 2016 [MUT16]	Membangun aplikasi pemesanan tiket kapal laut berbasis web untuk mempermudah orang-orang mendapatkan pemesanan tiket.	Karena menjelaskan tentang cara menyampaikan informasi pemesanan tiket.
3.	Salamun, Faerul, 2017 [FSA17]	Membangun aplikasi web, mobile, dan web service untuk Nmax Community.	Untuk memahami cara membuat aplikasi web, mobile, dan web service
4.	Mubarak, Husni, 2017 [HMU17]	Menerapkan proses pengelolaan nilai siswa berbasis web mulai dari kompetensi dasar hingga pembuatan rapor	Untuk memahami pembangunan aplikasi berbasis web menggunakan Codeigniter
5.	Hidayat Endang, 2015 [EHI15]	Membuat sebuah web service untuk e-voting pencatatan hingga perhitungan suara dalam pemilu raya HMTIF UNPAS	Untuk memahami bagaimana cara membuat web service yang digunakan oleh perangkat lain yang terhubung
6.	Akbar, Nurul Andri, 2017 [ANA10]	Menghasilkan perangkat lunak berbasis android untuk keperluan layanan warga di lingkungan Rukun Warga (RW)	Untuk memahami bagaimana cara-cara membuat perangkat lunak berbasis android



## **BAB 3**

### **SKEMA PENELITIAN**

Bab ini berisi alur penyelesaian tugas akhir, analisis persoalan, manfaat tugas akhir kerangka pemikiran teoritis, dan profil objek tempat penelitian.

#### **3.1 Rancangan Penelitian**

Rancangan Penelitian ini memiliki tujuan untuk memudahkan dalam memahami secara singkat mengenai tahapan pengerjaan dan literatur yang digunakan dalam pengerjaan tugas akhir. Setiap tahapan yang dilakukan memiliki tujuan dan output, sebagaimana dijelaskan pada Table 3.1.

Table 3.1 Alur Penyelesaian Tugas Akhir

Tahap dan Hasil	Langkah Penelitian	Literatur dan Referensi
Tahap 1 : menentukan topik TA dan identifikasi masalah  Hasil : mendapatkan topik tugas akhir dan masalah yang telah diidentifikasi  Kontribusi : Berguna untuk pengumpulan data	<p style="text-align: center;">Menentukan topik TA</p> <p style="text-align: center;">↓</p> <p style="text-align: center;">Identifikasi masalah</p>	-
Tahap 2 : Studi literatur dan Pengumpulan data  Hasil : teori yang sudah dipahami dan kumpulan data yang didapat	<p style="text-align: center;">Mengumpulkan jurnal penelitian</p> <p style="text-align: center;">↓</p> <p style="text-align: center;">Memahami teori yang terkait</p>	<ol style="list-style-type: none"><li>1. Rancang Bangun Layanan Web (Web Service) Untuk Aplikasi Rekam Medis Praktik Pribadi Dokter [YRD17]</li><li>2. Panduan Lengkap Pemrograman Android [ZPJ16]</li></ol>
Kontribusi : Berguna untuk tahap pengumpulan data dan spesifikasi kebutuhan	<p style="text-align: center;">Melakukan observasi dan wawancara</p> <p style="text-align: center;">↓</p> <p style="text-align: center;">1</p>	<ol style="list-style-type: none"><li>3. Implementasi metode rational unified process pada mobile digital library [UEY15]</li><li>4. Pemrograman Web [PHI17]</li></ol>

Tahap dan Hasil	Langkah Penelitian	Literatur dan Referensi
<p>Tahap 3 : menerapkan rancangan penelitian</p> <p>Hasil : alur penyelesaian tugas akhir</p> <p>Kontribusi : Berguna untuk perancangan perangkat lunak</p>	<pre> graph TD     1{1} --&gt; R1[Rancangan penelitian]     R1 --&gt; R2[Membuat alur penyelesaian TA]     R2 --&gt; R3[Analisis Persoalan TA]     R3 --&gt; R4[Membuat solusi untuk TA]   </pre>	<p>5. Implementasi metode rational unified process pada mobile digital library [UEY15]</p> <p>6. Rancang Bangun Layanan Web (Web Service) Untuk Aplikasi Rekam Medis Praktik Pribadi Dokter [YRD17]</p> <p>7. Panduan Lengkap Pemrograman Android [ZPJ16]</p> <p>8. Pemrograman Web [PHI17]</p>
<p>Tahap 4 : Melakukan perancangan perangkat lunak</p> <p>Hasil : mendapatkan kebutuhan fungsional dan non-fungsional dan spesifikasi perangkat lunak</p> <p>Kontribusi : berguna untuk tahap implementasi (koding) dan testing</p>	<pre> graph TD     R4 --&gt; AK[Analisis Kebutuhan]     AK --&gt; M1[Membuat pemodelan berbasis skenario]     M1 --&gt; M2[Membuat pemodelan berbasis data]     M2 --&gt; 2{2}   </pre>	<p>9. Implementasi metode rational unified process pada mobile digital library [UEY15]</p>

Tahap dan Hasil	Langkah Penelitian	Literatur dan Referensi
	<pre> graph TD     2[Membuat pemodelan berbasis kelas] --&gt; 3[Membuat pemodelan perilaku (sequence diagram)]     3 --&gt; 4{Perancangan}     4 --&gt; 5[Membuat perancangan data]     5 --&gt; 6[Membuat perancangan Arsitektural]     6 --&gt; 7[Membuat perancangan antarmuka]   </pre>	10. Software Engineering: A Practitioner's Approach, Seventh Edition 7 [PRE10]
Tahap 5 : implementasi (koding) dan testing Hasil : aplikasi yang sudah jadi Kontribusi : berguna untuk tahap pengambilan kesimpulan tugas akhir	<pre> graph TD     8[Pembuatan aplikasi (koding)] --&gt; 9[Pengujian aplikasi]     9 --&gt; 10{3}   </pre>	11. Rancang Bangun Layanan Web (Web Service) Untuk Aplikasi Rekam Medis Praktik Pribadi Dokter [YRD17] 12. Panduan Lengkap Pemrograman Android [ZPJ16] 13. Pemrograman Web [PHI17] 14. Software Engineering: A Practitioner's Approach, Seventh Edition 7 [PRE10]

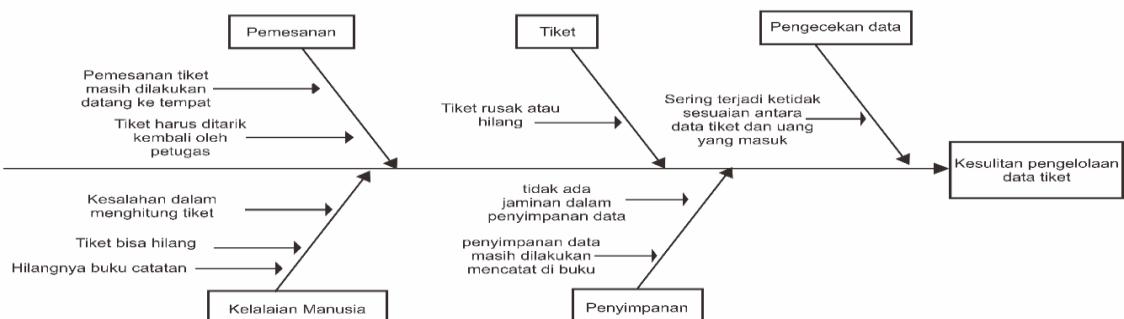
Tahap dan Hasil	Langkah Penelitian	Literatur dan Referensi
Tahap 6 : kesimpulan dan saran Hasil : kesimpulan dan saran tugas akhir Kontribusi : -	<p style="text-align: center;">3</p> <pre> graph TD     3[3] --&gt; Deployment[Deployment]     Deployment --&gt; Kesimpulan[Membuat kesimpulan dan saran]   </pre>	-

### 3.2 Analisis Masalah Dan Solusi TA

Analisis Masalah Dan Solusi TA bertujuan untuk mengetahui ketepatan solusi terhadap suatu permasalahan pada penelitian tugas akhir, yaitu pada aplikasi pemesanan tiket di kolam renang Failda, dengan memberikan sebuah solusi diharapkan dapat menyelesaikan permasalahan-permasalahan yang ada. Sehingga permasalahan yang telah didefinisikan dapat diselesaikan dengan solusi yang sesuai. Tahap analisis pada bagian ini menggunakan diagram sebab dan akibat (*cause and effect diagram*) atau biasa disebut dengan *Fishbone Diagram*. *Fishbone diagram* adalah alat untuk mengidentifikasi dan mengorganisir penyebab yang mungkin terjadi dari suatu masalah dalam format yang terstruktur [KEL95]. Terdapat 3 komponen utama pada *fishbone diagram*, yaitu sebagai berikut :

1. *Head* (Kepala), merupakan fokus masalah yang sedang diamati, dideskripsikan dalam kotak di bagian kepala pada diagram. *Head* terletak pada bagian kanan diagram.
2. *Spine* (Tulang Belakang), ditunjukkan dengan arah panah yang menunjukkan penyebab dari masalah yang terjadi pada bagian *head*.
3. *Bones* (Tulang atau Duri), menunjukkan kategori utama yang menyebabkan permasalahan terjadi. Jika terdapat *bones* yang lebih kecil, memperlihatkan kedalaman penyebab dari permasalahan tersebut. *Bones* yang saling terkoneksi memperlihatkan bahwa penyebab permasalahan tersebut saling berhubungan [KEL95].

Berikut merupakan Gambar 3.1 kerangka permasalahan untuk analisis peran aplikasi pemesanan tiket di kolam renang Failda.



Gambar 3.1 Kerangka Permasalahan

Berikut ini merupakan Tabel 3.2 hasil analisis manfaat aplikasi pemesanan tiket di kolam renang Failda. Terdapat masalah penyebab, dan solusi untuk menyelesaikan permasalahan di kolam renang Failda.

Table 3.2 Analisis Manfaat Aplikasi Pemesanan Tiket di Kolam Renang Failda

Masalah	Penyebab	Solusi
Kesulitan pengelolaan data tiket kolam renang Failda.	<b>Pemesanan</b> <ul style="list-style-type: none"> <li>Pemesanan tiket masih dilakukan datang ke tempat</li> <li>Tiket harus ditarik kembali oleh petugas</li> </ul> <b>Tiket</b> Tiket rusak atau hilang	Pembangunan Aplikasi Pemesanan Tiket di Kolam Renang Failda yang berguna untuk memudahkan melakukan pemesanan tiket dan memudahkan dalam mengelola data tiket.
	<b>Pengecekan Data</b> Sering terjadi ketidak sesuaian antara data tiket dan uang yang masuk	
	<b>Kelalaian Manusia</b> <ul style="list-style-type: none"> <li>Kesalahan dalam menghitung tiket</li> <li>Tiket bisa hilang</li> <li>Hilangnya buku catatan</li> </ul>	
	<b>Penyimpanan</b> <ul style="list-style-type: none"> <li>Tidak ada jaminan dalam penyimpanan data</li> <li>Penyimpanan data masih dilakukan mencatat di buku</li> </ul>	

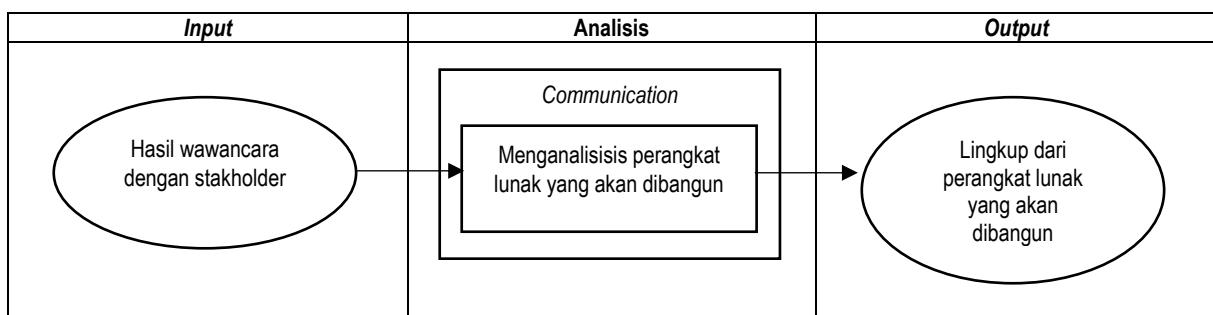
### 3.3 Kerangka Pemikiran Teoritis

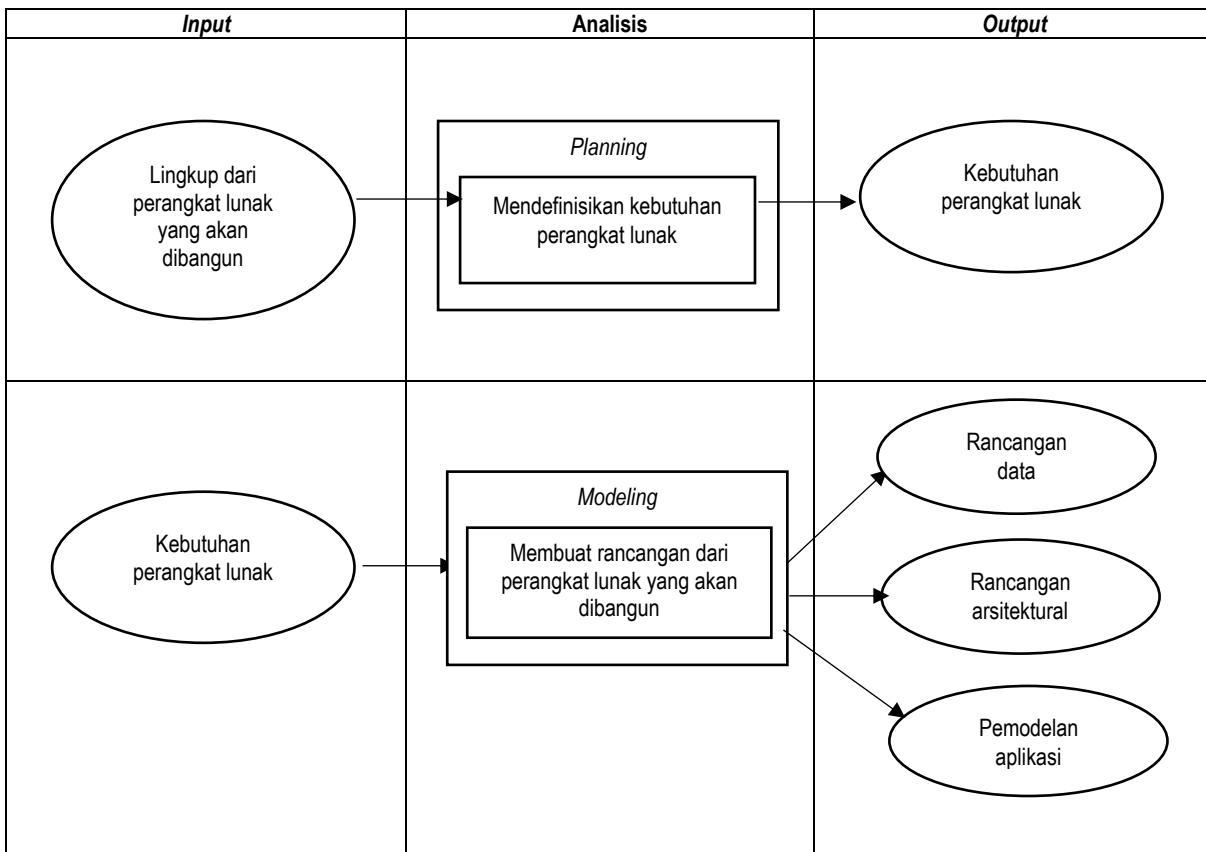
Pada tahap kerangka pemikiran teoritis ini akan dijelaskan mengenai tahapan yang dilakukan dalam perancangan aplikasi pemesanan tiket di kolam renang Failda. Tahap ini berisi mengenai kerangka pemikiran teoritis yang di dalamnya menunjukkan tentang komponen-komponen pembentuk produk tugas akhir yang disertai dengan kajian konsep dan teori-teori yang mendasarinya.

#### 3.3.1 Rancangan Analisis

Rancangan analisis merupakan gambaran dari proses-proses analisis yang dilakukan. Dalam penggambaran rancangan analisis, terdapat *input*, proses, dan *output*. *Input* merupakan data yang akan dimanfaatkan pada proses analisis yang akan menghasilkan *output*. Rancangan analisis dapat dilihat pada Table 3.3.

Table 3.3 Rancangan Analisis





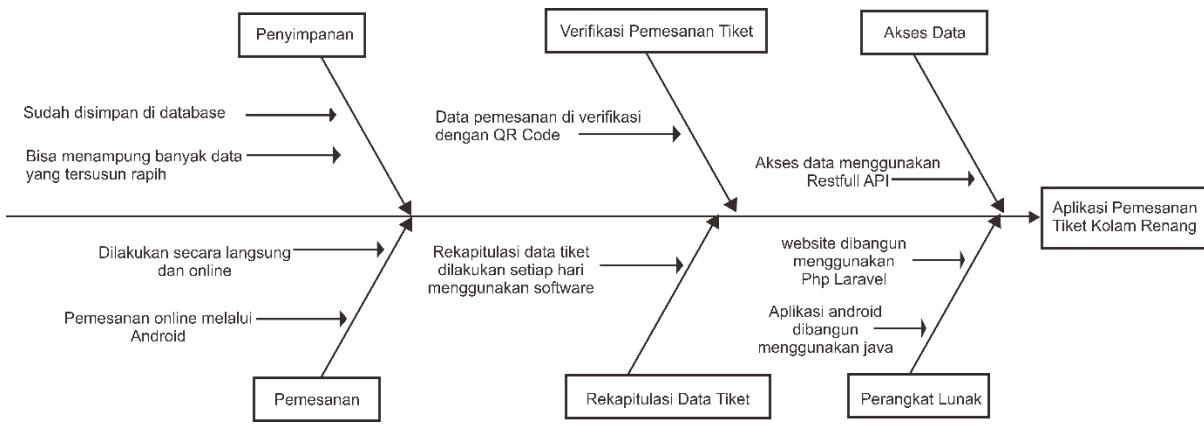
Rancangan analisis pada Table 3.3 dapat dijelaskan secara rinci pada Table 3.4.

Table 3.4 Penjelasan Rancangan Analisis

No	<i>Input</i>	<i>Analisis</i>	<i>Output</i>
1.	Hasil wawancara dengan stakeholder	Menganalisis perangkat lunak yang akan dibangun	Lingkup dari perangkat lunak yang akan dibangun Lingkup dari perangkat lunak yang akan dibangun bertujuan mendefinisikan lingkup pembangunan perangkat lunak yang akan dibangun terkait hasil wawancara yang dilakukan.
2.	Lingkup dari perangkat lunak yang akan dibangun	Mendefinisikan kebutuhan perangkat lunak	Kebutuhan perangkat lunak. Kebutuhan perangkat lunak tersebut mendeskripsikan fitur-fitur yang akan tersedia pada perangkat lunak demi menunjang penyelesaian masalah yang terjadi.
3.	Kebutuhan perangkat lunak	Membuat rancangan dari perangkat lunak yang akan dibangun	Rancangan antarmuka dan Pemodelan aplikasi yang akan dibangun. Pemodelan aplikasi bertujuan untuk merepresentasikan hasil analisis yang dilakukan berdasarkan kebutuhan perangkat lunak yang sudah ditentukan. Sedangkan untuk rancangan antarmuka bertujuan untuk membuat interaksi antara user dan aplikasi ketika digunakan.

### 3.3.2 Kerangka Pemikiran Teoritis

Kerangka pemikiran konsep digunakan untuk melihat keterkaitan dan dukungan literatur terhadap produk tugas akhir yang dihasilkan. Konsep *Rational Unified Process* (RUP) akan digunakan dalam pembangunan aplikasi. Aplikasi pemesanan tiket di kolam renang Failda digunakan untuk memudahkan pelanggan dalam memesan tiket kolam renang dan untuk memudahkan pegawai dalam mengelola data tiket yang terjual dan uang yang masuk setiap harinya. Berdasarkan kerangka pemikiran konsep, maka aplikasi pemesanan tiket di kolam renang Failda dapat digunakan untuk mengatasi permasalahan yang telah dijelaskan pada Gambar 3.1 Kerangka Permasalahan. Kerangka pemikiran konsep yang digunakan pada penelitian tugas akhir ini dapat dilihat pada Gambar 3.2 sebagai berikut.



Gambar 3.2 Kerangka Pemikiran Teoritis

### 3.4 Analisis Teknologi

Analisis teknologi bertujuan menentukan teknologi-teknologi yang sesuai untuk menyelesaikan permasalahan yang sesuai dengan topik tugas akhir. Teknologi-teknologi yang akan digunakan akan dipaparkan pada poin-poin berikut ini:

#### 3.4.1 Asynchronous Javascript and XML (AJAX)

AJAX adalah kependekan dari Asynchronous javascript and XML, dalam bahasa Indonesiaya asinkron antara javascript dan XML, pengertian mudahnya menggabungkan antara javascript dan xml untuk mengakses sumber data di server. Jadi server tidak diakses secara langsung, biarkan mesin AJAX yang mengaksesnya. Javascript sebagai pemrograman di sisi client (artinya program yang dibuat dengan javascript, bisa dijalankan tanpa menggunakan server) sekarang ini bisa digunakan untuk mengakses server secara asinkron (di belakang layar, artinya proses akses tidak terlihat oleh user) [SAR12].

AJAX adalah aplikasi cross-platform dan cross-browser. Melalui AJAX, JavaScript dapat dikomunikasikan secara langsung dengan server menggunakan obyek JavaScript XMLHttpRequest. Objek JavaScript ini dapat men-trade data sebuah web server harus me-reload (refresh) halaman web. AJAX menggunakan asynchronous data transfer (pada HTTP request) antara browser dan web

server, yang memperbolehkan halaman web me-request bit yang kecil atau seluruh informasi dari server. Teknik AJAX membuat aplikasi internet menjadi kecil, cepat dan lebih user-friendly [ASU10].

AJAX (*Asynchronous Javascript and XML*) digunakan untuk melakukan request pada aplikasi web yang memiliki kemampuan pengambilan data secara *asynchronous* dapat dilakukan dengan menggunakan XMLHttpRequest yang merupakan bagian paling penting dari teknologi Ajax. Oleh karena itu, permintaan ditangani secara efektif dan pemuatannya konten konten dinamis ditampilkan dengan peningkatan kinerja yang jauh lebih tinggi dan melakukan perubahan pada konten tanpa harus me *reload* halaman.

### 3.4.2 Laravel

Laravel merupakan *framework* PHP yang dirilis dibawah lisensi MIT dengan kode sumber yang sudah disediakan oleh Github, Laravel dibangun dengan konsep MVC (*Model View Controller*), kemudian Laravel dilengkapi juga *command line tool* yang bernama “Artisan” yang bisa digunakan untuk *packaging bundle* dan *instalasi bundle* melalui *command prompt* [AMI15].

#### 3.4.2.1 Fitur-Fitur Laravel

Laravel mempunyai fitur-fitur yang bisa digunakan diantaranya sebagai berikut :

1. *Bundles*, yaitu sebuah fitur dengan sistem pengemasan modular dan berbagai bundle telah tersedia untuk digunakan dalam aplikasi.
2. *Eloquent ORM*, merupakan penerapan PHP lanjutan dari ola “active record” menyediakan metode internal untuk mengatasi kendala hubungan antara objek *database*. Pembangun query Laravel Fluent didukung Eloquent.
3. *Application Logic*, merupakan bagian dari aplikasi yang dikembangkan baik menggunakan Controllers maupun sebagai bagian dari deklarasi Route.
4. *Reverse Routing*, mendefinisikan hubungan antara link dan Route, sehingga jika suatu saat ada perubahan pada route secara otomatis akan tersambung dengan link yang relevan.
5. *Restful Controllers*, memberikan sebuah pilihan untuk memisahkan logika melayani HTTP GET dan permintaan POST
6. *Class Auto Loading*, menyediakan otomatis loading untuk *class-class* PHP, tanpa membutuhkan pemeriksaan manual terhadap jalur masuknya
7. *View Composers* adalah kode unit *logical* yang dapat dijalankan ketika sebuah *View* di *load*.
8. *Migrations*, menyediakan versi sistem control untuk skema database, sehingga memungkin untuk menghubungkan perubahan adalah basis kode aplikasi dan keperluan yang dibutuhkan dalam merubah tata letak database, mempermudah dalam penempatan dan memperbarui aplikasi.

#### 3.4.2.2 Kelebihan Laravel

Berikut ini adalah kelebihan-kelebihan yang dimiliki oleh Laravel [AMI15]:

1. Expressif

Ketika melihat suatu sintaks Laravel, seorang programmer diharapkan akan langsung tahu kegunaan dai sintaks tersebut meskipun belum pernah mempelajarnya apalagi menggunakannya.

2. Simple

Sintaks yang digunakan di laravel sangatlah simple bisa memangkas beberapa baris ketika digunakan di *framework* lain. Contohnya dalam penggunaan Eloquent ORM.

3. Accessible

Maksud dari *accessible* disini adalah laravel membuat dokumentasi yang selengkap mungkin. Coe developernya dari Laravel sendiri berkomitmen untuk selalu menyertakan dokumentasi yang lengkap setiap kali rilis versi terbarunya.

Laravel digunakan penulis untuk membangun aplikasi web pengelolan tiket dan membuat *web service*. Alasan menggunakan Laravel adalah karena laravel framework PHP yang *powerfull* saat ini, banyak kelebihan yang dimiliki oleh Laravel salah satunya yaitu Artisan, dimana untuk membuat *controller*, *model*, *request*, *middleware* dan lain-lain dilakukan dengan mengetikan perintah di *command prompt*. Ini membuat pekerjaan menjadi lebih cepat. Selain itu, didukung juga dengan dokumentasi yang lengkap serta komunitas-komunitas yang ada sehingga memudahkan dalam mengembangkan aplikasi.

### **3.4.3 Android**

Android adalah system operasi yang dimodifikasi untuk perangkat bergerak (*mobile device*) yang terdiri dari system operasi, *middleware*, dan aplikasi-aplikasi utama. Awalnya, Android dikembangkan Android inc. Perusahaan ini kemudian dibeli Google pada tahun 2005. System operasi Android kemudian diluncurkan bersamaan dengan dibentuknya organisasi *Open Handset Alliance* tahun 2007. Selain Google, beberapa nama-nama besar juga ikut serta dalam *Open Handset Alliance*, antara lain Motorola, Samsung, LG, Sony Ericsson, T-Mobile, Vodafone, Toshiba, dan Intel [ZPJ16]. Android ini digunakan untuk membuat aplikasi berbasis mobile untuk pemesanan yang dilakukan *customer* dan untuk melakukan scan QR Code untuk keperluan verifikasi data pemesanan.

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, middleware, dan aplikasi [NSA11].

Alasan menggunakan android untuk membuat aplikasi mobile adalah karena android merupakan sistem operasi yang *open source*. Kode dengan sumber terbuka atau *open source* dan terdapat lisensi perizinan pada Android memungkinkan perangkat lunak untuk memodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat atau pengembang aplikasi.

#### **3.4.3.1 Android SDK**

Android SDK mencakup perangkat tools pengembangan yang komprehensif. Android SDK terdiri dari debugger, libraries, handset emulator, dokumentasi, contoh kode program dan tutorial. Saat

ini Android sudah mendukung arsitektur x86 pada Linux (distribusi Linux apapun untuk desktop modern), Mac OS X 10.4.8 atau lebih, Windows XP atau Vista. Persyaratan mencakup JDK, Apache Ant dan Python 2.2 atau lebih. IDE yang didukung secara resmi adalah Eclipse 3.2 atau lebih dengan menggunakan plugin Android Development Tools (ADT), dengan ini pengembang dapat menggunakan IDE untuk mengedit dokumen Java dan XML serta menggunakan peralatan command line untuk menciptakan, membangun, melakukan debug aplikasi Android dan pengendalian perangkat Android (misalnya reboot, menginstal paket perangkat lunak) [ASU16].

### 3.4.3.2 Arsitektur Aplikasi Berbasis Android

Google mengibaratkan Android sebagai sebuah tumpukan *software*. Setiap lapisan dari tumpukan ini menghimpun beberapa program yang mendukung fungsi-fungsi spesifik dari sistem operasi. Secara garis besar arsitektur Android dapat dijelaskan dan digambarkan sebagai berikut [ZPJ16] :

#### 1. *Applications* dan *Widgets*

*Applications* dan *Widgets* ini adalah layer dimana kita berhubungan dengan aplikasi saja, dimana biasanya kita download aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut.

#### 2. *Application Framework*

*Application Framework* adalah layer dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di system operasi Android, karena pada layer inilah aplikasi dapat dirancang dan dibuat. Komponen-komponennya yaitu *Views*, *Content Provider*, *Resource Manager*, *Notification Manager*, *Activity Manager*.

#### 3. *Libraries*

*Libraries* merupakan layer dimana fitur-fitur Android untuk menjalankan aplikasinya. Layer ini meliputi berbagai *library* C/C++ inti seperti *Libc* dan *SSL*, serta :

- *Libraries* media untuk pemutaran media audio dan video.
- *Libraries* untuk manajemen tampilan.
- *Libraries\_Graphic* mencakup SGL dan OpenGL untuk grafis 2D dan 3D.
- *Libraries SWLite* untuk dukungan *database*.
- *Libraries SSL* dan *Webkit* terintegrasi dengan *web browser* dan *security*
- *Libraries liveWebcore* mencakup modrn *web browser* dengan engine *embedded web view*.
- *Libraries* 3D yang mencakup implementasi OpenGL S 1.0 APIs.

#### 4. *Android Run Time*

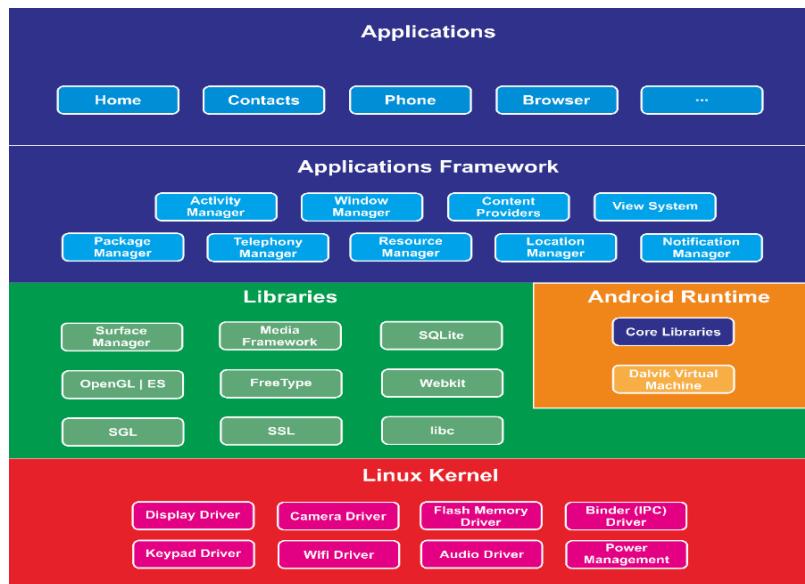
Layer yang membuat aplikasi Android dapat dijalankan di mana dalam prosesnya menggunakan implementasi linux. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android. Di dalam *run time* terdiri dari:

- *Core Libraries* : Aplikasi Android dibangun dalam bahasa *java*, sementara Dalvik sebagai virtual mesinnya bukan *Virtual Machine Java*, sehingga diperlukan sebuah *Libraries* yang berfungsi untuk menterjemahkan bahasa *java/C* yang ditangani oleh *Core Libraries*.

- *Dalvik Virtual Machine* : Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, di mana merupakan pengembangan yang mampu membuat linux kernel untuk melakukan *threading* dan manajemen tingkat rendah.

## 5. Linux Kernel

*Linux kernel* merupakan layer dimana inti dari *operating system* dari Android yang berisi *file file* sistem untuk mengatur sistem *processing, memory, resource, drivers*, dan sistem operasi Android lainnya. *Linux kernel* yang digunakan Android adalah *linux kernel release 2.6*. Arsitektur Android dapat dilihat pada Gambar 3.3 :



Gambar 3.3 Arsitektur Android [ZPJ16]

### 3.4.4 Retrofit

Retrofit merupakan REST client library yang aman untuk android dan java. Dibuat oleh Square. Retrofit menyediakan cara yang aman untuk autentikasi dan interaksi dengan berbagai API lainnya. Sehingga memungkinkan pengiriman permintaan jaringan dengan OkHttp. Retrofit mengambil data JSON atau XML dari web API dan saat data diterima akan langsung diubah ke Plain Old Java Object (POJO). Sehingga harus ditentukan setiap class yang akan dipakai saat response diterima [MTA18].

Alasan menggunakan retrofit adalah karena kemudahan dalam interaksi dengan API dan fleksibilitasnya yang memudahkan untuk membuat custom service sesuai yang kita inginkan.

### 3.4.5 Guzzle

Guzzle adalah *library* PHP untuk melakukan HTTP *request* dengan sangat mudah dan dapat diintegrasikan dengan berbagai macam *web service*. Guzzle memiliki kelebihan untuk melakukan HTTP *request* secara asinkron ataupun sinkron. Selain itu dia juga mendukung standar PSR-7 dalam penulisan *library*-nya [MAR17].

Alasan menggunakan Guzzle yaitu sudah disupport oleh laravel yang memudahkan untuk menghubungkan ke web service. Selain itu, Guzzle juga digunakan untuk mengambil data saldo rekening yang telah terdaftar di perusahaan pihak ketiga yaitu Moota. Sehingga memudahkan mengelola keuangan dari pembayaran tiket yang menggunakan aplikasi *mobile*.

### 3.4.6 JSON

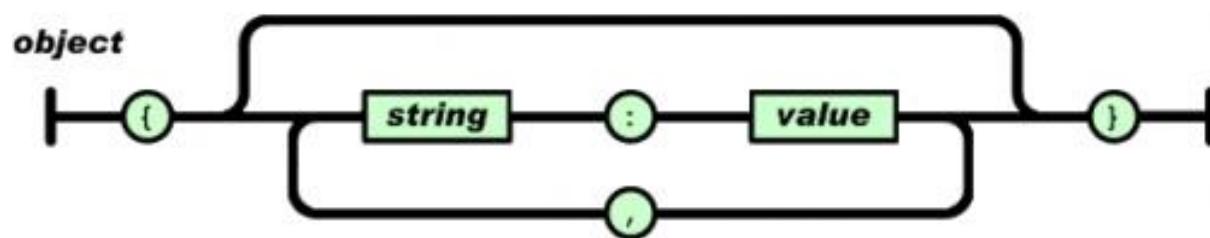
JSON adalah JSON (JavaScript Object Notation) adalah format pertukaran data (lightweight data interchangeformat), mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 –Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C ermasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data [JSO00].

JSON terbuat dari dua struktur:

- Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
- Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

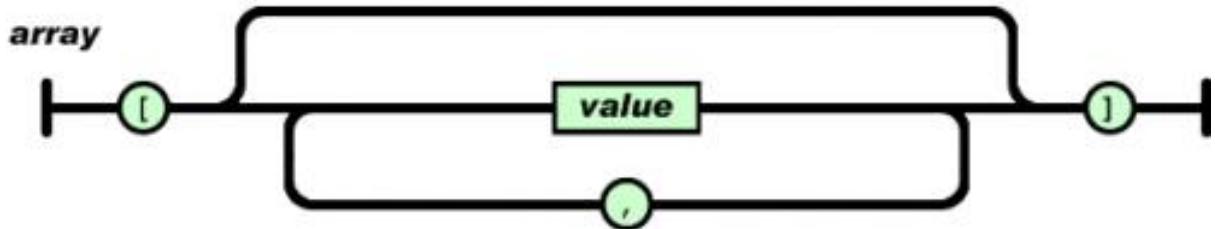
Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman modern mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini JSON menggunakan bentuk sebagai berikut [JSO00]:

**Objek** adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).



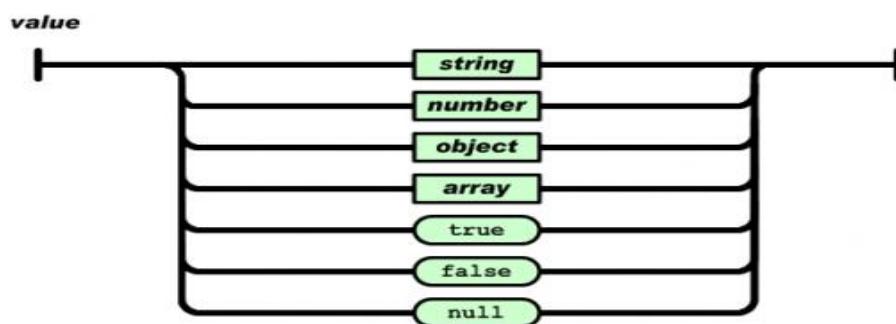
Gambar 3.4 Bentuk Object Pada JSON [JSO00]

**Array** adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [ (kurung kotak buka) dan diakhiri dengan ] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma).



Gambar 3.5 Bentuk Array Pada JSON [JSON00]

**Nilai (value)** dapat berupa sebuah **string** dalam tanda kutip ganda, atau **angka**, atau **true** atau **false** atau **null**, atau sebuah **objek** atau sebuah **larik**. Struktur-struktur tersebut dapat disusun bertingkat.



Gambar 3.6 Bentuk Value Pada JSON [JSO00]

Alasan utama menggunakan JSON (*Javascript Object Notation*) adalah karena JSON format pertukaran datanya yang ringan dan juga banyak didukung oleh bahasa pemrograman. Selain itu, *Restfull* output keluarannya adalah JSON. Maka dari itu, penulis menggunakan JSON sebagai format pertukaran datanya.

### 3.4.7 QR Code

QR Code merupakan teknik yang mengubah data tertulis menjadi kode-kode 2-dimensi yang tercetak kedalam suatu media yang lebih ringkas. QR Code adalah barcode 2-dimensi yang diperkenalkan pertama kali oleh perusahaan Jepang Denso-Wave pada tahun 1994. Barcode ini pertama kali digunakan untuk pendaftaran inventaris produksi suku cadang kendaraan dan sekarang sudah digunakan dalam berbagai bidang. QR adalah singkatan dari Quick Response karena ditujukan untuk diterjemahkan isinya dengan cepat. QR Code merupakan pengembangan dari barcode satu dimensi, QR Code salah satu tipe dari barcode yang dapat dibaca menggunakan kamera handphone [JRO08].

QR Code adalah suatu jenis image dua dimensi yang menampilkan data berupa teks. Dengan QR Code pengguna dapat mengakses suatu web lebih cepat daripada harus melakukan secara manual [IBG16].



Gambar 3.7 QR Code [IBG16]

Alasan menggunakan QR Code adalah karena QR Code dapat menyimpan dan membaca banyak jenis data seperti angka, huruf, symbol, kanji, dan lain-lain. Selain itu, QR Code menampung data secara horizontal dan vertikal, sehingga ukuran dari tampilan gambar kode QR bisa hanya sepersepuluh dari ukuran sebuah kode batang.

### 3.5 Analisis Penggunaan Konsep

Analisis penggunaan konsep bertujuan menentukan konsep teknologi yang sesuai untuk menyelesaikan permasalahan pada topik tugas akhir. Berikut ini konsep teknologi yang akan digunakan akan dipaparkan pada poin-poin di bawah ini:

#### 3.5.1 Web Service

Analisis konsep *web service* bertujuan agar penulis dapat memahami secara teoritis konsep untuk membangun sebuah *web service*. *Web service* merupakan sebuah perangkat lunak yang digunakan sebagai jembatan sehingga memungkinkan berbagai sistem dapat saling berkomunikasi tanpa terpengaruh dengan perbedaan *platform*.

Pada aplikasi pemesanan tiket di kolam renang Failda terdapat dua *platform* yang akan menggunakan *web service* untuk melakukan pertukaran data dengan database yang berada di sistem yaitu aplikasi *Web* dan *Android*. Dengan adanya *web service* kedua platform tersebut hanya perlu memanggil fungsi-fungsi yang sudah disediakan oleh *web service*.

#### 3.5.2 Aplikasi Web

Aplikasi web adalah program apapun yang diakses melalui jaringan HTTP. Aplikasi berbasis web berjalan pada *web browser*. Bagaimanapun, aplikasi berbasis web juga dapat dianggap *client-based*, di mana sebagian program diunduh ke perangkat *desktop*, tetapi diproses melalui internet pada *server* eksternal [TPD18].

Aplikasi web sebenarnya merujuk pada aplikasi apapun yang berkomunikasi dengan penggunanya melalui HTTP. Hal ini termasuk aplikasi ringan, seperti gim berbasis *flash*, kalkulator

*online*, kalender, dan sebagainya sebagaimana aplikasi intensif lainnya, seperti aplikasi pemroses kata berbasis web dan *spreadsheet* [TPD18].

### 3.5.3 Aplikasi Mobile

*Mobile Application* adalah jenis perangkat lunak aplikasi yang dirancang untuk dijalankan di perangkat seluler, seperti komputer smartphone atau tablet. *Mobile application* sering berfungsi untuk menyediakan layanan serupa bagi pengguna yang diakses di PC [TIN18].

## 3.6 Analisis Aplikasi Serupa

Analisis aplikasi serupa bertujuan untuk melihat proses bisnis dan teknologi-teknologi yang digunakan dalam aplikasi tersebut. Sebagai acuan dalam pembangunan aplikasi yang akan dibuat pada tugas akhir ini. Berikut ini adalah beberapa contoh aplikasi yang penulis pilih sebagai acuan untuk aplikasi yang akan dibangun.

### 1. Aplikasi Tiket.com

Tiket.com adalah salah satu perusahaan OTA (Online Travel Agent) terbesar di Indonesia yang menyediakan dan menyelenggarakan sistem dan fasilitas pemesanan dan pembelian tiket secara online. Produk yang ditawarkan lebih bervariasi, tidak hanya untuk memenuhi kebutuhan perjalanan tetapi juga waktu luang Anda. Tiket.com memungkinkan untuk mencari informasi produk yang diinginkan, melakukan pemesanan, pembelian, sekaligus melakukan pembayaran secara online dan aman melalui berbagai sistem dan fasilitas pembayaran yang telah kami sediakan.

Berikut ini adalah cara untuk memesan tiket melalui tiket.com :

- a. Cari penerbangan yang diinginkan
- b. Pilih penerbangan
- c. Isi data penumpang
- d. Lakukan pembayaran
- e. Setelah pembayaran berhasil, maka akan menerima e-ticket yang dikirim melalui email.

### 2. Aplikasi Digiticket

Digiticket adalah aplikasi berbasis *mobile* yang dibangun untuk memesan tiket wahana rekreasi secara online. Digiticket ini menyediakan pemesanan tiket untuk wahana-wahana yang ada di Indonesia contohnya The Lodge Maribaya, Rinjani Waterpark, Kraton Waterpark, Singaporeland Waterpark, dan lain-lain. Berikut ini adalah cara untuk memesan tiket melalui tiket.com :

1. Pilih tempat wisata, buka aplikasi Digiticket dan pilih tempat wisata yang ingin dikunjungi.
2. Pesan tiket, pesan tiket dengan memilih tanggal kunjungan anda dan masukan jumlah tiket yang akan dipesan.
3. Konfirmasi tiket, konfirmasi tiket yang anda pesan cek apakah sudah sesuai dengan tanggal kunjungan yang diinginkan.
4. Masukan kupon, pilih kode promo yang aktif untuk mendapatkan potongan harga tiket.

5. Pilih metode pembayaran, sementara ini terdapat 3 metode pembayaran yang tersedia yaitu via Mandiri, Permata, dan Transfer via ATM Bersama.
  6. Transfer kode booking, setelah memilih metode pembayaran anda memiliki waktu 3 jam untuk mentransfer tiket yang telah dipesan sesuai dengan kode pembayaran pada aplikasi.
  7. E-ticket diterbitkan, setelah pembayaran sukses diterima, e-tiket akan muncul pada aplikasi Anda dalam maksimal 15 menit.
3. Aplikasi TIX ID
- TIX ID adalah aplikasi layanan hiburan terdepan di Indonesia yang memberikan pengalaman baru dalam pembelian tiket film dan hiburan lainnya. Dengan TIX ID, pengguna dapat mengetahui infomasi tentang film terkini serta melakukan pemesanan tiket bioskop dengan mudah, cepat, dan aman.
1. Pilih judul film, lokasi bioskop, jenis studio, tanggal dan jam tayang film
  2. Pilih kursi pada studio yang telah dipilih
  3. Periksa kembali detail pesanan sebelum melakukan pembayaran
  4. Lakukan pembayaran dari tiket yang dibeli
  5. Setelah pembayaran berhasil, maka akan menerima notifikasi status pembelian tiket
  6. Kode pesanan dan rincian lainnya akan diberitahukan melalui aplikasi dan SMS

Dari ketiga aplikasi serupa diatas, dapat ditarik kesimpulan bahwa dalam pemesanan tiket itu yang harus diperhatikan adalah jumlah tiket yang akan dipesan, kemudian tanggal kunjungan, serta pembayaran. Untuk pembayaran bisa dilakukan secara manual dengan mengirim bukti pembayaran atau secara otomatis di verifikasi oleh sistem apabila aplikasi sudah bekerja sama dengan pihak bank. Dan untuk tahap terakhir yaitu di dalam aplikasi harus ada notifikasi pemberitahuan bahwa pembayaran sudah diverifikasi. Notifikasi bisa melalui email atau notifikasi langsung dari aplikasi yang digunakan.

### **3.7 Profile Penelitian**

Berikut ini adalah profil objek penelitian dan profil tempat penelitian yang penulis lakukan, diantaranya sebagai berikut :

#### **3.7.1 Profil Objek Penelitian**

Di kolam renang Failda rata-rata pengunjung datang setiap harinya mencapai 300 orang. Terkecuali pada saat musim libur sekolah pengunjung bisa datang lebih banyak lagi kira-kira mencapai 1000 - 2000 orang setiap harinya. Selanjutnya proses pemesanan tiket saat ini pengunjung datang langsung ke kolam renang. Ketika pengunjung datang, ada petugas yang bertugas untuk menghitung jumlah orang yang akan berenang. Selanjutnya pengunjung membayar uang masuk dan setelah itu mendapatkan tiket masuk ke kolam renang. Kemudian tiket akan ditarik kembali oleh petugas sebelum pengunjung memasuki kolam renang. Hal tersebut dilakukan untuk keperluan rekapitulasi penyesuaian data tiket yang dibeli pengunjung dan pemasukan uang setiap harinya.

### 3.7.2 Profil Tempat Penelitian

Profil tempat penelitian tugas akhir ini dilakukan di kolam renang Failda pada bagian kasir. Adapun profil dari kolam renang Failda adalah sebagai berikut :

Tempat Penelitian	: Bagian kasir kolam renang Failda
Alamat	: Kp. Rawasari RT/RW 02/03 Kecamatan Sukanagara Desa Sukanagara Kabupaten Cianjur, Jawa Barat, 42364.



Gambar 3.8 Logo Perusahaan

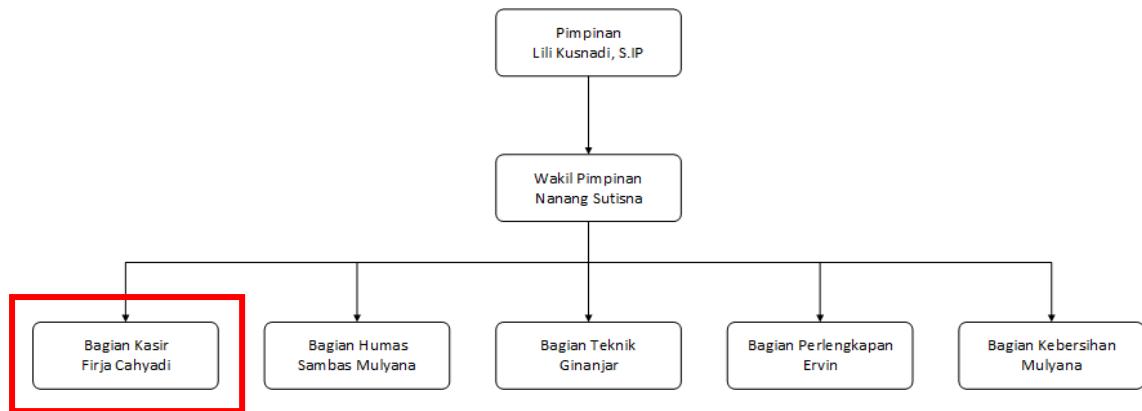
#### 3.7.2.1 Sejarah Singkat

Kolam renang Failda merupakan salah satu kolam renang yang berada di Sukanagara Cianjur Selatan dan didirikan pada tahun 2006. Didirikannya kolam renang ini di latar belakangi kebutuhan akan sarana olahraga yang memang belum tersedia, mengingat menjadi skala prioritas karena keberadaannya berlokasi di luar kota dalam hal ini berada di wilayah kecamatan. Agar tidak ketinggalan anak-anak yang mempunyai potensi dengan wilayah yang berada di wilayah kota tercetus ide untuk membangun fasilitas kolam renang dengan nama Failda.

Sebelum dibangun sarana kolam renang, aslinya lokasi tersebut digunakan untuk kolam pemancingan yang biasa digunakan masyarakat sekitar. Karena sesuai dengan kebutuhan tersebut, maka berkordinasi dengan pemerintah setempat dalam hal ini MUSPIKA kecamatan Sukanagara kab. Cianjur untuk membangun fasilitas kolam renang dengan tujuan membantu masyarakat terutama para pelajar yang mempunya skil di bidang berenang.

#### 3.7.2.2 Struktur Organisasi

Berikut ini adalah gambar dari struktur organisasi tempat penelitian tugas akhir yaitu kolam renang Failda. Sebagaimana dapat dilihat pada Gambar 3.9.



Gambar 3.9 Struktur Organisasi

Bagian kasir memiliki tugas untuk melakukan pelayanan pada transaksi penjualan tiket di Kolam Renang Failda. Tugas bagian kasir dimulai dari melakukan penjualan tiket kolam renang dan melakukan pembukuan penjualan tiket harian.

## **BAB 4**

### **PERENCANAAN DAN PEMODELAN**

Bab ini menjelaskan proses awal pembangunan perangkat lunak yang terdiri dari analisis dan perancangan. Proses analisis terdiri dari pendefinisian perangkat lunak, pendefinisian kebutuhan yang harus dipenuhi perangkat lunak, dan analisis terhadap kebutuhan dengan menggunakan pemodelan spesifikasi kebutuhan dan pemodelan kebutuhan. Sedangkan proses perancangan perangkat lunak terdiri dari perancangan arsitektural, perancangan antarmuka, dan perancangan data.

#### **4.1 Current System (Sistem saat ini)**

Saat ini, proses pemesanan tiket di kolam renang Failda yaitu pengunjung harus datang langsung ke tempat untuk membeli tiket. Ketika pengunjung sudah berada di tempat kasir, maka ada petugas yang akan menghitung jumlah orang yang akan masuk ke kolam renang. Tiket yang digunakan saat ini yaitu masih menggunakan tiket kertas, dimana tiket tersebut harus ditarik kembali oleh petugas untuk keperluan rekapitulasi data tiket yang terjual dan uang yang masuk setiap harinya. Proses rekapitulasinya sendiri saat ini masih menggunakan catatan di buku kemudian ketika kolam renang sudah tutup pimpinan baru bisa melihat hasil penjualan tiket per harinya.

#### **4.2 Analisis Kebutuhan**

Analisis kebutuhan akan menghasilkan spesifikasi-spesifikasi dari karakteristik-karakteristik operasional yang akan dimiliki oleh perangkat lunak yang akan dikembangkan. Pada tahap ini penulis selaku analis sistem, melakukan interaksi dengan calon pengguna, yaitu dengan pemilik (*owner*) Failda Waterpark dan karyawan yang nantinya akan menggunakan sistem tersebut.

##### **4.2.1 Menentukan Kebutuhan Dasar**

Pada tahap analisis kebutuhan, harus ditentukan kebutuhan aplikasi yang akan dibangun. Hal-hal apa saja yang harus ada dan bisa dilakukan oleh aplikasi. Berikut ini adalah kebutuhan fungsional dan kebutuhan non fungsional, diantaranya sebagai berikut :

###### **4.2.1.1 Analisis Stakeholder**

Berikut ini adalah *stakeholder-stakeholder* atau orang yang berkepentingan yang terlibat dalam pengelolaan tiket dan pemesanan tiket di kolam renang Failda. Dapat dilihat pada Table 4.1 diantaranya sebagai berikut :

Table 4.1 Daftar Stakeholder

No.	Stakeholder	Deskripsi
1.	Pengembang aplikasi	Orang yang bertugas untuk membuat aplikasi pengelolaan tiket dan pemesanan tiket.
2.	Pimpinan	Orang yang memiliki perusahaan Failda yang memantau atau memonitor jumlah tiket yang terjual dan pemasukan uang yang masuk.

3.	Kasir	Orang yang bertugas untuk melayani customer yang akan masuk ke kolam renang yang memesan tiket datang langsung ke tempat.
4.	Petugas penarik tiket	Orang yang bertugas menarik tiket sebelum pengunjung masuk ke kolam renang.

#### 4.2.1.2 Analisis Fungsional

Berikut ini adalah analisis fungsional yang dibutuhkan oleh aplikasi yang akan dibangun, dapat dilihat pada Table 4.2 diantaranya sebagai berikut :

Table 4.2 Analisis Fungsional

No.	Kode	Kebutuhan Fungsional	Deskripsi
1.	FR-01	Perangkat lunak dapat menginput pemesanan tiket	Perangkat lunak dapat menginput pemesanan tiket yang dilakukan oleh customer baik secara langsung ataupun online melalui aplikasi mobile
2.	FR-02	Perangkat lunak dapat mengupdate pemesanan dari customer	Perangkat lunak bisa mengupdate pemesanan tiket yang dilakukan oleh customer baik secara langsung.
3.	FR-03	Perangkat lunak dapat mencetak bukti pembayaran pemesanan tiket dari customer	Perangkat lunak dapat mencetak bukti pembayaran pemesanan tiket dari customer. Bukti pembayaran ini akan dijadikan sebagai tiket masuk yang nantinya qr code dalam bukti pembayaran tersebut akan dipindai. Requirement ini khusus bagi customer yang memesan langsung dan petugas menginput data pemesanan langsung dari web.
4.	FR-04	Perangkat lunak dapat mengelola data konfirmasi pembayaran dari customer	Perangkat lunak dapat mengelola data konfirmasi pemesanan dari customer yang melakukan pemesanan tiket melalui aplikasi mobile. Customer harus mengupload bukti pembayarannya untuk dikonfirmasi.
5.	FR-05	Perangkat lunak dapat menampilkan notifikasi pembayaran masuk secara realtime	Perangkat lunak akan menampilkan notifikasi masuk dari pembayaran tiket yang dilakukan oleh customer secara online melalui mobile yang akan tampil di web.
6.	FR-06	Perangkat lunak dapat memverifikasi pemesanan dari customer	Perangkat lunak dapat memverifikasi pemesanan dari customer. Verifikasi yang dilakukan adalah dengan memindai qr code dari pemesanan tiket yang dilakukan oleh customer.
7.	FR-07	Perangkat lunak dapat menginput form konfirmasi pembayaran	Perangkat lunak harus bisa menginput form konfirmasi pembayaran dan mengupload bukti pembayaran setelah melakukan pemesanan. Khusus untuk pemesanan melalui aplikasi mobile.
8.	FR-08	Perangkat lunak dapat mengelola akun karyawan	Perangkat lunak harus bisa mengelola akun karyawan untuk menggunakan aplikasi web.
9.	FR-09	Perangkat lunak dapat menginput akun customer	Perangkat lunak harus bisa menginput akun customer ketika melakukan registrasi di aplikasi mobile.
10.	FR-10	Perangkat lunak dapat mengelola content berita	Perangkat lunak harus bisa mengelola content berita untuk di aplikasi mobile.
11.	FR-11	Perangkat lunak dapat menampilkan data rekap penjualan tiket	Perangkat lunak harus bisa menampilkan data rekap penjualan tiket yang ditampilkan di menu dashboard

#### 4.2.2 Analisis Pengguna

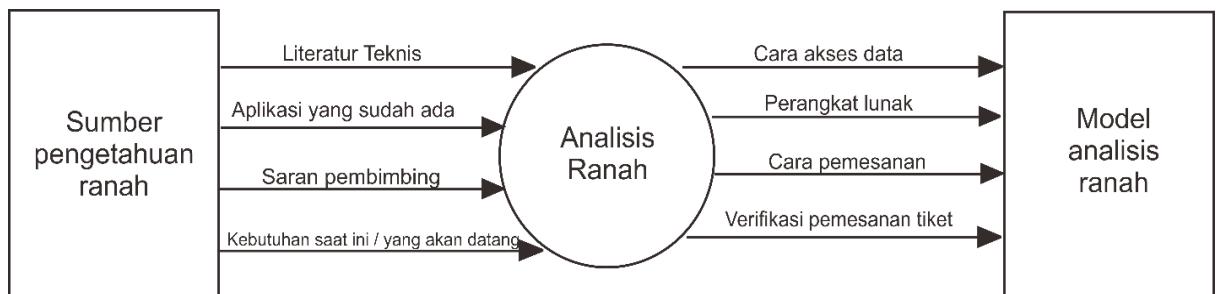
Analisis pengguna merupakan analisis yang digunakan untuk menentukan siapa saja user yang akan menggunakan aplikasi yang akan dibangun, baik dari aplikasi berbasis web ataupun aplikasi berbasis *mobile*. Dapat dilihat pada Table 4.3 diantaranya sebagai berikut :

Table 4.3 Daftar Pengguna

No.	User	Deskripsi
1.	Kasir	Orang yang bertugas untuk melayani customer yang akan masuk ke kolam renang yang memesan tiket datang secara langsung ke kolam renang dan melayani customer yang melakukan pemesanan tiket secara online lewat mobile.
2.	Petugas penarik tiket	Orang yang bertugas memindai QR Code dari tiket yang dibawa oleh customer, baik yang secara langsung atau online melalui aplikasi mobile.
3.	Admin	Orang yang memiliki hak akses penuh terhadap menu yang ada di aplikasi dan bertugas mengelola aplikasi.
4.	Customer	Orang yang memesan tiket secara online melalui aplikasi mobile.

#### 4.2.3 Analisis Ranah

Analisis ranah adalah proses mengidentifikasi, menganalisis, dan menspesifikasikan kebutuhan-kebutuhan umum dari suatu ranah aplikasi yang sifatnya spesifik, biasnaya agar suatu saat dapat digunakan pada berbagai proyek lain yang ada pada ranah aplikasi yang sama. Dapat dilihat pada Gambar 4.1.



Gambar 4.1 Analisis Ranah [PRE10]

Dibawah ini adalah deskripsi dari hasil output yang ada pada analisis ranah, diantaranya sebagai berikut :

Table 4.4 Deskripsi Hasil Output Analisis Ranah

No.	Nama Output	Deskripsi
1.	Cara akses data	Hasil dari output cara akses data ini adalah menghasilkan teknologi untuk pertukaran data yaitu menggunakan web service.
2.	Perangkat lunak	Hasil dari output perangkat lunak ini adalah akan dibangunnya dua aplikasi yaitu aplikasi berbasis web untuk pengelolaan data tiket dan aplikasi berbasis mobile untuk pemesanan tiket secara online.
3.	Cara pemesanan	Hasil dari output cara pemesanan adalah akan ada dua cara pemesanan tiket yaitu pemesanan secara langsung yang datang ke tempat dan pemesanan secara online melalui aplikasi mobile.
4.	Verifikasi pemesanan tiket	Hasil dari output verifikasi pemesanan tiket adalah menghasilkan cara verifikasi tiket yang menggunakan teknologi QR Code yang berguna untuk penyesuaian data tiket yang terjual dan uang yang masuk.

#### 4.2.4 Pendekatan Untuk Pemodelan Spesifikasi Kebutuhan

Suatu pandangan dari pemodelan spesifikasi-spesifikasi kebutuhan, yang dinamakan sebagai analisis terstruktur, memperlakukan data dan proses yang melakukan transformasi data tersebut sebagai entitas yang terpisah. Jadi, pemodelan spesifikasi kebutuhan ini untuk memisahkan mana untuk

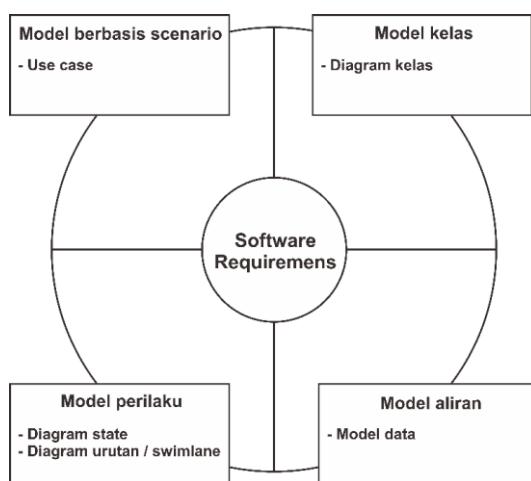
keperluan data dan mana untuk proses. Pada sub bab ini akan dijelaskan mengenai perkakas yang digunakan dan menggambarkan dari spesifikasi kebutuhan yang akan dibuat.

#### 4.2.4.1 UML (*Unified Modelling Process*)

Unified Modelling Language (UML) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. Abstraksi konsep dasar UML terdiri dari structural classification, dynamic behavior, dan model management yang dapat dipahami main concepts sebagai term yang akan muncul pada saat membuat diagram dan view adalah kategori dari diagram tersebut. UML mendefinisikan diagram-diagram sebagai *Use case diagram*, *Class diagram*, *Statechart diagram*, *Activity diagram*, *Sequence diagram*, *Collaboration diagram*, *Component diagram*, dan *Deployment diagram*. Sebagian dari diagram-diagram itulah yang akan digunakan untuk pemodelan sistem atau perangkat lunak yang akan dibuat.

#### 4.2.4.2 Elemen-Elemen Model Analisis

Dibawah ini adalah elemen-elemen model analisis kebutuhan yang digunakan. Dapat dilihat pada gambar Gambar 4.2 diantaranya sebagai berikut :



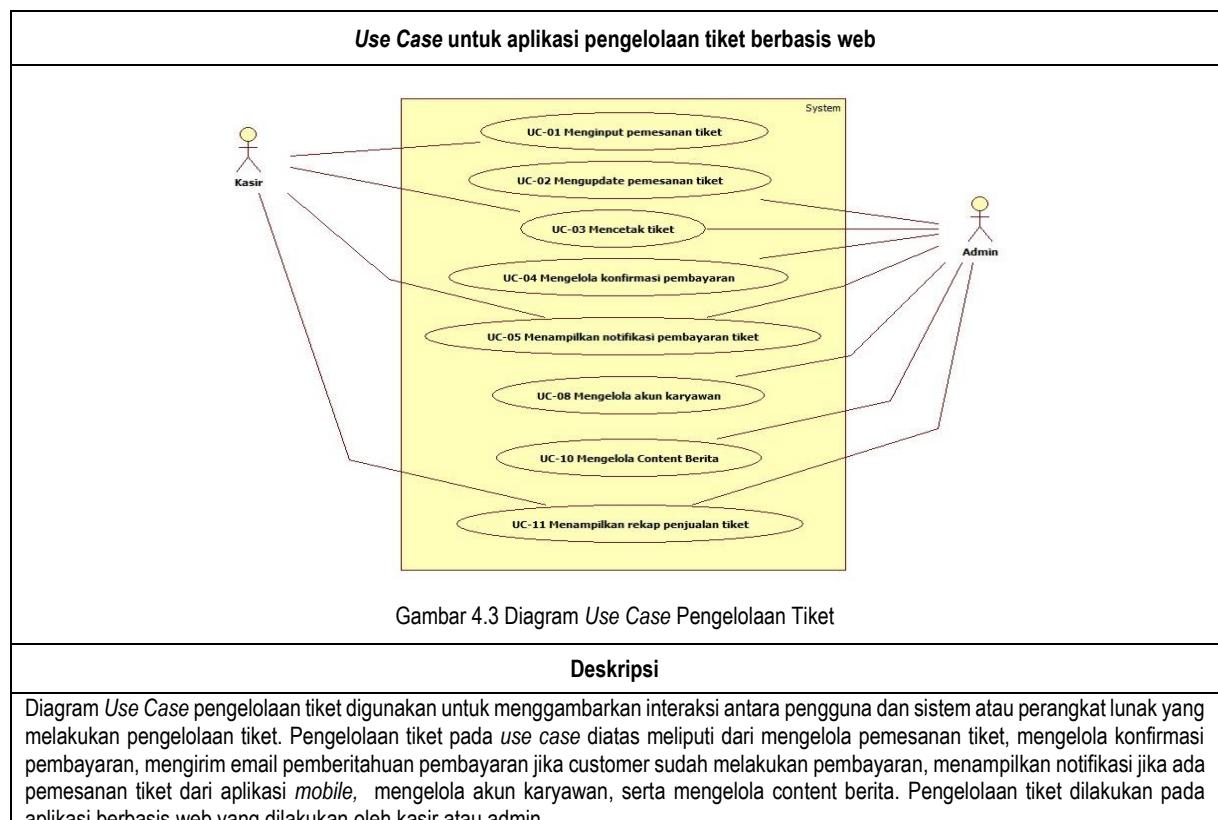
Gambar 4.2 Elemen Model Analisis [PRE10]

Masing-masing unsur model kebutuhan pada Gambar 4.2 memperlihatkan permasalahan dari berbagai sudut pandang yang berbeda. Elemen-elemen berbasis skenario memperlihatkan bagaimana interaksi yang akan terjadi antara pengguna dengan sistem atau perangkat lunak yang akan dikembangkan. Elemen model berbasis kelas memodelkan objek-objek yang akan dimanipulasi oleh sistem, memodelkan operasi-operasi yang akan diterapkan pada objek-objek untuk melakukan manipulasi, memodelkan relasi (beberapa bersifat hierarkis) yang terjadi diantara objek satu dengan yang lainnya. Elemen-elemen perilaku memperlihatkan bagaimana *event-event* eksternal melakukan perubahan pada keadaan (*state*) sistem atau kelas-kelas yang ada didalamnya. Terakhir, elemen-elemen berorientasi aliran memperlihatkan sistem atau perangkat lunak yang bertindak sebagai pelaku transformasi informasi, memperlihatkan bagaimana objek-objek data ditransformasikan saat megnalir melintasi berbagai fungsi yang dimiliki sistem.

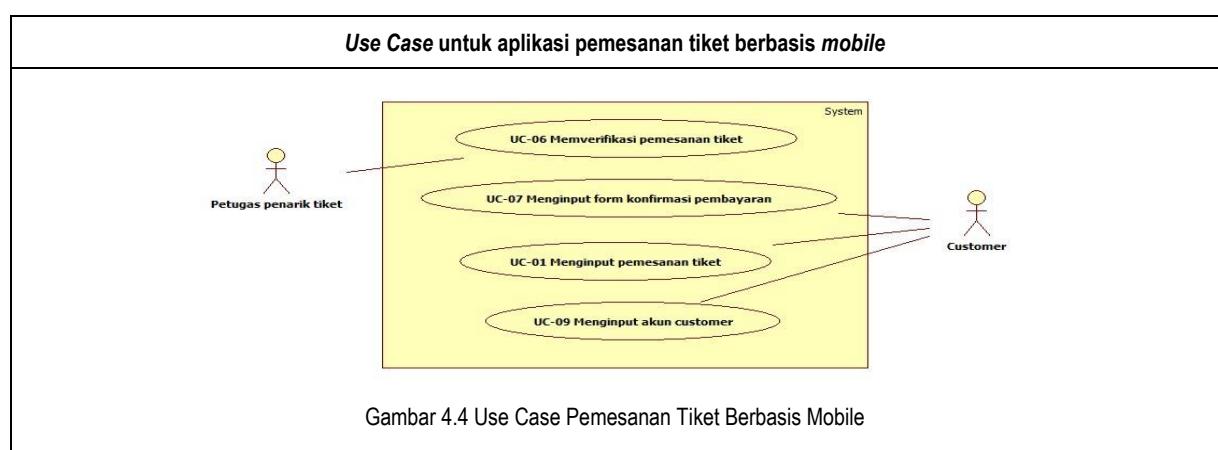
#### 4.2.5 Pemodelan Berbasis Skenario

Pemodelan berbasis skenario memperlihatkan bagaimana interaksi yang akan terjadi antara pengguna dengan sistem atau perangkat lunak yang akan dikembangkan. Pemodelan spesifikasi kebutuhan pengguna menggunakan UML (*Unified Modelling Language*) pada umumnya dimulai dengan pembuatan skenario-skenario dalam bentuk *use case-use case*, dan diagram *swimlane*. Berikut ini adalah *use case* yang dibangun untuk aplikasi pengelolaan tiket berbasis web dan aplikasi pemesanan tiket berbasis *mobile*. Diagram *use case* untuk aplikasi pengelolaan tiket dan pemesanan tiket dapat dilihat pada Gambar 4.3 dan Gambar 4.4.

##### 1. Use Case untuk aplikasi pengelolaan tiket



##### 2. Use Case untuk aplikasi pemesanan tiket berbasis mobile



Deskripsi
Diagram <i>Use Case</i> pemesanan tiket berbasis mobile ini dilakukan oleh customer yang melakukan pemesanan tiket secara online melalui aplikasi <i>mobile</i> . Dan juga pada <i>use case</i> diagaram ini ada <i>use case</i> untuk memverifikasi pemesanan tiket yang dilakukan oleh petugas penarik tiket.

#### 4.2.5.1 Deskripsi *Use Case*

Diagram *Use Case* pada Gambar 4.3 dan Gambar 4.4 dapat dijelaskan lebih detail. Penjelasan *use case* pada Gambar 4.3 dan Gambar 4.4 dapat dilihat pada Table 4.5.

Table 4.5 Deskripsi *Use Case*

No.	Kode	Nama <i>Use Case</i>	Deskripsi
1.	UC-01	Menginput pemesanan tiket	<i>Use case</i> mengelola pemesanan tiket digunakan untuk mengelola pemesanan tiket dari customer baik yang dilakukan secara langsung ataupun online melalui aplikasi <i>mobile</i> .
2.	UC-02	Mengupdate pemesanan tiket	<i>Use case</i> mengupdate pemesanan tiket digunakan oleh admin yang akan jika terjadi kesalahan.
3.	UC-03	Mencetak tiket	<i>Use case</i> mencetak tiket digunakan untuk mencetak tiket yang dipesan oleh customer untuk masuk ke kolam renang yang akan di scan qr code pada tiket tersebut oleh petugas.
4.	UC-04	Menglola konfirmasi pembayaran	<i>Use case</i> mengelola konfirmasi pembayaran digunakan untuk mengelola pembayaran dari customer yang melakukan pemesanan tiket secara online melalui aplikasi <i>mobile</i> .
5.	UC-05	Menampilkan notifikasi pembayaran tiket	<i>Use case</i> menampilkan notifikasi pembayaran tiket digunakan untuk mengecek pembayaran tiket yang dilakukan oleh customer secara online melalui aplikasi <i>mobile</i> . Jika ada maka notifikasi akan muncul berupa angka sebanyak pemesanan tiket yang masuk.
6.	UC-06	Memverifikasi pemesanan tiket	<i>Use case</i> memverifikasi pemesanan tiket digunakan untuk memverifikasi tiket yang dibawa oleh customer. Dan akan di scan qr codenya oleh petugas.
7.	UC-07	Menginput form konfirmasi pembayaran	<i>Use case</i> menginput form konfirmasi pembayaran digunakan oleh customer untuk mengisi form konfirmasi pembayaran dan mengupload bukti pembayaran setelah melakukan pembayaran.
8.	UC-08	Mengelola akun karyawan	<i>Use case</i> mengelola akun karyawan adalah <i>use case</i> yang diperuntukan untuk mengelola akun karyawan yang menggunakan aplikasi/ <i>mobile</i> .
9.	UC-09	Menginput akun customer	<i>Use case</i> menginput akun customer adalah <i>use case</i> yang digunakan untuk menginput akun customer ketiga daftar di aplikasi <i>mobile</i> .
10.	UC-10	Mengelola content berita	<i>Use case</i> mengelola content adalah <i>use case</i> yang digunakan untuk mengelola content berita yang digunakan di aplikasi <i>mobile</i> .
11.	UC-11	Menampilkan rakap penjualan tiket	<i>Use case</i> menampilkan rakap penjualan tiket adalah <i>use case</i> yang digunakan untuk melihat rekapitulasi penjualan tiket setiap harinya.

#### 4.2.5.2 Skenario *Use Case*

Skenario *use case* merupakan gambaran proses berjalannya perangkat lunak berdasarkan *use case* yang telah didefinisikan. Skenario *use case* melibatkan aktor yang berinteraksi dengan *use case* tersebut serta aksi-reaksi yang terjadi antara aktor dengan sistem atau perangkat lunak. Berikut ini

merupakan skenario *use case* aplikasi pengelolaan tiket berbasis *web* dan aplikasi pemesanan tiket berbasis *mobile*.

Table 4.6 Skenario *Use Case* Mengelola Pemesanan Tiket

<b>Identifikasi</b>	
No	UC-01
Nama	Menginput pemesanan tiket
Tujuan	Untuk menginput pemesanan tiket yang dilakukan oleh customer secara langsung dan melalui aplikasi <i>mobile</i> .
Deskripsi	Menginput pemesanan tiket adalah <i>use case</i> yang digunakan untuk menginput pemesanan tiket yang dilakukan oleh customer baik datang langsung ataupun melakukan pemesanan secara online melalui aplikasi <i>mobile</i> .
Tipe	High level
Aktor	Customer, kasir
<b>Skenario Utama</b>	
Kondisi awal	Masuk ke menu pemesanan tiket
<b>Aksi aktor</b>	
1. Aktor masuk ke menu pemesanan tiket	2. Sistem menampilkan menu pemesanan tiket
3. Aktor menginput form pemesanan tiket	4. Sistem memproses aksi yang dilakukan oleh aktor
5. Aktor menerima notifikasi dari sistem	
Kondisi akhir	Aktor mengelola pemesanan tiket

Table 4.7 Skenario *Use Case* Mengupdate Pemesanan Tiket

<b>Identifikasi</b>	
No	UC-02
Nama	Mengupdate pemesanan tiket
Tujuan	Untuk mengupdate pemesanan tiket yang dilakukan oleh customer baik langsung ataupun lewat aplikasi <i>mobile</i> apabila ada kesalahan.
Deskripsi	Mengupdate pemesanan tiket adalah <i>use case</i> yang digunakan untuk mengupdate pemesanan tiket yang dilakukan oleh customer apabila ada kesalahan.
Tipe	High level
Aktor	Admin
<b>Skenario Utama</b>	
Kondisi awal	Masuk ke menu pemesanan tiket
<b>Aksi aktor</b>	
1. Aktor masuk ke menu pemesanan tiket	2. Sistem menampilkan menu pemesanan tiket
3. Aktor mengupdate data pemesanan tiket.	4. Sistem memproses aksi yang dilakukan oleh aktor
5. Aktor menerima notifikasi dari sistem	
Kondisi akhir	Aktor mengelola pemesanan tiket

Table 4.8 Skenario *Use Case* Mencetak Tiket

<b>Identifikasi</b>	
No	UC-03
Nama	Mencetak tiket
Tujuan	Untuk mencetak tiket yang dipesan oleh customer secara langsung

Deskripsi	Mencetak tiket adalah <i>use case</i> yang digunakan untuk mencetak tiket yang dipesan oleh customer secara langsung yang nantinya akan discan qr code yang ada pada tiket.
Tipe	High level
Aktor	Admin
<b>Skenario Utama</b>	
Kondisi awal	Masuk ke menu pemesanan tiket
<b>Aksi aktor</b>	
1. Aktor masuk ke menu pemesanan tiket	2. Sistem menampilkan menu pemesanan tiket
3. Aktor memilih data pada tiket yang akan dicetak	4. Sistem menampilkan halaman cetak
5. Aktor mencetak tiket	
Kondisi akhir	Tiket sudah tercetak

Table 4.9 Skenario *Use Case* Mengelola Konfirmasi Pembayaran

Identifikasi	
No	UC-04
Nama	Mengelola konfirmasi pembayaran
Tujuan	Untuk mengelola pembayaran pemesanan tiket yang dilakukan oleh customer lewat aplikasi <i>mobile</i>
Deskripsi	Mengelola konfirmasi pembayaran adalah <i>use case</i> yang digunakan untuk mengupdate, menampilkan, serta menghapus data konfirmasi pembayaran pemesanan tiket yang dilakukan oleh customer.
Tipe	High level
Aktor	Admin
<b>Skenario Utama</b>	
Kondisi awal	Masuk ke menu konfirmasi pembayaran
<b>Aksi aktor</b>	
1. Aktor masuk ke menu konfirmasi pembayaran	2. Sistem menampilkan menu konfirmasi pembayaran
3. Aktor mengelola konfirmasi pemesanan yang dapat mengupdate, menghapus data pembayaran customer.	4. Sistem memproses aksi yang dilakukan oleh aktor
5. Aktor menerima notifikasi dari sistem	
Kondisi akhir	Aktor mengelola konfirmasi pembayaran

Table 4.10 Skenario Use Case Menampilkan Notifikasi Pemesanan Tiket

Identifikasi	
No	UC-05
Nama	Menampilkan notifikasi pemesanan tiket
Tujuan	Untuk mengetahui apabila ada pemberitahuan customer sudah melakukan pembayaran pemesanan tiket.
Deskripsi	Menampilkan notifikasi pemesanan tiket adalah <i>use case</i> yang digunakan untuk mengetahui pemberitahuan masuk adanya pembayaran pemesanan tiket yang sudah dilakukan oleh customer.
Tipe	High level
Aktor	Admin, kasir
<b>Skenario Utama</b>	
Kondisi awal	Ada notifikasi pemesanan yang masuk
<b>Aksi aktor</b>	
<b>Reaksi sistem</b>	

1. Aktor mengklik icon lonceng	2. Sistem menampilkan list pemesanan tiket yang masuk
3. Memilih data pada list pemesanan	4. Sistem mengarahkan halaman ke halaman konfirmasi pembayaran
Kondisi akhir	Masuk ke halaman konfirmasi pembayaran

Table 4.11 Skenario *Use Case* Memverifikasi Pemesanan Tiket

Identifikasi	
No	UC-06
Nama	Memverifikasi pemesanan tiket
Tujuan	Untuk memverifikasi QR Code dari tiket yang dibawa customer
Deskripsi	Memverifikasi pemesanan tiket adalah <i>use case</i> yang digunakan untuk menverifikasi tiket dari customer dengan cara memindai QR Code yang ada pada tiket.
Tipe	High level
Aktor	Petugas penarik tiket
Skenario Utama	
Kondisi awal	Memindai QR Code tiket
Aksi aktor	
1. Petugas penarik tiket menekan tombol scan	2. Sistem memunculkan kamera untuk scan QR Code
3. Petugas tiket melakukan scan QR Code	4. Sistem menampilkan pesan hasil dari scan QR Code
Kondisi akhir	QR Code sudah di scan

Table 4.12 Skenario *Use Case* Form Konfirmasi Pembayaran

Identifikasi	
No	UC-07
Nama	Menginput form konfirmasi pembayaran
Tujuan	Untuk melakukan konfirmasi pembayaran dengan mengisi form konfirmasi pembayaran
Deskripsi	Menginput form konfirmasi pembayaran adalah <i>use case</i> yang digunakan untuk melakukan konfirmasi pembayaran yang dilakukan oleh customer yang melakukan pemesanan melalui aplikasi mobile.
Tipe	High level
Aktor	Customer, Admin
Skenario Utama	
Kondisi awal	Membuka menu list transaksi pembayaran
Aksi aktor	
1. Customer memilih menu list transaksi pemesanan	2. Sistem menampilkan halaman list transaksi pemesanan
3. Customer menekan tombol view bukti pembayaran	4. Sistem memeriksa batas pembayaran
5. Customer menekan tombol konfirmasi pembayaran	6. Sistem menampilkan halaman konfirmasi pembayaran
7. Customer mengupload bukti pembayaran	9. Sistem mengupdate status pembayaran
8. Admin mengecek bukti pembayaran	10. Sistem menampilkan pemberitahuan
11. Customer menerima pemberitahuan	
Kondisi akhir	Form konfirmasi pemesanan sudah terisi

Table 4.13 Skenario Use Case Mengelola Akun Karyawan

Identifikasi	
No	UC-08
Nama	Mengelola akun karyawan
Tujuan	Untuk mengelola akun karyawan yang menggunakan aplikasi web.
Deskripsi	Mengelola akun karyawan adalah use case yang digunakan untuk mengelola akun-akun dari karyawan yang menggunakan aplikasi web.
Tipe	High level
Aktor	Admin
Skenario Utama	
Kondisi awal	Memilih menu kelola pengguna
Aksi actor	Reaksi sistem
1. Admin memilih menu kelola pengguna	2. Sistem menampilkan halaman kelola pengguna
3. Admin mengelola akun karyawan	4. Sistem memproses data yang dilakukan oleh customer
Kondisi akhir	Admin selesai mengelola akun karyawan

Table 4.14 Skenario Use Case Menginput Akun Customer

Identifikasi	
No	UC-09
Nama	Menginput akun customer
Tujuan	Untuk menginput akun customer ketika registrasi di aplikasi <i>mobile</i> .
Deskripsi	Menginput akun customer adalah use case yang digunakan untuk menginput akun customer ketika registrasi di aplikasi <i>mobile</i> untuk kepentingan pemesanan tiket.
Tipe	High level
Aktor	Customer
Skenario Utama	
Kondisi awal	Mengisi form pendaftaran
Aksi actor	Reaksi sistem
1. Customer memilih menu <i>registrasi</i>	2. Sistem menampilkan halaman resgitrasi
3. Customer mengisi form <i>registrasi</i>	4. Sistem menyimpan <i>registrasi</i> customer
5. Customer menerima pesan pemberitahuan pemesanan	
Kondisi akhir	Form registrasi sudah terisi

Table 4.15 Skenario Use Case Mengelola Content Berita

Identifikasi	
No	UC-10
Nama	Mengelola content berita
Tujuan	Untuk menampilkan berita di aplikasi <i>mobile</i> .
Deskripsi	Mengelola content berita adalah use case yang digunakan untuk mengelola content berita yang akan digunakan di aplikasi <i>mobile</i> .
Tipe	High level
Aktor	Admin
Skenario Utama	
Kondisi awal	Mengisi form content berita

Aksi actor	Reaksi sistem
1. Admin memilih menu konfigurasi <i>mobile</i>	2. Sistem menampilkan halaman konfigurasi <i>mobile</i>
3. Admin mengelola content berita	4. Sistem memproses data dari admin
5. Admin menerima pemberitahuan	
Kondisi akhir	Form content berita terisi

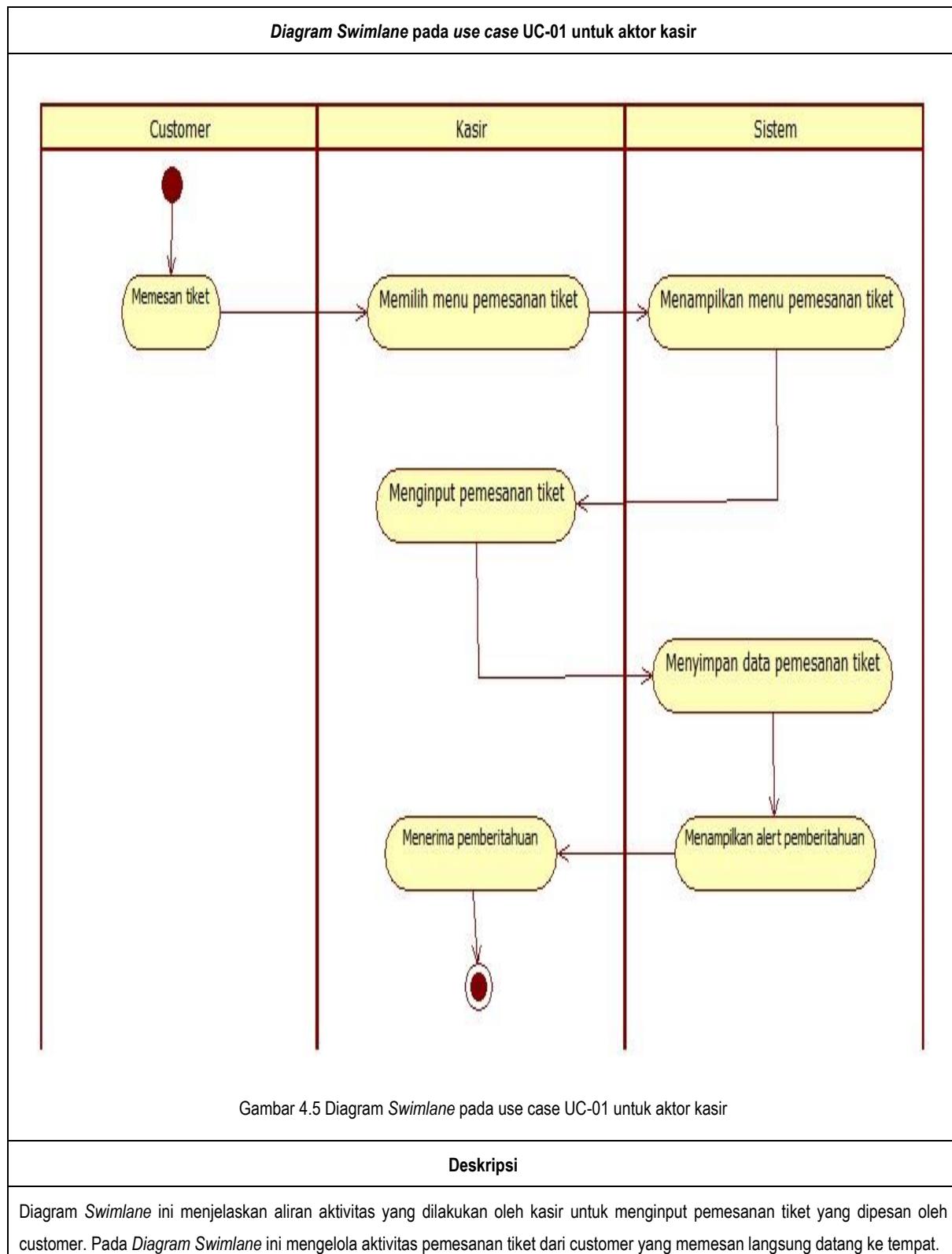
Table 4.16 Skenario Use Case Menampilkan Rekap Data Penjualan Tiket

Identifikasi	
No	UC-11
Nama	Menampilkan rekap data penjualan tiket
Tujuan	Untuk menampilkan data rekap penjualan tiket
Deskripsi	Menampilkan rekap data penjualan tiket adalah use case yang digunakan untuk menampilkan rekapitulasi data penjualan tiket setiap harinya
Tipe	High level
Aktor	Admin
Skenario Utama	
Kondisi awal	Membuka menu dashboard
Aksi actor	Reaksi sistem
1. Admin memilih menu dashboard	2. Sistem menampilkan halaman dashboard
4. Admin melihat jumlah grafik dan jumlah penjualan tiket	3. Sistem menampilkan grafik dan jumlah penjualan tiket per harinya
Kondisi akhir	Admin melihat rekap penjualan tiket

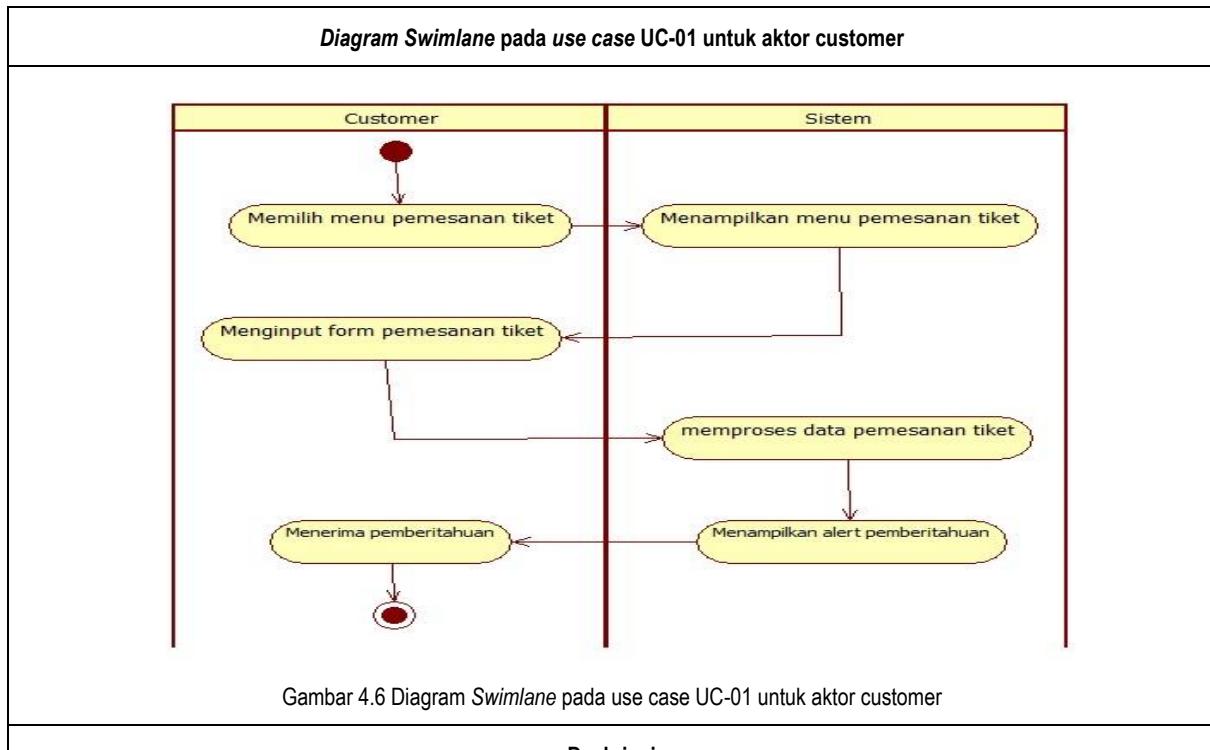
#### 4.2.5.3 Aliran Aktivitas

Aliran aktivitas digunakan untuk mengetahui alur dari aktivitas-aktivitas yang ada. Untuk membuat aliran aktivitas ini akan digunakan diagram *Swimlane*. Diagram *Swimlane* menjelaskan aliran aktivitas yang dijelaskan oleh use case dan pada saat yang sama menunjukkan aktor mana (jika ada beberapa aktor yang terlibat dalam use case tertentu) atau kelas analisis sebagai tanggung jawab atas tindakan yang dijelaskan oleh sebuah aktivitas [PRE10]. Berikut ini adalah beberapa Diagram *Swimlane* yang dirancang untuk aplikasi berbasis web dan berbasis *mobile*. Dapat dilihat pada gambar-gambar dibawah ini.

1. Diagram *Swimlane* pada *use case* UC-01 untuk aktor kasir



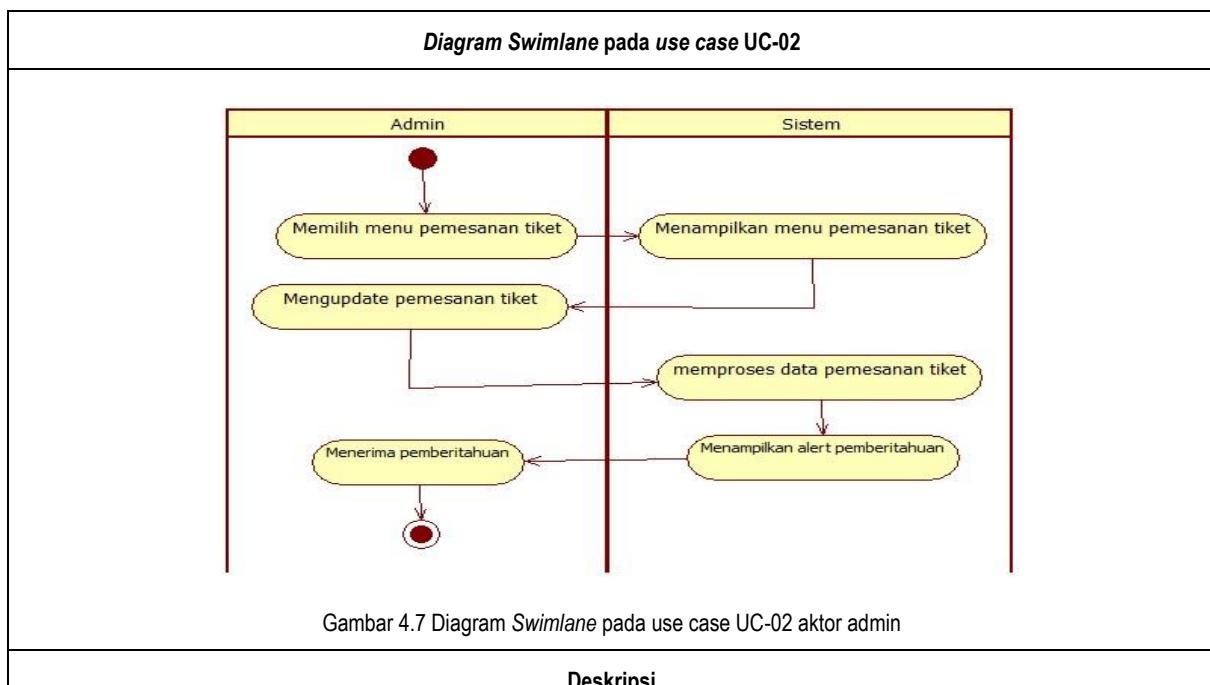
2. Diagram *Swimlane* pada *use case* UC-01 untuk aktor customer



**Deskripsi**

Diagram *Swimlane* ini menjelaskan aliran aktivitas yang dilakukan oleh customer pemesanan tiket secara online. Pada Diagram *Swimlane* ini menginput aktivitas pemesanan tiket dari customer yang memesan tiket secara online melalui aplikasi *mobile*.

3. Diagram *Swimlane* pada *use case* UC-02

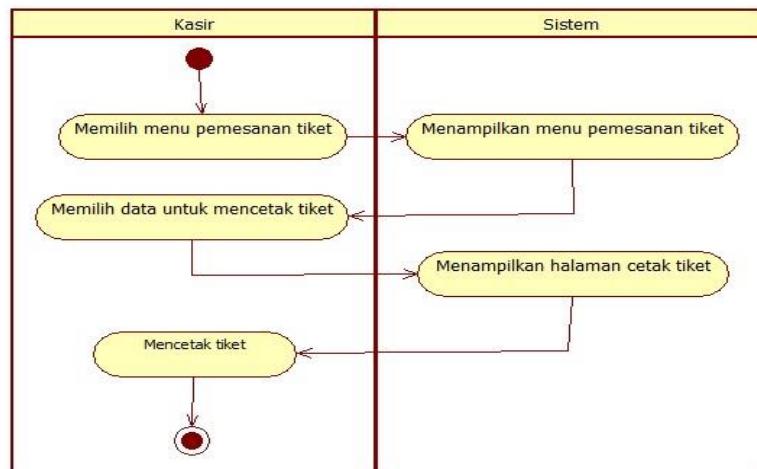


**Deskripsi**

Diagram *Swimlane* ini menjelaskan aliran aktivitas yang dilakukan oleh admin untuk mengupdate pemesanan tiket apabila kasir ada kesalahan dalam menginput pemesanan tiket.

#### 4. Diagram *Swimlane* pada *use case* UC-03

Diagram *Swimlane* pada *use case* UC-03



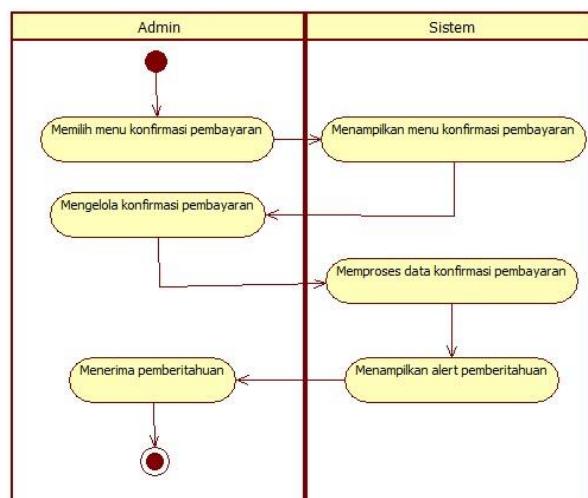
Gambar 4.8 Diagram *Swimlane* pada *use case* UC-03

#### Desripsi

Diagram *Swimlane* ini menjelaskan aliran aktivitas yang dilakukan oleh kasir mencetak tiket yang telah dipesan oleh customer.

#### 5. Diagram *Swimlane* pada *use case* UC-04

Diagram *Swimlane* pada *use case* UC-04

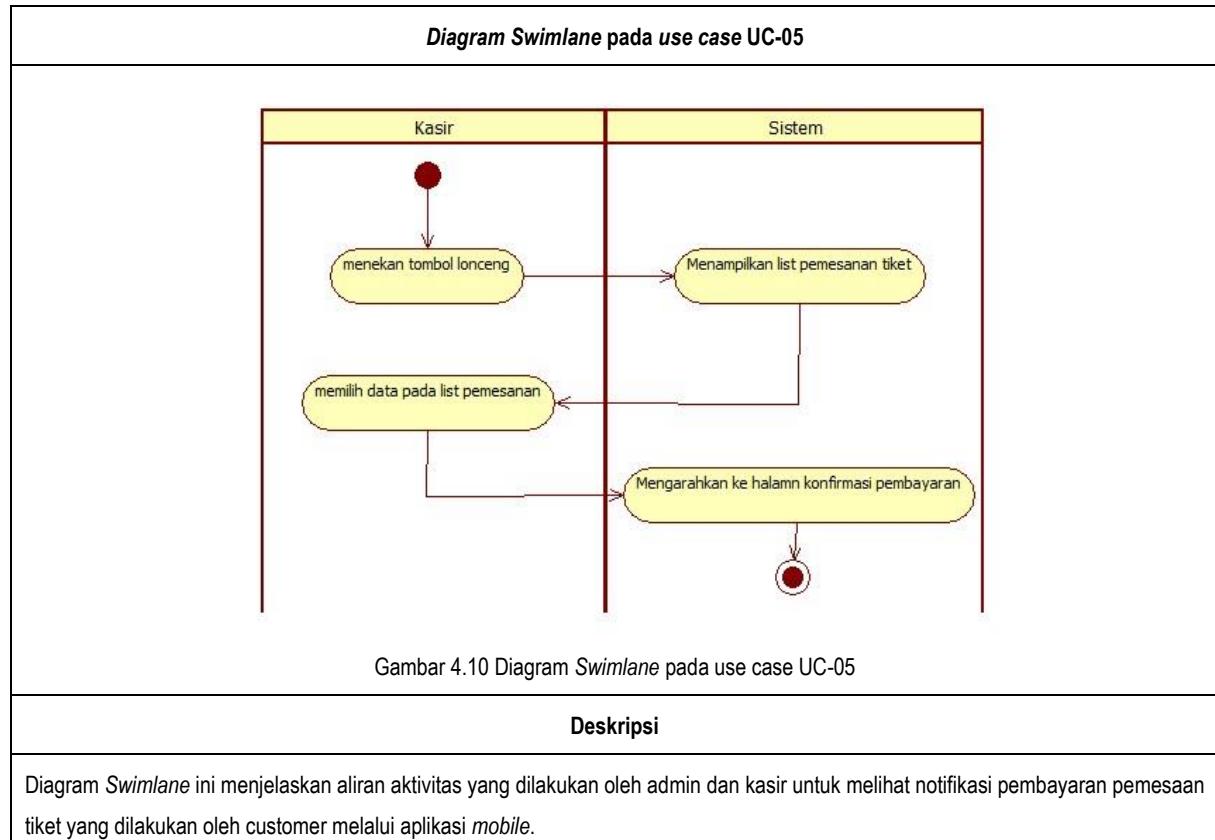


Gambar 4.9 Diagram *Swimlane* pada *use case* UC-04

#### Desripsi

Diagram *Swimlane* ini menjelaskan aliran aktivitas yang dilakukan oleh admin untuk mengelola konfirmasi pembayaran. Yang terdiri dari mengupdate konfirmasi pembayaran dan menghapus konfirmasi pembayaran.

#### 6. Diagram *Swimlane* pada *use case* UC-05



#### 7. Diagram *Swimlane* pada *use case* UC-06

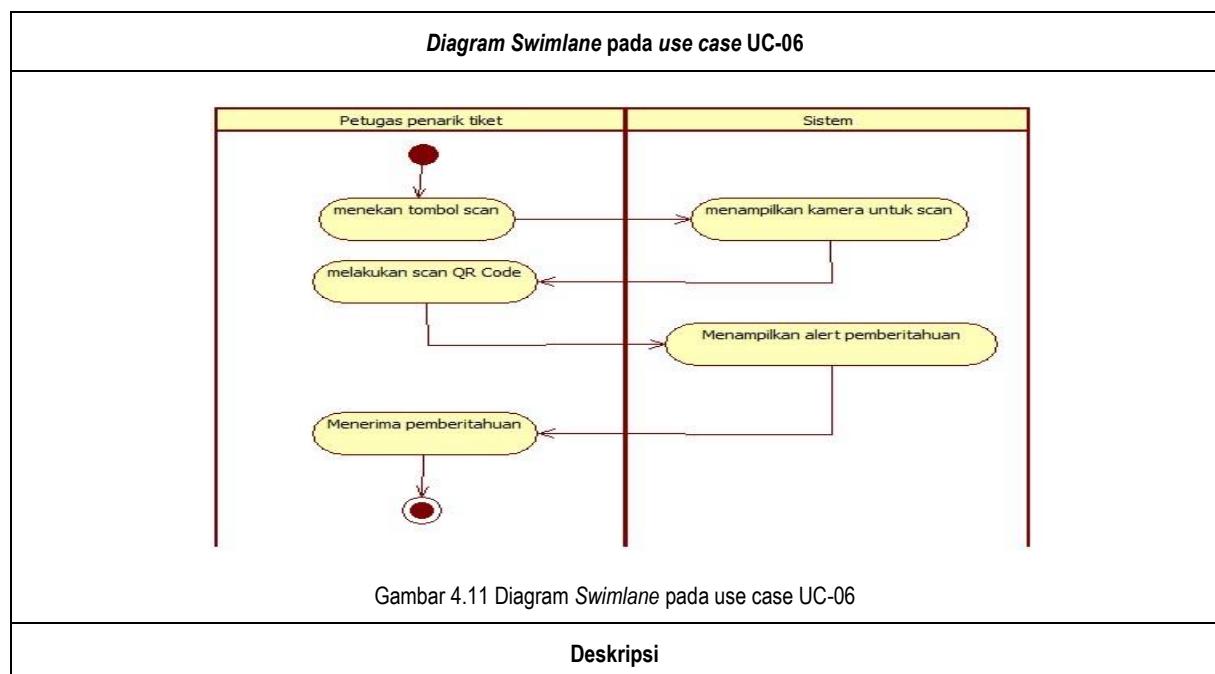
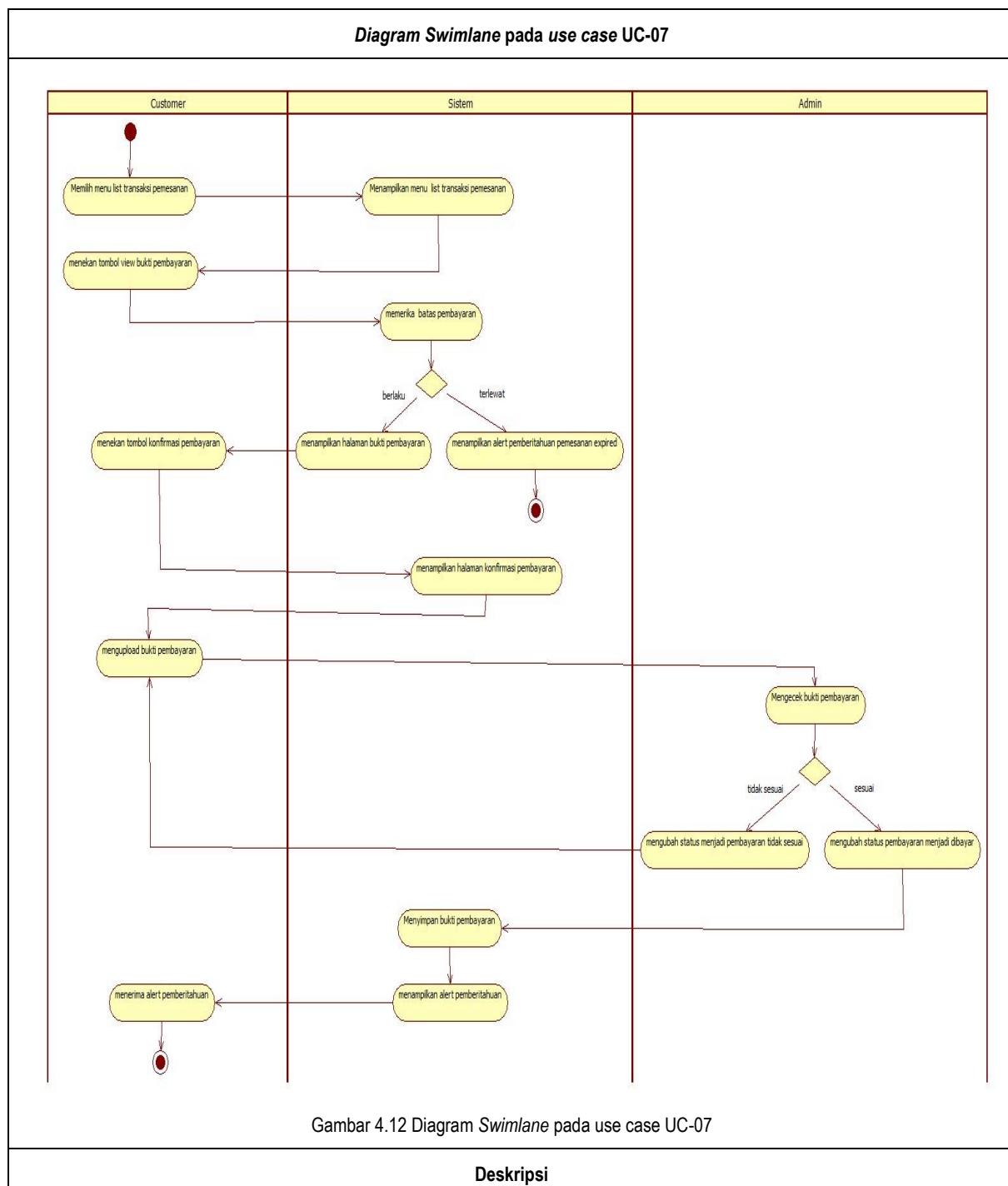


Diagram *Swimlane* ini menjelaskan aliran aktivitas yang dilakukan oleh petugas penarik tiket untuk melakukan pemindai QR Code pada tiket yang dibawa oleh customer.

## 8. Diagram *Swimlane* pada *use case* UC-07

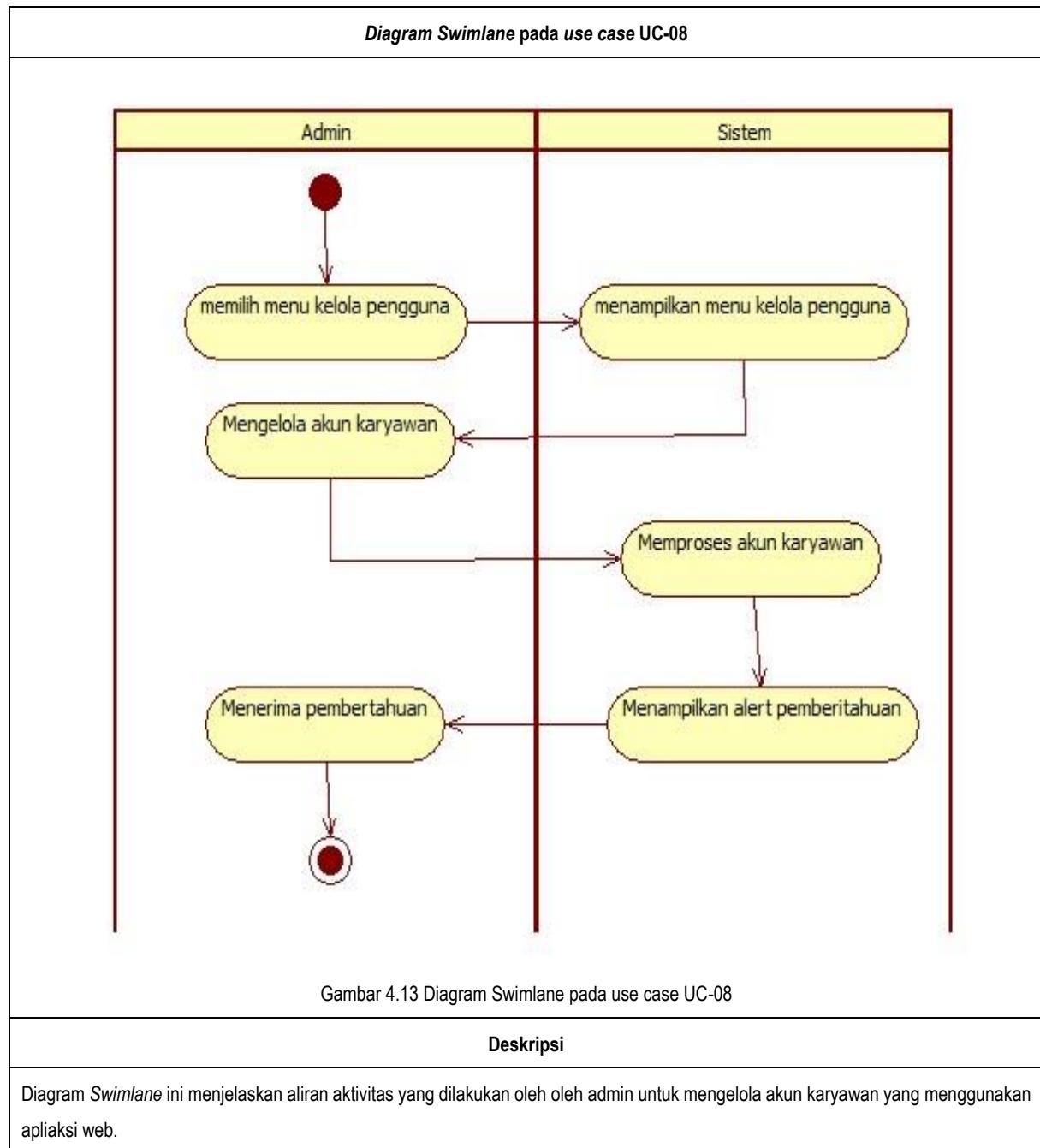


Gambar 4.12 Diagram *Swimlane* pada *use case* UC-07

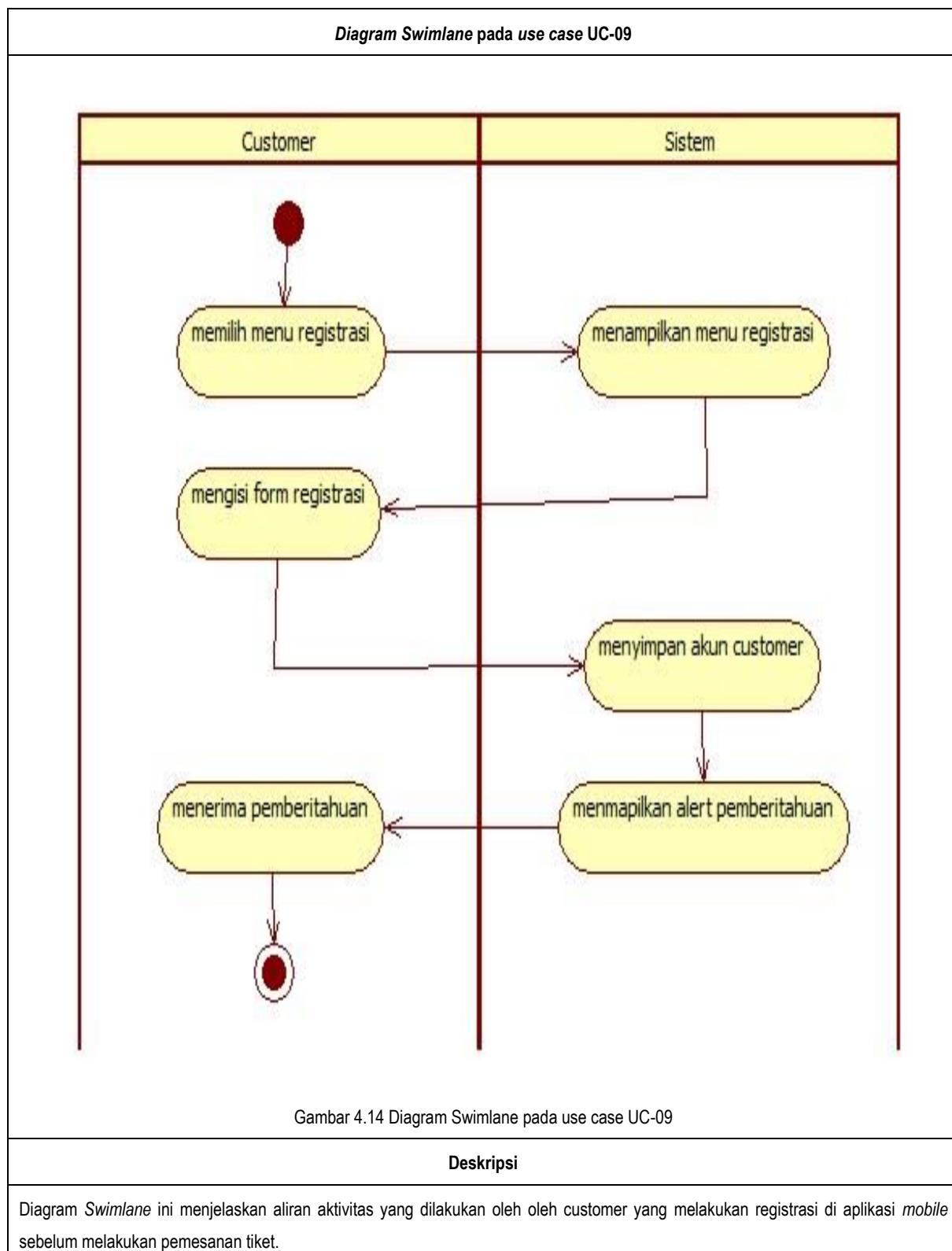
**Deskripsi**

Diagram *Swimlane* ini menjelaskan aliran aktivitas yang dilakukan oleh customer untuk melakukan konfirmasi pembayaran yang telah memesan tiket. Dimana setelah memesan tiket customer harus mengupload bukti pembayaran sebelum waktu yang telah ditentukan. Apabila belum mengupload bukti pembayaran dan waktu batas pembayaran sudah terlewat maka pemesanan akan dibatalkan dan statusnya menjadi expired. Jika bukti pembayaran yang diupload tidak sesuai atau bukti pembayaran telah sesuai tetapi jumlah pembayaran tidak sesuai, maka admin akan mengubah status pembayaran menjadi tidak sesuai dan customer diharuskan memperbaiknya serta mengupload ulang bukti pembayaran sebelum batas pembayaran berakhir.

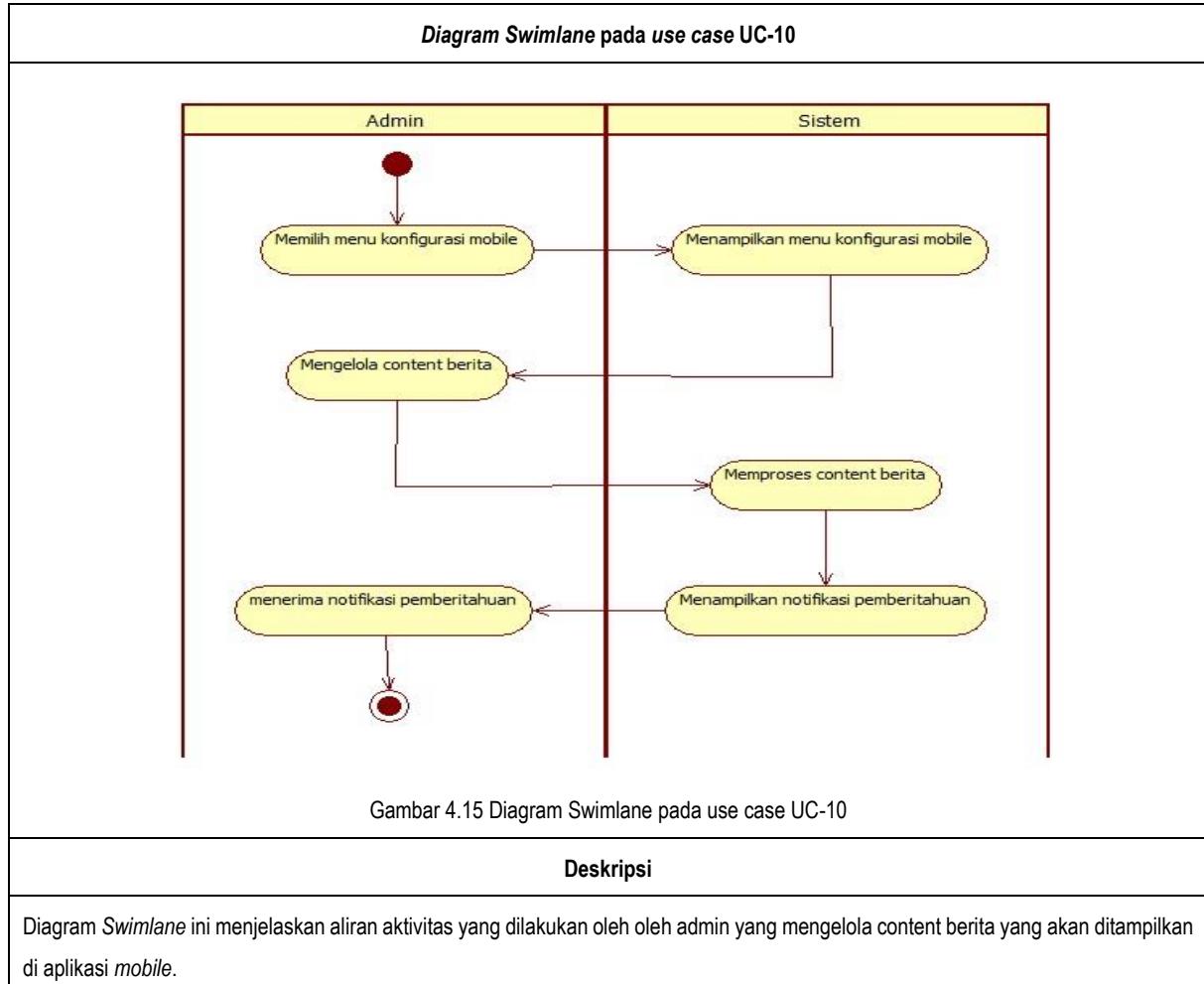
#### 9. Diagram *Swimlane* pada use case UC-08



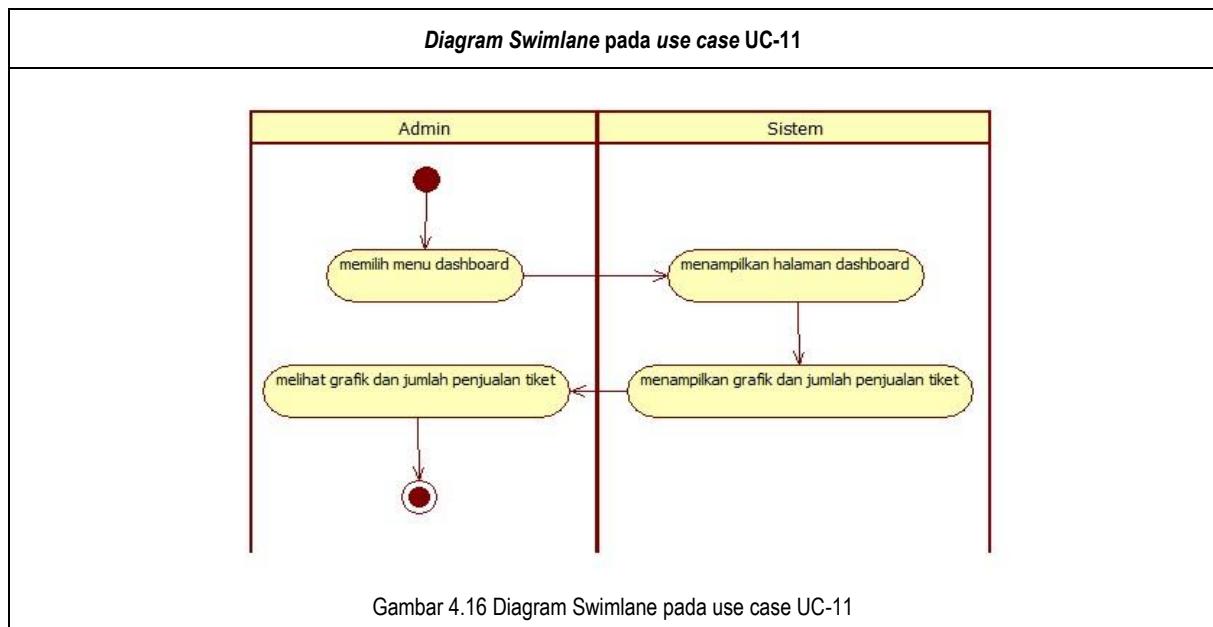
10. Diagram *Swimlane* pada use case UC-09



11. Diagram *Swimlane* pada use case UC-10



12. Diagram *Swimlane* pada use case UC-11



Deskripsi
Diagram <i>Swimlane</i> ini menjelaskan aliran aktivitas yang dilakukan oleh admin atau kasir untuk melihat rekapitulasi data tiket yang terjual.

#### 4.2.6 Pemodelan Data

Kebutuhan-kebutuhan perangkat lunak mencakup di dalamnya kebutuhan-kebutuhan untuk membuat, memperluas, atau bersinggungan dengan basis data. Untuk itu, harus didefinisikan semua objek data yang akan diproses di dalam sistem atau perangkat lunak, mendefinisikan relasi antar objek, dan mendefinisikan informasi lain terkait relasi [PRE10].

##### 4.2.6.1 Objek Data

Sebuah objek data sesungguhnya adalah representasi dari informasi komposit yang harus dipahami oleh perangkat lunak. Memiliki sejumlah property atau atribut yang berbeda. Suatu objek data dapat saja berupa entitas eksternal (sesuatu yang menghasilkan atau menggunakan informasi), laporan atau tampilan, panggilan telepon atau suatu alarm, suatu peran misalnya wiraniaga, suatu unit organisasi, suatu tempat, atau suatu struktur contohnya berkas [PRE10]. Berikut ini adalah objek data yang akan dibuat, dapat dilihat pada Table 4.17.

Table 4.17 Objek Data

No.	Nama Objek	Deskripsi
1.	Role Level	Objek ini digunakan untuk menentukan hak akses setiap pengguna aplikasi web.
2.	Jenis Pemesanan	Objek ini digunakan untuk menentukan jenis pemesanan apakah melakukan pemesanan secara langsung atau melalui aplikasi mobile.
3.	Konfirmasi Pembayaran	Objek ini digunakan untuk mengelola pembayaran pemesanan tiket dari customer, apakah sudah melakukan pembayaran atau belum.
4.	Menu	Objek ini digunakan untuk mengelola menu yang akan digunakan pada aplikasi web.
5.	Pemesanan tiket	Objek ini digunakan untuk mengelola pemesanan tiket yang dilakukan oleh customer.
6.	Status Pembayaran	Objek ini digunakan untuk mengetahui apakah customer pembayarannya sudah dikonfirmasi atau masih belum melakukan pembayaran.
7.	User Customer	Objek ini digunakan untuk mengelola akun dari customer.
8.	Karyawan	Objek ini digunakan untuk mengelola akun dari karyawan.
9.	User Level	Objek ini digunakan untuk menentukan level dari user yang menggunakan aplikasi web.
10.	User Karyawan	Objek ini digunakan untuk membuat akun karyawan yang berelasi dengan objek karyawan.
11.	Role User Karyawan	Objek ini digunakan untuk menyimpan data user yang memiliki level yang telah ditentukan

##### 4.2.6.2 Atribut Data

Atribut data pada dasarnya mendefinisikan property-properti yang dimiliki oleh objek data dan pada saranya memiliki tiga karakteristik yang berbeda. Pertama, memberi nama objek dari objek data, mendeskripsikan objek, atau merujuk pada objek lain yang ada pada table yang lain. Berikut ini adalah daftar atribut data dari objek data yang sudah dibuat sebelumnya. Dapat dilihat pada Table 4.18.

Table 4.18 Atribut Data

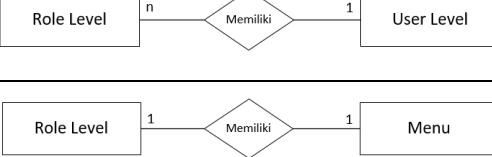
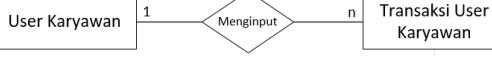
No.	Objek Data	Atribut Data	Deskripsi
1.	Role Level	id	Atribut ini digunakan untuk primary key pada table.
		id_user_level	Atribut ini digunakan untuk user yang dapat mengakses menu berdasarkan level
		id_menu	Atribut ini relasi dari table menu untuk dapat diakses oleh user level yang mendapatkan akses.
		create	Atribut ini untuk mengecek user boleh melakukan tambah data.
		read	Atribut ini untuk mengecek user boleh melakukan lihat data.
		update	Atribut ini untuk mengecek user boleh melakukan update data.
		delete	Atribut ini untuk mengecek user boleh melakukan hapus data.
2.	Jenis Pemesanan	id	Atribut ini digunakan untuk primary key pada table.
		nama_jenis	Atribut ini digunakan memberi nama jenis pemesanan.
3.	Konfirmasi Pembayaran	id	Atribut ini digunakan untuk primary key pada table.
		kode_pemesanan	Atribut ini berelasi dengan kode pemesanan di table pemesanan tiket untuk mengecek pembayaran.
		id_status	Atribut ini digunakan untuk mengecek status yang berelasi dengan table status pembayaran
		bukti_pembayaran	Atribut ini digunakan untuk menyimpan bukti pembayaran dari customer
		batas_pembayaran	Atribut ini digunakan untuk menyimpan batas waktu pembayaran tiket
		id_karyawan	Atribut ini digunakan untuk mengetahui siapa yang telah menggunakan aplikasi
4.	Menu	id_menu	Atribut ini digunakan untuk primary key pada table.
		title	Atribut ini digunakan untuk memberi nama pada menu.
		url	Atribut ini digunakan untuk memberi nama url pada menu.
		icon	Atribut ini digunakan untuk memberi logo pada menu.
		is_main_menu	Atribut ini digunakan untuk mengecek apakah menu itu disimpan di menu utama atau jadi sub menu.
		is_aktif	Atribut ini digunakan untuk menentukan menu itu aktif atau tidak.
		order_num	Atribut ini digunakan untuk menyusun urutan menu.
5.	Pemesanan Tiket	id	Atribut ini digunakan untuk primary key pada table.
		kode_pemesanan	Atribut ini digunakan sebagai penanda transaksi yang masuk.
		tgl_pemesanan	Atribut ini digunakan untuk mengetahui kapan tanggal pemesanan.
		tgl_masuk	Atribut ini digunakan untuk mengetahui kapan tanggal masuk untuk pemesanan dari aplikasi mobile.
		id_karyawan	Atribut ini digunakan untuk mengetahui siapa yang melayani customer
		jumlah_tiket	Atribut ini digunakan untuk mengetahui berapa jumlah tiket.
		total_uang_masuk	Atribut ini digunakan untuk mengetahui berapa jumlah uang yang masuk.
		uang_pembayaran	Atribut ini digunakan untuk mengetahui uang yang diberikan oleh customer untuk pembayaran.
		status_penggunaan	Atribut ini digunakan untuk mengetahui pemesanan sudah digunakan atau belum.
		id_jenis	Atribut ini digunakan untuk mengetahui jenis pemesanan yang masuk.

No.	Objek Data	Atribut Data	Deskripsi
		id_customer	Atribut ini digunakan untuk mengetahui siapa customer yang melakukan pemesanan.
		qr_code	Atribut ini digunakan untuk keperluan scan saat akan masuk ke kolam renang.
6.	Status Pembayaran	id	Atribut ini digunakan untuk primary key pada table.
		nama_status	Atribut ini digunakan untuk memberi nama status pembayaran untuk keperluan konfirmasi pembayaran.
7.	User Customer	nama	Atribut ini digunakan untuk menyimpan nama customer.
		Username	Atribut ini digunakan untuk menyimpan username customer.
		email	Atribut ini digunakan untuk authentikasi saat login.
		password	Atribut ini digunakan untuk authentikasi saat login.
		no_telepon	Atribut ini digunakan untuk menyimpan nomor telepon customer.
		images	Atribut ini digunakan untuk menambahkan gambar untuk profil.
		id_customer	Atribut ini digunakan untuk primary key pada table.
		api_token	Atribut ini digunakan untuk menyimpan token
		player_id	Atribut ini digunakan untuk menyimpan id onesignal.
8.	Karyawan	id	Atribut ini digunakan untuk primary key pada table.
		nama	Atribut ini digunakan untuk menyimpan nama karyawan.
		email	Atribut ini digunakan untuk authentikasi saat login.
		no_hp	Atribut ini digunakan untuk menyimpan no telepon karyawan.
9.	User Level	id	Atribut ini digunakan untuk primary key pada table.
		nama_level	Atribut ini digunakan untuk memberi nama level untuk keperluan pemberian hak akses untuk karyawan.
10.	Role User Karyawan	id	Atribut ini digunakan untuk primary key pada table.
		id_karyawan	Atribut ini digunakan untuk merelasikan dengan table karyawan untuk keperluan login.
		id_user_level	Atribut ini digunakan untuk menyimpan level yang dimiliki karyawan.
11.	User Karyawan	id	Atribut ini digunakan untuk merelasikan dengan table karyawan untuk mengambil data karyawan.
		id_karyawan	Atribut ini digunakan untuk merelasikan dengan table karyawan untuk mengambil data karyawan.
		username	Atribut ini digunakan untuk login ke aplikasi
		password	Atribut ini digunakan untuk login ke aplikasi
		images	Atribut ini digunakan untuk menambahkan gambar untuk profil.
		status	Atribut ini digunakan untuk menentukan user itu aktif atau tidak.
		api_token	Atribut ini digunakan untuk menyimpan token

#### 4.2.6.3 Relasi

Objek-objek data saling terhubung satu dengan yang lainnya dengan menggunakan berbagai cara yang berbeda [PRE10]. Berikut ini relasi antar objek yang akan dibuat, dapat dilihat pada Table 4.19.

Table 4.19 Relasi Antar Objek

No.	Objek Data	Relasi Data	Kardinalitas	Deskripsi
1.	Hak Akses		one to many	Relasi ini digunakan untuk user level yang mempunyai banyak hak akses terhadap menu di aplikasi.
			one to one	Relasi ini digunakan satu menu untuk untuk setiap hak akses.
2.	Konfirmasi Pembayaran		one to one	Relasi ini digunakan untuk satu pemesanan tiket memiliki satu konfirmasi pembayaran
			one to one	Relasi ini bertujuan untuk menghubungkan satu konfirmasi pembayaran memiliki satu status pembayaran
			one to many	Relasi ini bertujuan untuk menghubungkan banyak konfirmasi pembayaran dapat diupdate oleh satu karyawan.
3.	Pemesanan Tiket		one to one	Relasi ini bertujuan untuk menghubungkan satu pemesanan tiket memiliki satu jenis pemesanan.
			one to many	Relasi ini bertujuan untuk satu customer yang bisa memiliki banyak pemesanan.
			one to many	Relasi ini bertujuan untuk mengetahui satu karyawan dapat menginput banyak pemesanan.
4.	User Karyawan		one to many	Relasi ini bertujuan untuk menghubungkan satu user karyawan memiliki banyak user level.
			one to many	Relasi ini bertujuan untuk menghubungkan satu user karyawan memiliki banyak transaksi user karyawan.
			one to one	Relasi ini bertujuan untuk menghubungkan satu user karyawan memiliki satu akun.

#### 4.2.7 Pemodelan Berbasis Kelas

Pemodelan berbasis kelas pada dasarnya memperlihatkan objek-objek yang akan dimanipulasi oleh sistem atau perangkat lunak, memperlihatkan operasi-operasi yang akan diterapkan pada objek-objek untuk menghasilkan umpan tertentu pada manipulasi objek, memperlihatkan relasi-relasi antar objek [PRE10]. Pada pemodelan berbasis kelas ini akan mengidentifikasi kelas-kelas analisis kemudian menentukan atribut dan perilaku setiap kelas dan terakhir akan membuat diagram kelas.

#### 4.2.7.1 Mengidentifikasi Kelas-Kelas Analisis

Proses mengidentifikasi kelas dengan cara memeriksa skenario penggunaan sistem atau perangkat lunak yang telah dikembangkan sebelumnya sebagai bagian dari model-model kebutuhan [PRE10]. Berikut ini adalah kelas-kelas analisis yang akan dibuat, bisa dilihat pada Table 4.20.

Table 4.20 Kelas-Kelas Analisis

No.	Nama Kelas	Jenis	Deskripsi
1.	PemesananController	Controller	Kelas ini digunakan untuk mengelola pemesanan tiket dari customer baik yang melakukan pemesanan secara langsung atau melalui aplikasi <i>mobile</i> .
2.	UserController	Controller	Kelas ini digunakan untuk mengelola akun karyawan yang menggunakan aplikasi web.
3.	KonfirmasiPembayaranController	Controller	Kelas ini digunakan untuk mengelola konfirmasi pembayaran dari customer apakah sudah melakukan pembayaran atau belum.
4.	KonfigurasiMobileController	Controller	Kelas ini digunakan untuk mengelola content berita yang akan ditampilkan di aplikasi <i>mobile</i> .
5.	KonfigurasiMobileModel	Model	Kelas ini digunakan untuk berinteraksi dengan <i>database</i> pada aplikasi web untuk mengambil data content berita.
6.	PemesananModel	Model	Kelas ini digunakan untuk berinteraksi dengan <i>database</i> pada aplikasi web untuk mengambil data pemesanan tiket.
7.	KonfirmasiPemesananModel	Model	Kelas ini digunakan untuk berinteraksi dengan <i>database</i> pada aplikasi web untuk mengambil data konfirmasi pembayaran.
8.	UserModel	Model	Kelas ini digunakan untuk berinteraksi dengan <i>database</i> pada aplikasi web untuk mengambil data user karyawan.
9.	KaryawanModel	Model	Kelas ini digunakan untuk berinteraksi dengan <i>database</i> pada aplikasi web untuk mengambil data karyawan.
10.	CustomerModel	Model	Kelas ini digunakan untuk berinteraksi dengan <i>database</i> pada aplikasi web untuk mengambil data customer.
11.	WebServicePengelolaanTiket	Model	Kelas ini digunakan sebagai <i>web service</i> yang berinteraksi dengan <i>database</i> .
12.	DetailPemesananActivity	Controller	Kelas ini digunakan sebagai controller untuk aplikasi <i>mobile</i> dalam menginput pemesanan tiket
13.	PembayaranActivity	Controller	Kelas ini digunakan sebagai controller untuk aplikasi <i>mobile</i> dalam menginput konfirmasi pembayaran pemesanan tiket yang dilakukan oleh customer
14.	ScanActivity	Controller	Kelas ini digunakan sebagai controller untuk aplikasi <i>mobile</i> dalam memindai QR Code.
15.	RegisterActivity	Controller	Kelas ini digunakan sebagai controller untuk aplikasi <i>mobile</i> untuk registrasi akun customer.
16.	PemesananView	View	Digunakan untuk keperluan tampilan pada halaman pemesanan tiket di browser.
17.	PembayaranView	View	Digunakan untuk keperluan tampilan pada halaman konfirmasi pembayaran di browser.
18.	UserView	View	Digunakan untuk keperluan tampilan pada halaman kelola pengguna di browser.
19.	KonfigurasiMobileView	View	Digunakan untuk keperluan tampilan pada halaman kelola content berita.
20.	Activity_detail_pemesanan	View	Digunakan untuk keperluan tampilan pada halaman pemesanan di aplikasi <i>mobile</i> .
21.	Activity_pembayaran	View	Digunakan untuk keperluan tampilan pada halaman konfirmasi pembayaran di aplikasi <i>mobile</i> .
22.	Activity_scan	View	Digunakan untuk keperluan tampilan pada halaman scan QR Code setelah login di aplikasi <i>mobile</i> .
23.	Activity_register	View	Digunakan untuk keperluan tampilan pada halaman registrasi akun customer.

#### 4.2.7.2 Menentukan Atribut-Atribut

Atribut mendeskripsikan kelas-kelas yang telah terpilih untuk dimasukan dalam model spesifikasi kebutuhan sistem atau perangkat lunak. Atribut-atribut merupakan sesuatu yang

mendefinikan kelas yang mengklasifikasikan makna suatu kelas dalam konteks ruang permasalahan yang telah diketahui sebelumnya. Berikut ini adalah atribut dari kelas yang di definisikan sebelumnya. Dapat dilihat pada Table 4.21.

Table 4.21 Atribut-Atribut Kelas

No.	Nama Kelas	Nama Atribut
1.	PemesananController	idJenis
		tiket
		uangPembayaran
		namaKasir
		qrCode
		id
		jumlahUang
2.	UserController	username
		karyawan
		password
		status
		id
3.	KonfirmasiPembayaranController	status
		pengubah
		deskripsi
		kode
		id
4.	KonfigurasiMobileController	title
		deskripsi
		images
		id
		insert
5.	KonfigurasiMobileModel	title
		deskripsi
		images
		id
		kodePemesanan
6.	PemesananModel	tglPemesanan
		tglMasuk
		IdKaryawan
		jumlahTiket
		totalUangMasuk
		uangPembayaran
		statusPenggunaan
		idCustomer
		idJenis
		qrCode
7.	KonfirmasiPembayaranModel	kodePemesanan
		idStatus
		buktiPembayaran
		batas_pembayaran
		idKaryawan
		komentar
8.	UserModel	idKaryawan
		username
		password
		images

No.	Nama Kelas	Nama Atribut
		status
9.	KaryawanModel	nama
		email
		noHp
10.	CustomerModel	nama
		email
		noHp
		username
		password
		images
11.	WebServicePengelolaanTiket	request
		user
		loginAs
		token
		customer
		nama
		noHp
		size
		username
		Konfigurasi
		email
		password
		kode
		images
		pemesanan
		idCustomer
		tiket
		jumlahUang
		uangPembayaran
		idJenis
		total
		playerId
		validator
12.	PembayaranActivity	imgHolder
		btnSubmit
		mApilInterface
		progressDialog
		imagePath
		idCustomer
		nama
		email
		username
		noHp
		images
		token
		kode
13.	DetailPemesananActivity	mApilInterface
		jmlTiketText
		totalText
		id
		nama
		email
		username

No.	Nama Kelas	Nama Atribut
		noHp images token kode btnTambah btnKurang cekPesanan harga jumlah btnOrder progressDialog
14.	ScanActivity	btnScan view intentIntegrator token mApilnterface mViewPager mSectionAdapter
15.	RegisterActivity	namaText usernameText emailText passwordText noHpText btnSubmit btnLogin mApilnterface playerId progressDialog
16.	PemesananView	jumlahTiket jumlahUang
17.	PembayaranView	status
18.	UserView	idKaryawan username password level status
19.	KonfigurasiMobileView	title deskripsi images
20.	Activity_detail_pemesanan	jumlahTiketText jumlahUangText
21.	Activity_pembayaran	images btnUpload
22.	Activity_scan	btnScan
23.	Activity_register	namaText usernameText emailText passwordText noHpText

#### 4.2.7.3 Mendefinisikan Operasi-Operasi

Operasi-operasi pada dasarnya mendefinisikan perilaku (*behavior*) suatu objek. Berikut ini adalah operasi-operasi dari setiap kelas yang telah didefinisikan sebelumnya. Dapat dilihat pada Table 4.22.

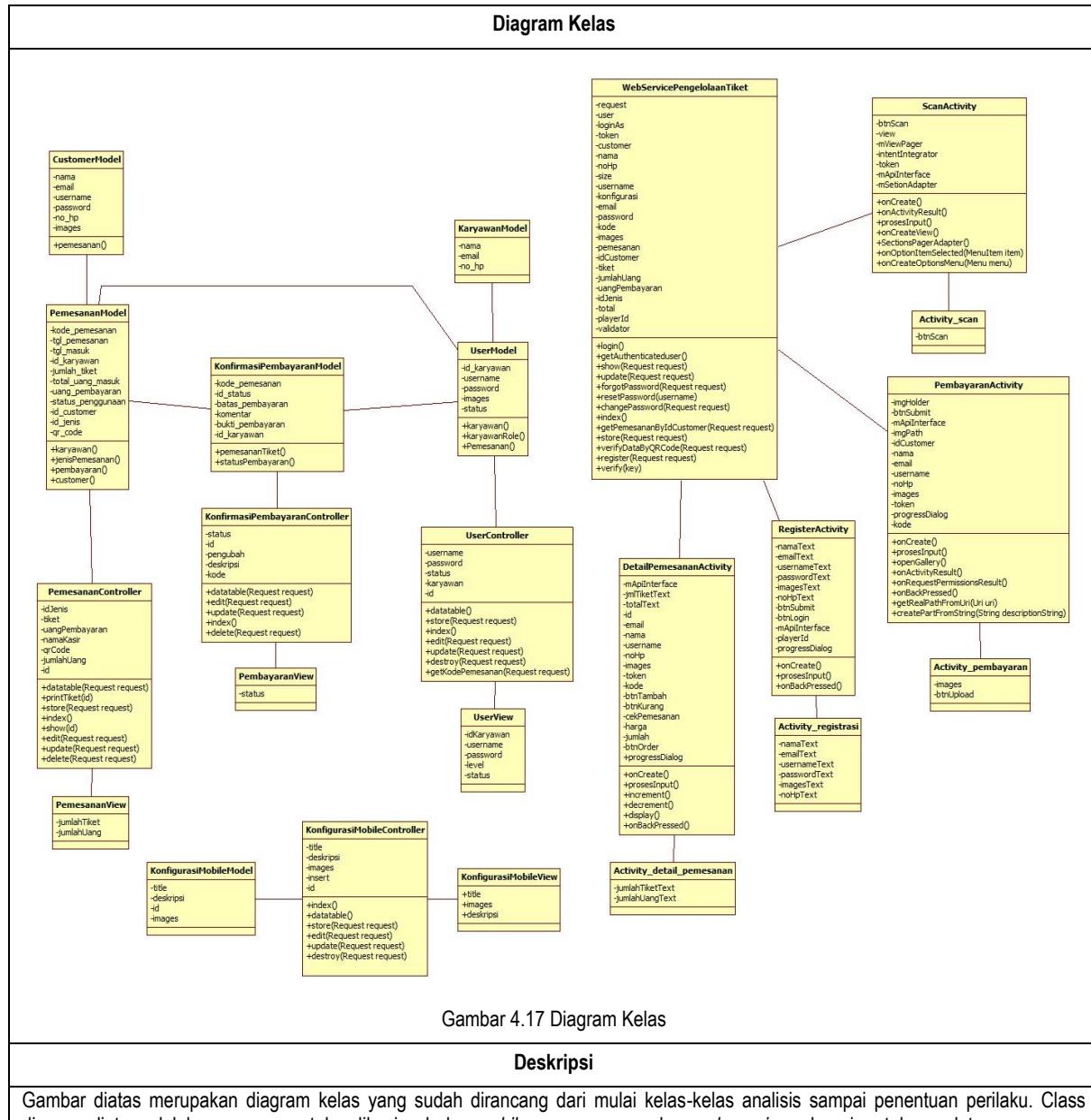
Table 4.22 Perilaku Pada Setiap Kelas

No.	Nama Kelas	Kandidat Perilaku	Nama Perilaku
1.	PemesananController	Menampilkan data pemesanan	datatable(Request request)
		Mengambil data untuk cetak tiket	printTiket(id)
		Menyimpan data pemesanan	store(Request request)
		Mengambil data pemesanan tiket	edit(Request request)
		Menampilkan detail pemesanan	show(id)
		Mengupdate data Pemesanan	update(Request request)
		Menghapus data pemesanan	destroy(Request request)
		Menampilkan halaman pemesanan	index()
2.	KonfirmasiPembayaranController	Menampilkan data konfirmasi pembayaran	datatable(Request request)
		Mengambil data konfirmasi pembayaran	edit(Request request)
		Mengupdate data konfirmasi pembayaran	update(Request request)
		Menghapus data konfirmasi pembayaran	destroy(Request request)
		Menampilkan halaman pembayaran	index()
3.	KonfigurasiMobileController	Menampilkan data content berita	datatable(Request request)
		Mengambil data content berita	edit(Request request)
		Mengupdate data content berita	update(Request request)
		Menghapus data content berita	destroy(Request request)
		Menampilkan halaman content berita	index()
		Menampilkan data content berita	datatable(Request request)
4.	KonfigurasiMobileModel	-	-
5.	UserController	Menampilkan data user	datatable()
		Mengambil data user	edit(Request request)
		Mengupdate data user	update(Request request)
		Menghapus data user	delete(Request request)
		Menyimpan data user	store(Request request)
		Menampilkan halaman kelola pengguna	index()
		Mengambil data berdasarkan kode pemesanan	getKodePemesanan(Request request)
6.	PemesananModel	Membuat relasi dengan kelas karyawan	karyawan()
		Membuat relasi dengan kelas jenisPemesanan	jenisPemesanan()
		Membuat relasi dengan kelas pembayaran	pembayaran()
		Membuat relasi dengan kelas customer	customer()
7.	KonfirmasiPembayaranModel	Membuat relasi dengan kelas pemesanan tiket	pemesananTiket()
		Membuat relasi dengan kelas status pembayaran	statusPembayaran()
8.	UserModel	Membuat relasi dengan kelas user level	karyawan()
		Membuat relasi dengan kelas pemesanan tiket	pemesanan()
		Membuat relasi dengan kelas choose role	karyawanRole()
9.	KaryawanModel	-	-
10.	CustomerModel	Membuat relasi dengan kelas pemesanan tiket	pemesanan()
11.	WebServicePengelolaanTiket	Mengecek akun user	login()
		Menampilkan data user yang login	getAuthenticateduser()
		Menampilkan data per id	show(Request request)
		Mengupdate data	update(Request request)

No.	Nama Kelas	Kandidat Perilaku	Nama Perilaku
12.	PembayaranActivity	Mengirim link ubah password	forgotPassword(Request request)
		Menampilkan halaman ubah password	resetPassword(username)
		Mengubah password user	changePassword(Request request)
		Mengambil data konfigurasi mobile	index()
		Mengambil data pemesanan berdasarkan id customer	getPemesananByldCustomer(Request request)
		Menyimpan data ke database	store(Request request)
		Memverifikasi data tiket menggunakan QR Code	verifyDataByQRCode(Request request)
		Menyimpan data akun customer	register(Request request)
		Memverifikasi akun customer	verify(key)
13.	DetailPemesananActivity	Menciptakan activity	onCreate()
		Menginput konfirmasi pembayaran	prosesInput()
		Membuka folder untuk upload gambar	openGallery()
		Menampilkan gambar yang dipilih	onActivityResult()
		Memindahkan ke halaman yang dituju	onBackPressed()
		Menangani respons permintaan izin	onRequestPermissionsResult()
		Mendapatkan nama gambar	getRealPathFromUri(Uri uri)
		Membuat request body	createPartFromString(String descriptionString)
14.	ScanActivity	Menciptakan activity	onCreate()
		Menginput pemesanan tiket	prosesInput()
		Menambah jumlah tiket	increment()
		Mengurangi jumlah tiket	decrement()
		Menampilkan jumlah tiket	display()
		Memindahkan ke halaman yang dituju	onBackPressed()
		Menciptakan fragment	onCreateView()
15.	RegisterActivity	Menampilkan hasil scan QR Code	onActivityResult()
		Menciptakan activity	onCreate()
		Membuat pager adapter	SectionsPagerAdapter()
		Meng-handle jika ada sebuah optionsmenu yang di-klik oleh User	onOptionsItemSelected(MenuItem item)
		Menambahkan optionsmenu tambahan pada ActionBar saat menu onCreate dipanggil	onCreateOptionsMenu(Menu menu)
		Menambahkan kode pemesanan pengganti QR Code	prosesInput()
16.	PemesananView	Menciptakan activity	onCreate()
		Menyimpan data customer	prosesInput()
		Memindahkan ke halaman yang dituju	onBackPressed()
17.	PembayaranView	-	-
18.	UserView	-	-
19.	KonfigurasiMobileView	-	-
20.	Activity_detail_pemesanan	-	-
21.	Activity_pembayaran	-	-
22.	Activity_scan	-	-
23.	Activity_register	-	-

#### 4.2.7.4 Perancangan Kelas

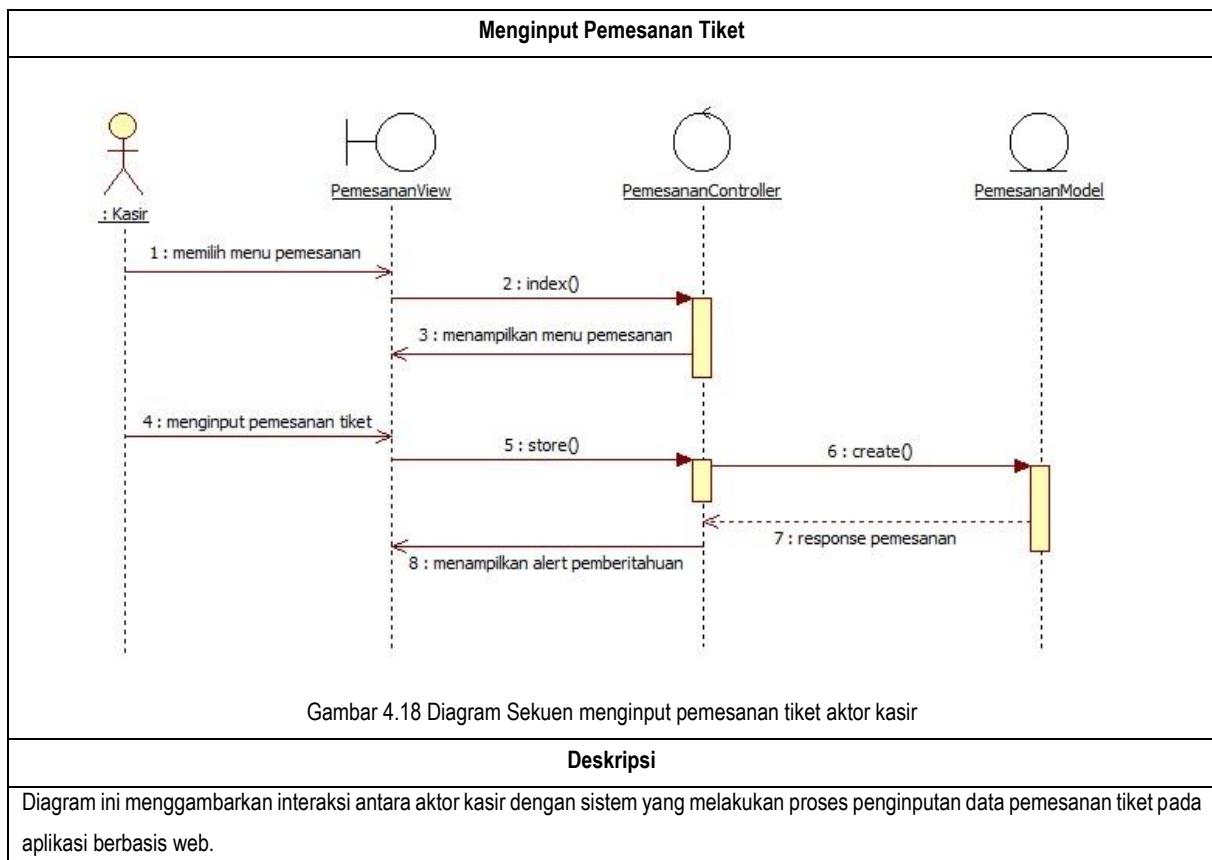
Perancangan kelas disini adalah merancang diagram kelas dari kelas-kelas yang telah didefinisikan sebelumnya beserta atribut dan perilakunya. Berikut ini adalah diagram kelas yang akan dibuat dapat dilihat pada Gambar 4.17.



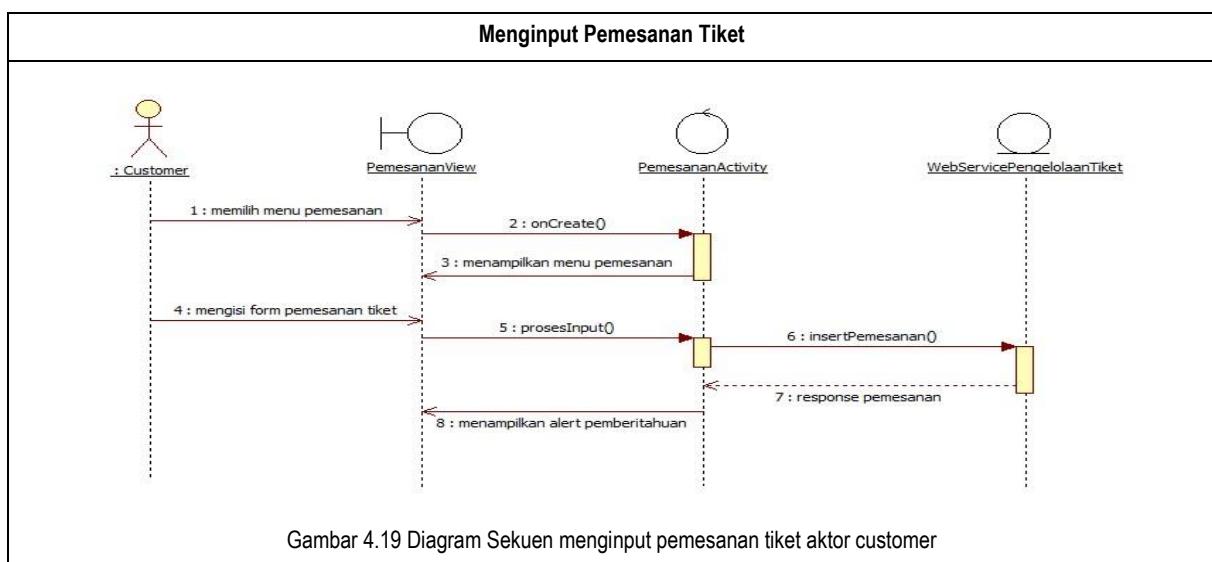
#### 4.2.8 Pemodelan Berbasis Perilaku

Model perilaku pada dasarnya menggambarkan bagaimana perangkat lunak akan berperilaku dalam menanggapi *event-event* yang datang dari arah luar atau bagaimana perangkat lunak akan berperilaku terhadap tindakan yang muncul dari luar [PRE10]. Pemodelan berbasis perilaku akan digambarkan dengan *sequence diagram*.

##### 1. Diagram Sekuen menginput pemesanan tiket aktor kasir



##### 2. Diagram Sekuen menginput pemesanan tiket aktor customer

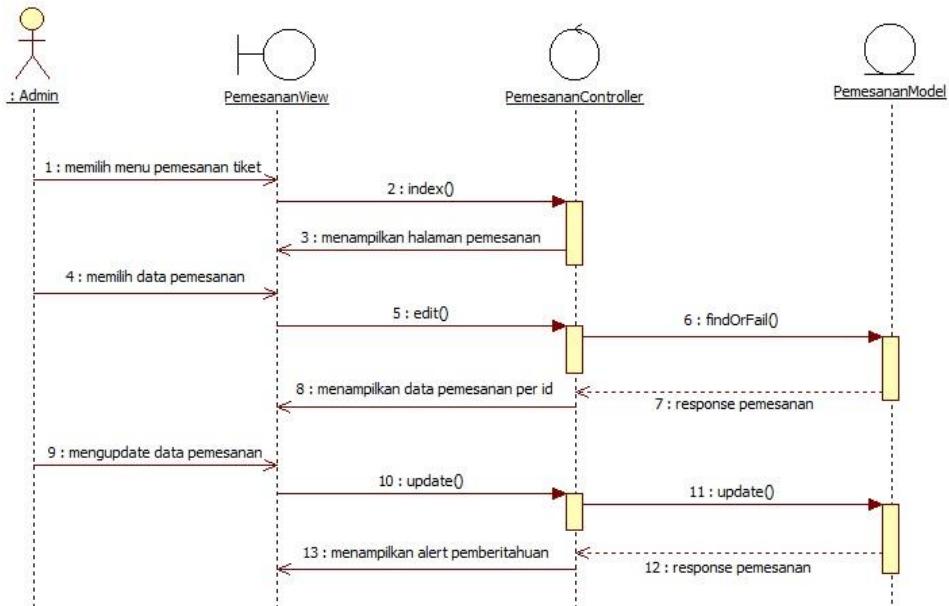


### Deskripsi

Diagram ini menggambarkan interaksi antara aktor customer dengan sistem yang melakukan proses penginputan data pemesanan tiket pada aplikasi berbasis mobile.

### 3. Diagram Sekuen mengupdate pemesanan tiket

Mengupdate Pemesanan Tiket



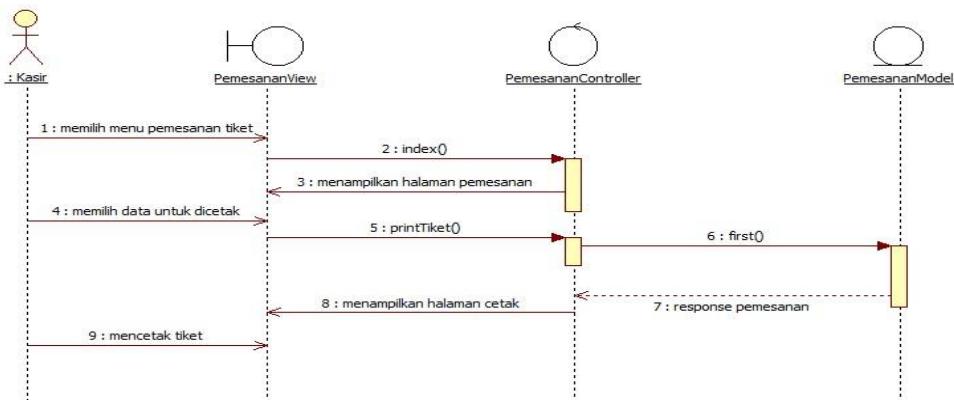
Gambar 4.20 Diagram Sekuen mengupdate pemesanan tiket

### Deskripsi

Diagram ini menggambarkan interaksi antara aktor admin dengan sistem yang melakukan proses update data pemesanan tiket pada aplikasi berbasis web.

### 4. Diagram Sekuen mencetak tiket

Mencetak Tiket



Gambar 4.21 Diagram Sekuen mencetak tiket

Deskripsi
Diagram ini menggambarkan interaksi antara aktor kasir dengan sistem yang melakukan proses mencetak tiket pada aplikasi berbasis web.

## 5. Diagram Sekuen mengelola konfirmasi pembayaran (update)

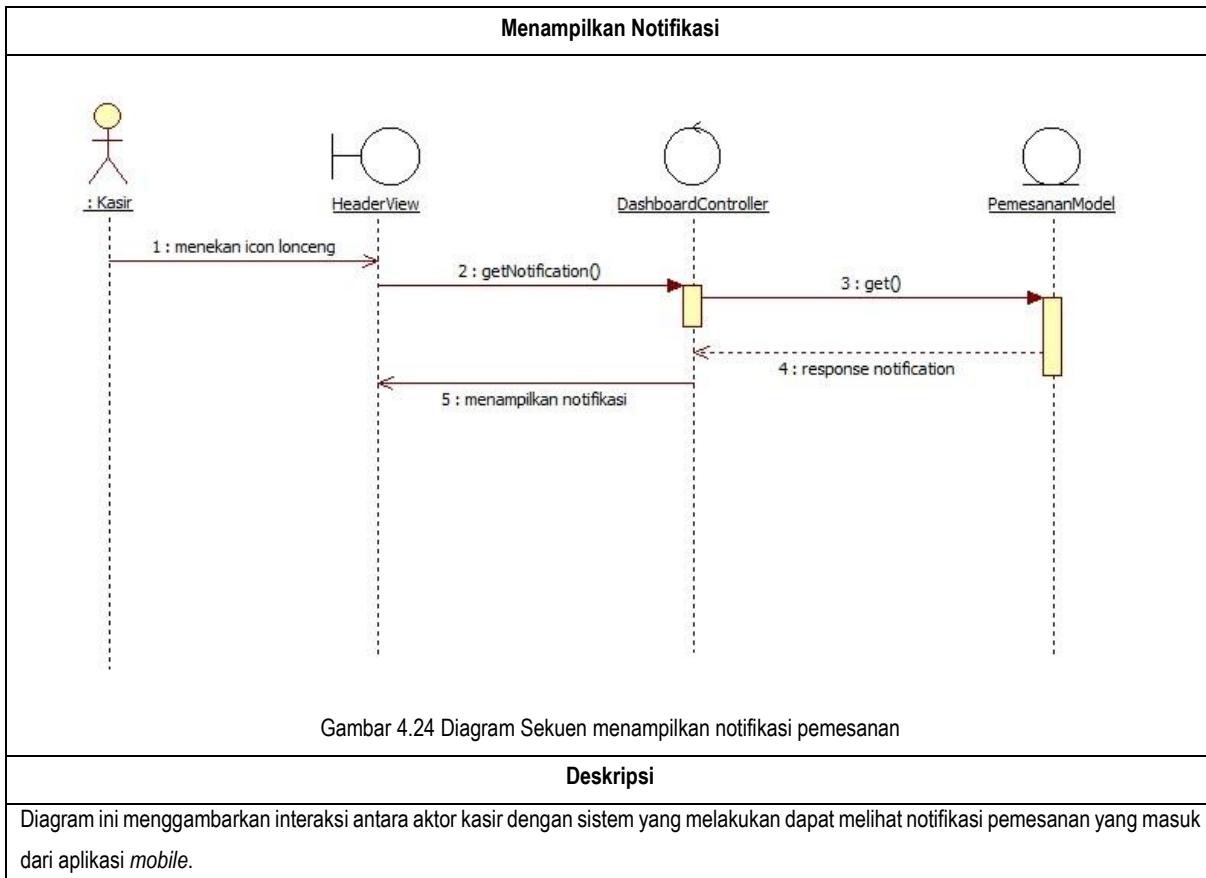
Mengelola Konfirmasi Pembayaran
<pre> sequenceDiagram     participant Admin     participant PembayaranView     participant KonfirmasiPembayaranController     participant KonfirmasiPembayaranModel     Admin-&gt;&gt;PembayaranView: 1 : memilih menu konfirmasi pembayaran     PembayaranView-&gt;&gt;KonfirmasiPembayaranController: 2 : index()     KonfirmasiPembayaranController-&gt;&gt;Admin: 3 : menampilkan halaman konfirmasi pembayaran     Admin-&gt;&gt;PembayaranView: 4 : memilih data pembayaran customer     PembayaranView-&gt;&gt;KonfirmasiPembayaranController: 5 : edit()     KonfirmasiPembayaranController-&gt;&gt;KonfirmasiPembayaranModel: 6 : findOrFail()     KonfirmasiPembayaranModel--&gt;&gt;KonfirmasiPembayaranController: 7 : response konfirmasi pembayaran     KonfirmasiPembayaranController-&gt;&gt;PembayaranView: 8 : menampilkan data per id     Admin-&gt;&gt;PembayaranView: 9 : mengupdate data pembayaran customer     PembayaranView-&gt;&gt;KonfirmasiPembayaranController: 10 : update()     KonfirmasiPembayaranController-&gt;&gt;KonfirmasiPembayaranModel: 11 : update()     KonfirmasiPembayaranModel--&gt;&gt;KonfirmasiPembayaranController: 12 : response konfirmasi pembayaran     KonfirmasiPembayaranController-&gt;&gt;PembayaranView: 13 : menampilkan alert pemberitahuan   </pre>
Gambar 4.22 Diagram Sekuen mengelola konfirmasi pembayaran (update)

## 6. Diagram Sekuen mengelola konfirmasi pembayaran (delete)

Mengelola Konfirmasi Pembayaran
<pre> sequenceDiagram     participant Admin     participant PembayaranView     participant KonfirmasiPembayaranController     participant KonfirmasiPembayaranModel     Admin-&gt;&gt;PembayaranView: 1 : memilih menu konfirmasi pembayaran     PembayaranView-&gt;&gt;KonfirmasiPembayaranController: 2 : index()     KonfirmasiPembayaranController-&gt;&gt;Admin: 3 : menampilkan halaman konfirmasi pembayaran     Admin-&gt;&gt;PembayaranView: 4 : memilih data pembayaran customer     PembayaranView-&gt;&gt;KonfirmasiPembayaranController: 5 : destroy()     KonfirmasiPembayaranController-&gt;&gt;KonfirmasiPembayaranModel: 6 : delete()     KonfirmasiPembayaranModel--&gt;&gt;KonfirmasiPembayaranController: 7 : response konfirmasi pembayaran     KonfirmasiPembayaranController-&gt;&gt;PembayaranView: 8 : menampilkan alert pemberitahuan   </pre>
Gambar 4.23 Diagram Sekuen mengelola konfirmasi pembayaran (delete)

Diagram ini menggambarkan interaksi antara aktor admin dengan sistem yang melakukan proses menghapus konfirmasi pembayaran tiket pada aplikasi berbasis web.

## 7. Diagram Sekuen menampilkan notifikasi



## 8. Diagram Sekuen memverifikasi pemesanan tiket

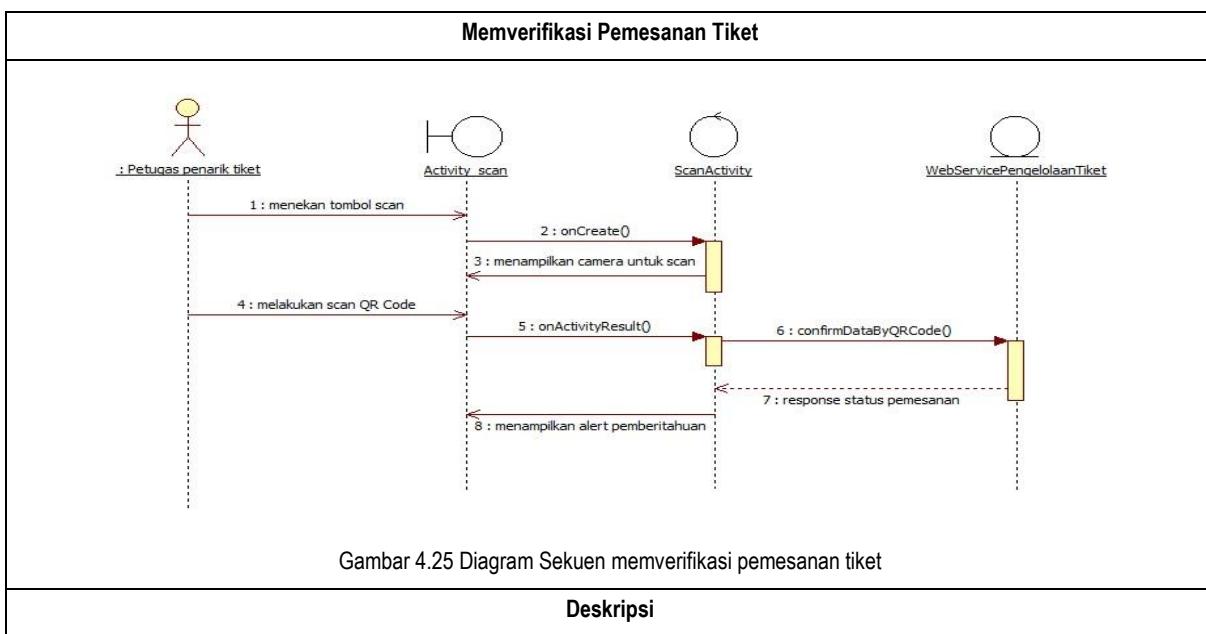
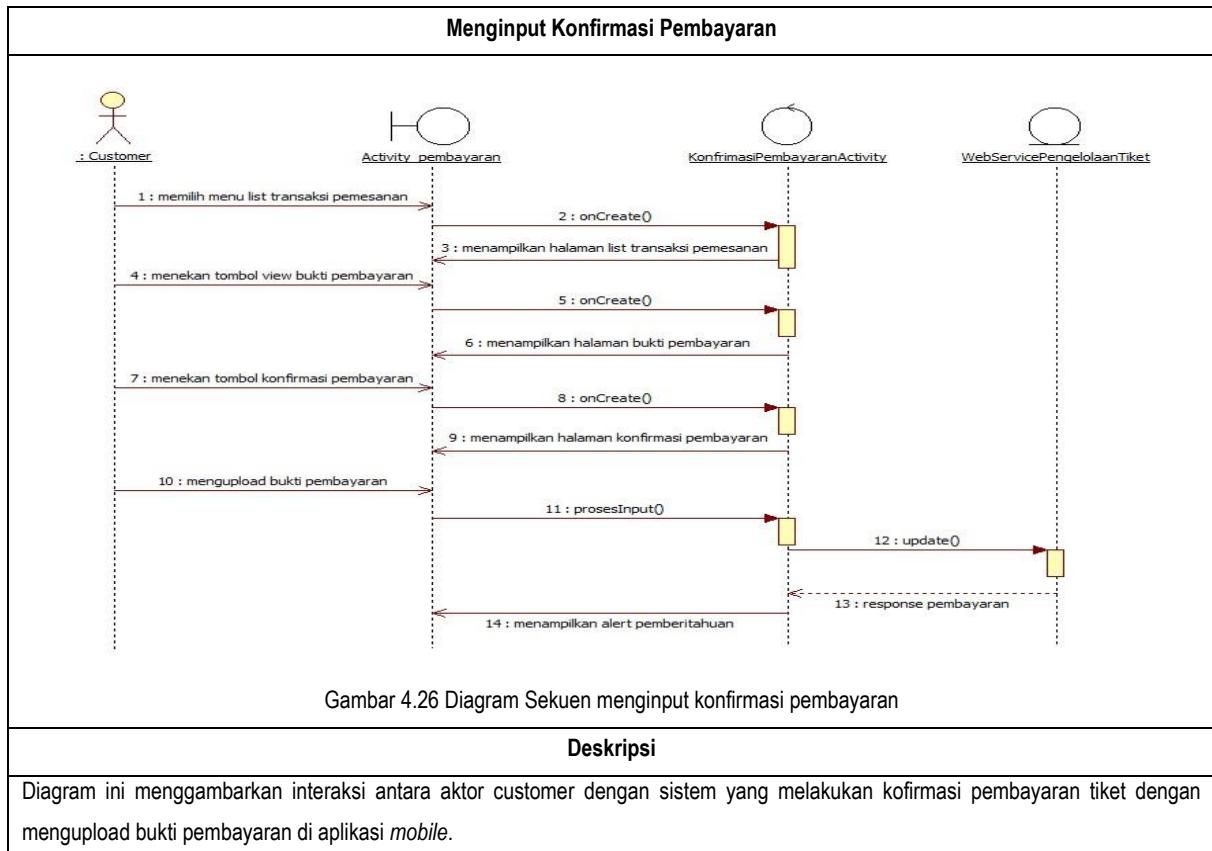


Diagram ini menggambarkan interaksi antara aktor petugas penarik tiket dengan sistem yang melakukan pemindaian QR Code pada tiket di aplikasi mobile.

#### 9. Diagram Sekuen menginput konfirmasi pembayaran



#### 10. Diagram Sekuen mengelola akun karyawan (tambah)

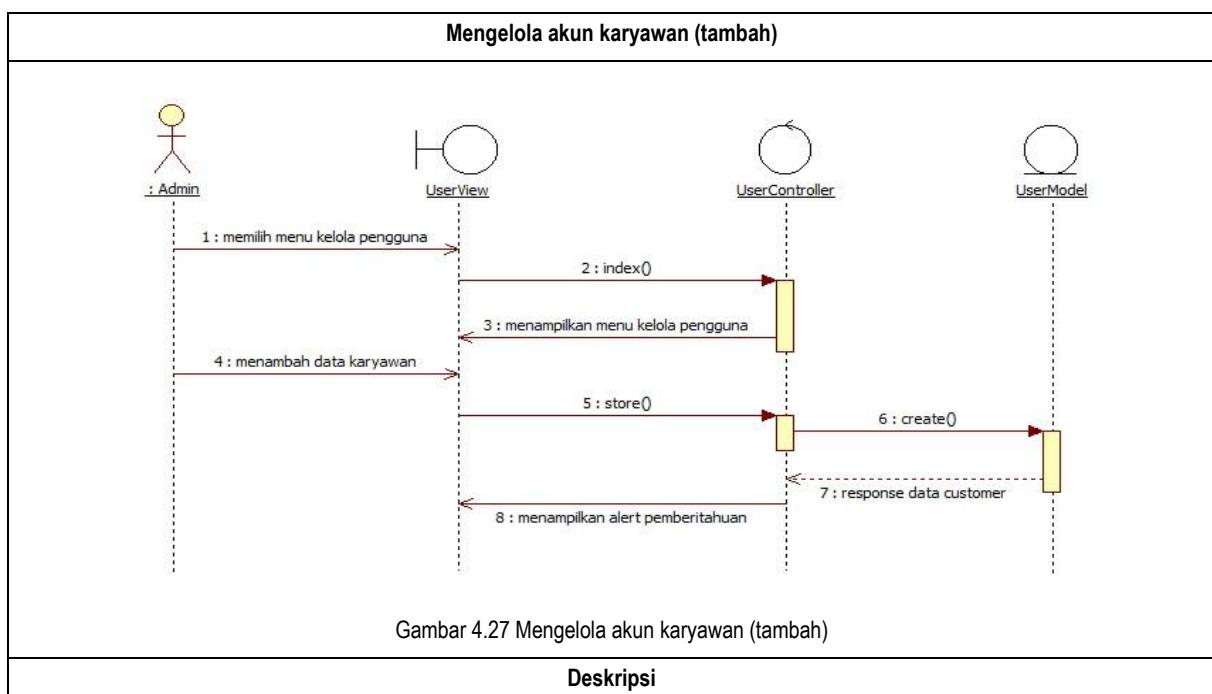
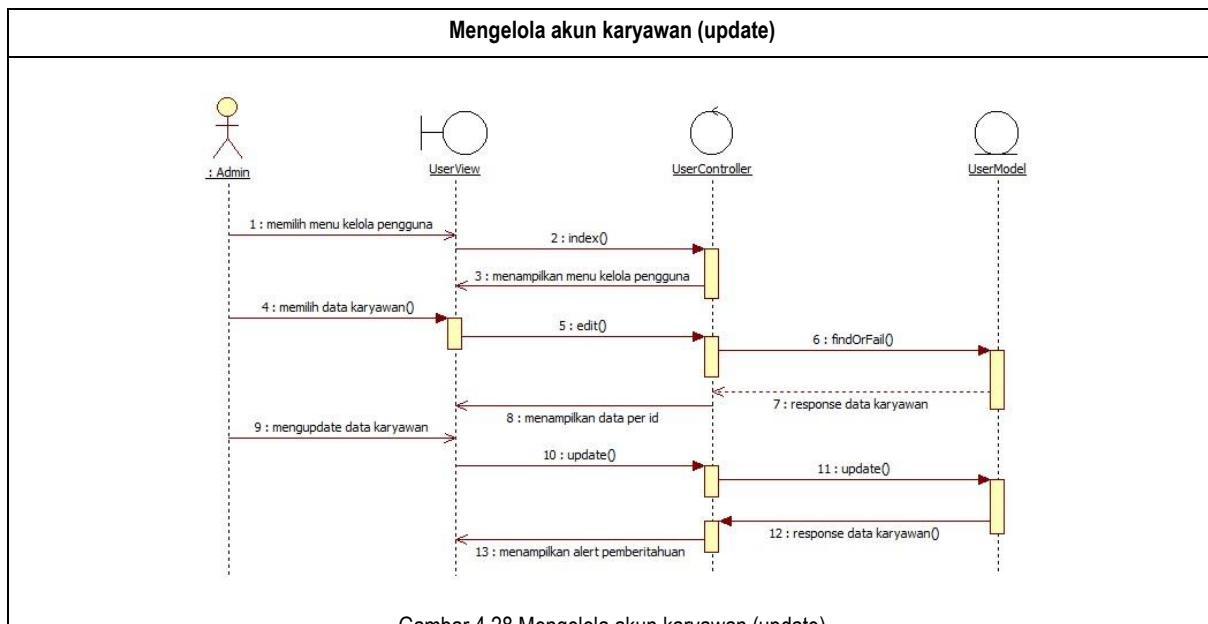


Diagram ini menggambarkan interaksi antara aktor admin dan sistem untuk membuat akun karyawan yang nantinya digunakan karyawan untuk menggunakan aplikasi web.

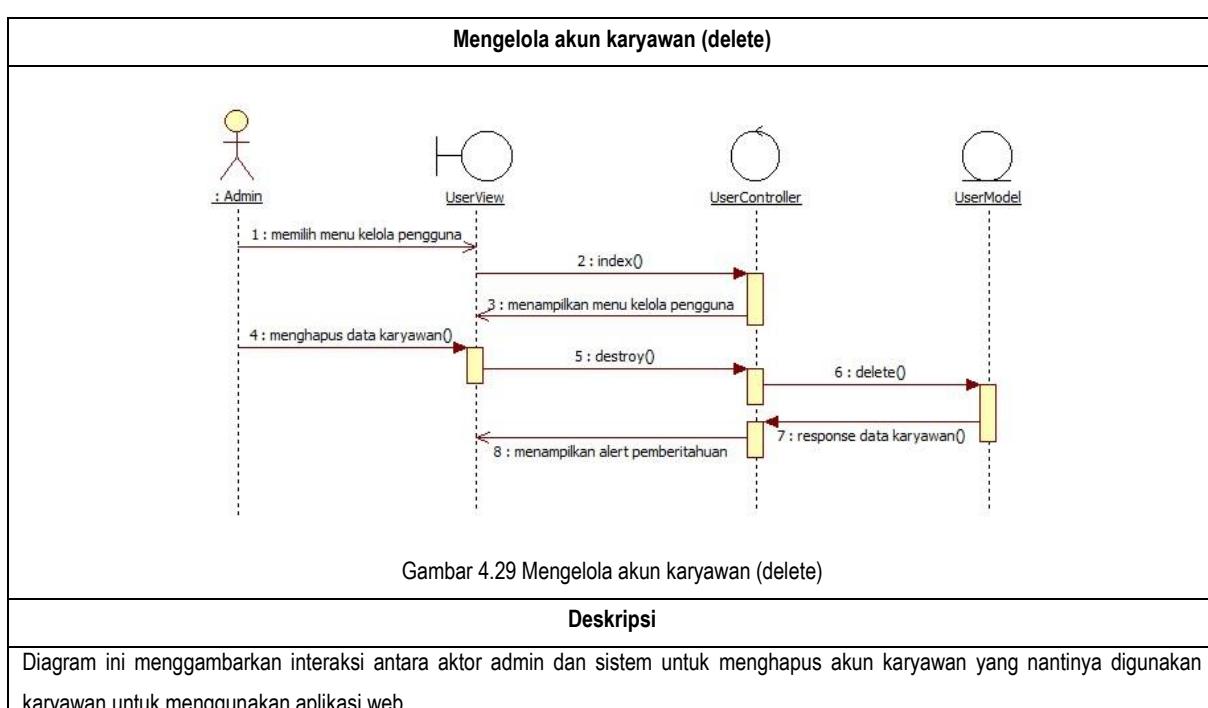
## 11. Diagram Sekuen mengelola akun karyawan (update)



### Deskripsi

Diagram ini menggambarkan interaksi antara aktor admin dan sistem untuk mengupdate akun karyawan yang nantinya digunakan karyawan untuk menggunakan aplikasi web.

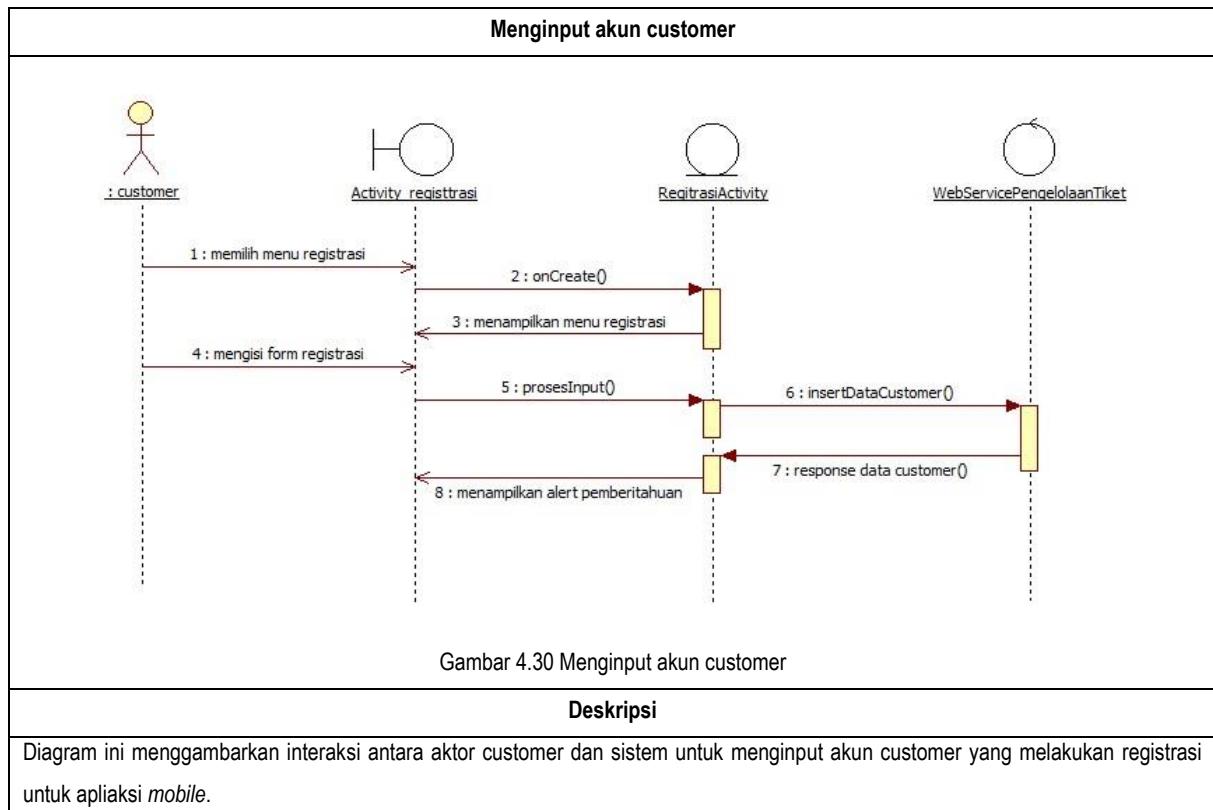
## 12. Diagram Sekuen mengelola akun karyawan (delete)



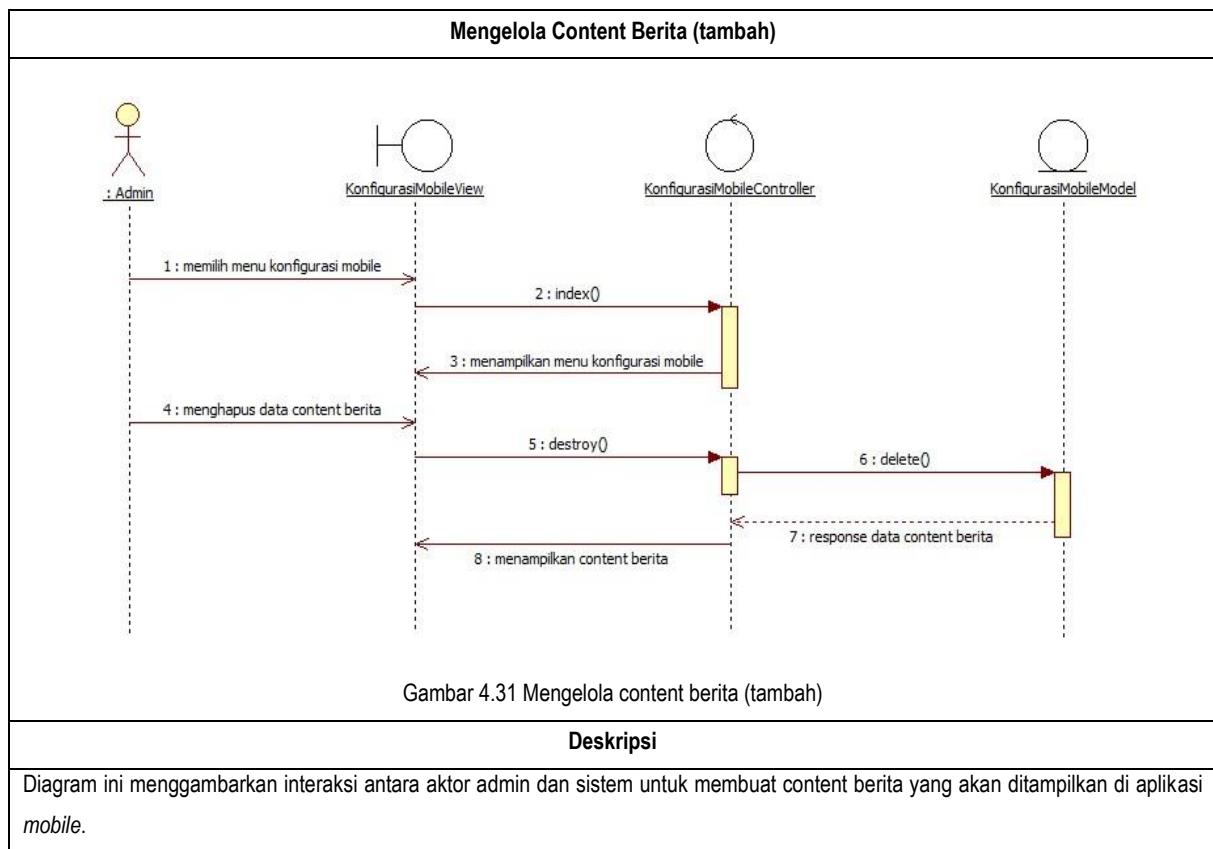
### Deskripsi

Diagram ini menggambarkan interaksi antara aktor admin dan sistem untuk menghapus akun karyawan yang nantinya digunakan karyawan untuk menggunakan aplikasi web.

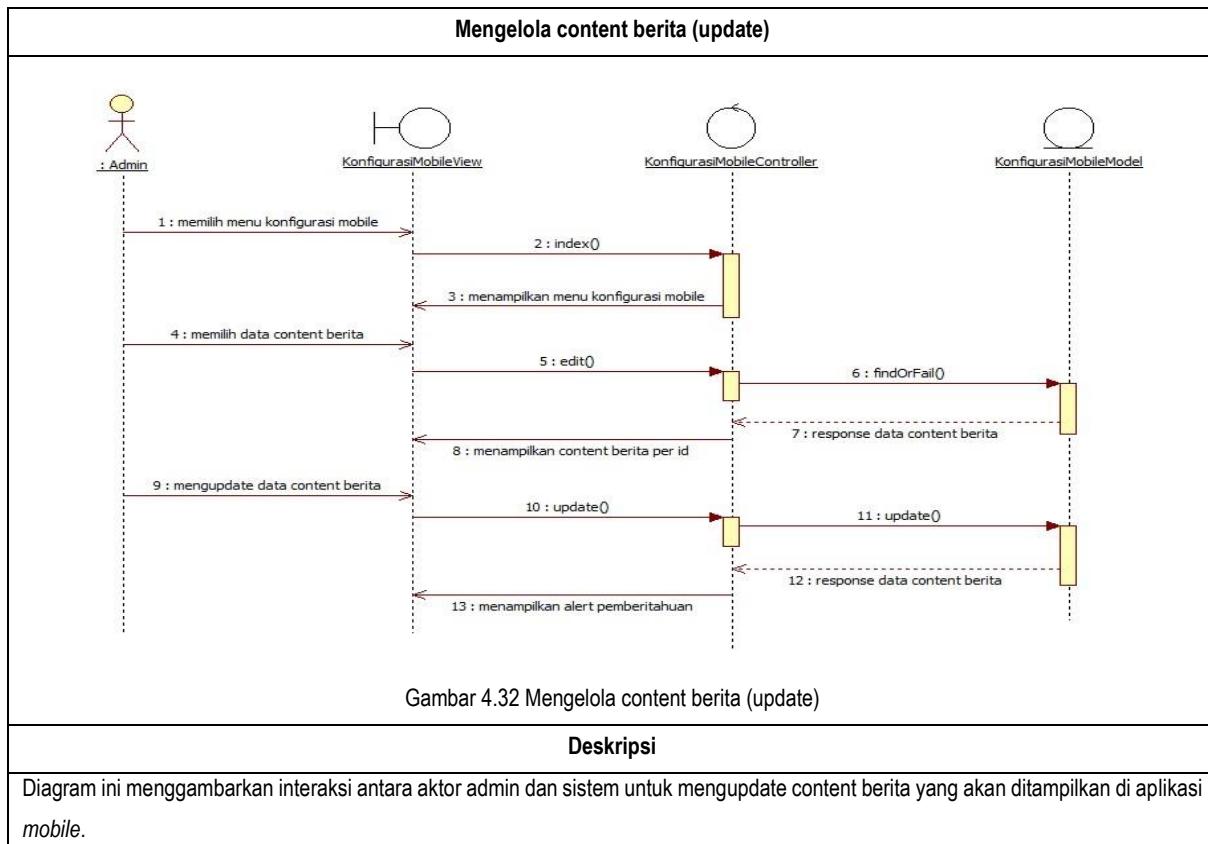
### 13. Diagram Sekuen menginput akun customer



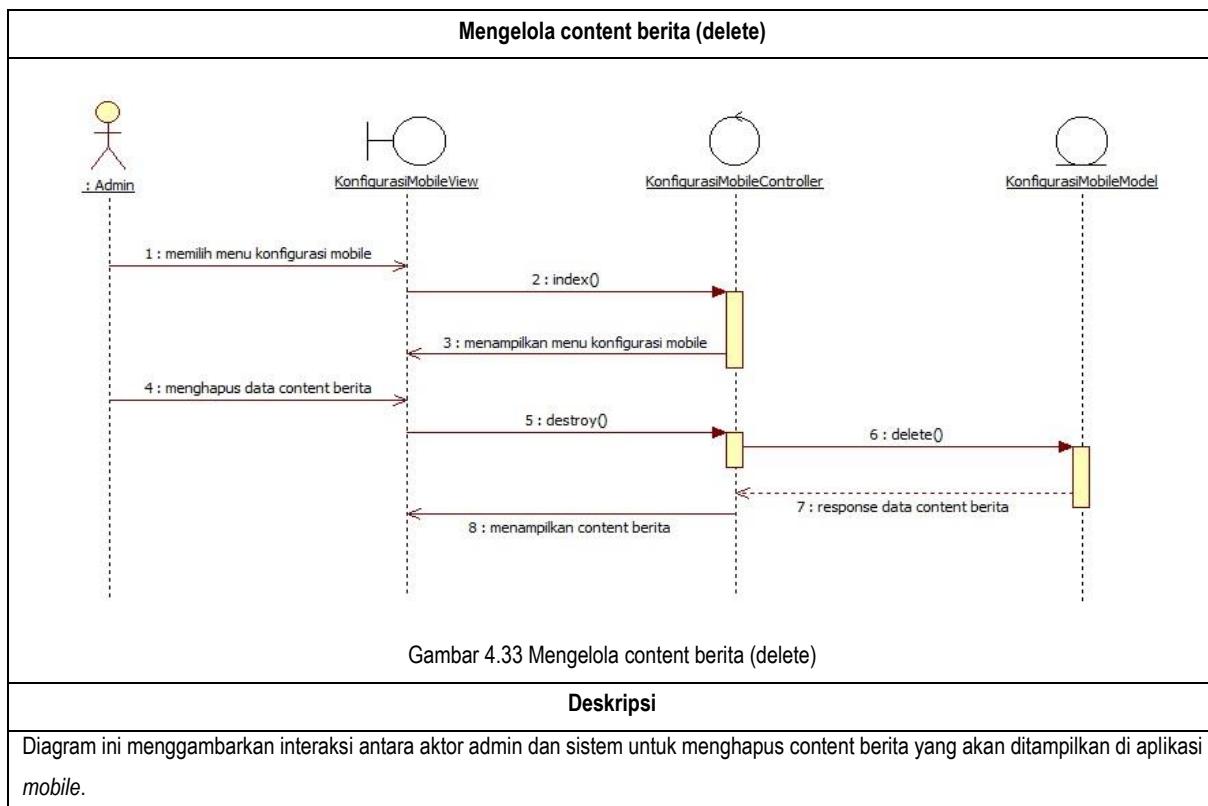
### 14. Diagram Sekuen mengelola content berita (tambah)



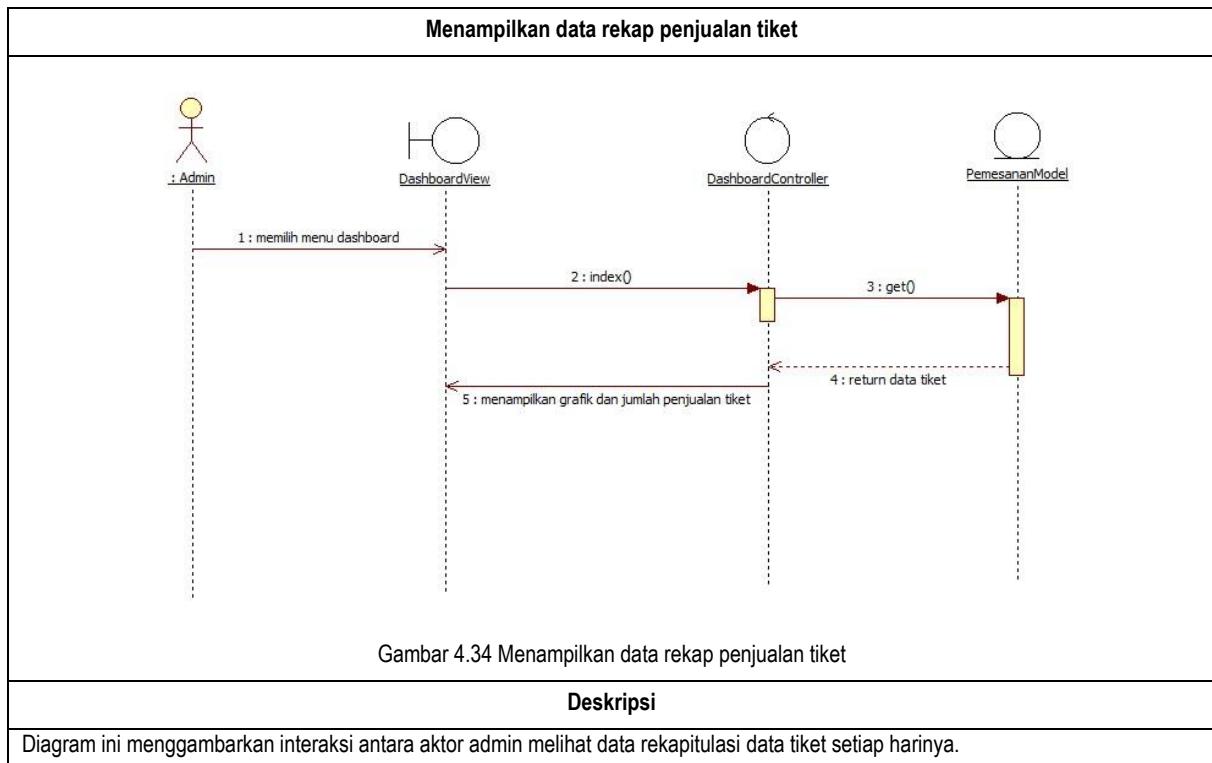
## 15. Diagram Sekuen mengelola content berita (update)



## 16. Diagram Sekuen mengelola content berita (delete)



17. Diagram Sekuen menampilkan data rekap penjualan tiket



### 4.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak pada dasarnya untuk menghasilkan suatu model atau representasi perangkat lunak yang memperlihatkan kekuatan, komoditi, serta kenyamanan. Selain itu, perancangan perangkat lunak merupakan tindakan rekayasa perangkat lunak yang terakhir didalam aktivitas pemodelan dan kemudian merupakan landasan yang sangat penting bagi aktivitas konstruksi [PRE10]. Ada beberapa tahapan dalam perancangan perangkat lunak yaitu perancangan data, perancangan arsitektural, dan perancangan antarmuka.

#### 4.3.1 Perancangan Data

Perancangan data merupakan tahap mendetailkan dari tahap sebelumnya yaitu pemodelan data. Yang sudah mendefinisikan objek-objek data, atribut, dan relasi antar objek serta membuat *entity relationship diagram*. Berikut ini adalah perancangan data yang akan dibuat, dapat dilihat pada Table 4.23.

Table 4.23 Perancangan Data

No.	Nama Objek	Atribut	Tipe Data	Length
1.	Role Level	id_user_level	int	11
		id_menu	int	11
		create	int	11
		read	int	11
		update	int	11
		delete	int	11
		id	Int	11

No.	Nama Objek	Atribut	Tipe Data	Length
2.	Jenis Pemesanan	id	int	11
		nama_jenis	varchar	20
3.	Konfirmasi Pembayaran	id	int	11
		id_status	int	11
		kode_pemesanan	varchar	20
		bukti_pembayaran	text	-
		batas_pembayaran	datetime	-
		id_karyawan	int	11
4.	Menu	id	int	11
		title	varchar	50
		url	varchar	30
		icon	varchar	30
		is_main_menu	int	11
		is_aktif	int	11
		order_num	int	11
5.	Pemesanan Tiket	id	Int	11
		kode_pemesanan	varchar	20
		tgl_pemesanan	date	-
		tgl_masuk	date	-
		nama_kasir	int	11
		jumlah_tiket	int	11
		total_uang_masuk	int	11
		uang_pembayaran	int	11
		status_penggunaan	tinyInt	4
		id_jenis	int	11
		id_customer	int	11
		qr_code	varchar	100
6.	Status Pembayaran	id	int	11
		nama_status	varchar	10
7.	User Customer	id	int	11
		nama	varchar	100
		email	varchar	100
		password	varchar	255
		no_telepon	char	12
		images	text	-
		api_token	varchar	255
8.	Karyawan	id	int	11
		nama	varchar	50
		email	varchar	50
		no_hp	char	12
9.	Role User Karyawan	id	Int	11
		id_karyawan	int	11
		id_user_level	int	11
10.	User Karyawan	id	int	11
		id_karyawan	int	11
		username	varchar	20
		password	varchar	255
		images	text	-
		status	enum	2

No.	Nama Objek	Atribut	Tipe Data	Length
		api_token	varchar	255
11.	User Level	id	int	11
		nama_level	varchar	30

### 4.3.2 Perancangan Arsitektural

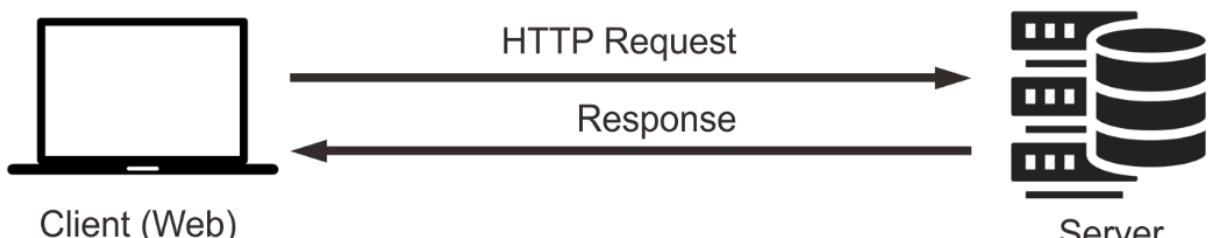
Perancangan arsitektural merepresentasikan struktur data dan komponen-komponen program yang diperlukan untuk mengembangkan sistem-sistem berbasis komputer. Perancangan arsitektural pada dasarnya mempertimbangkan gaya arsitektural yang akan diambil pada sistem dan mempertimbangkan juga hubungan-hubungan yang terjadi di antara semua komponen-komponen yang ada dalam arsitektur sistem [PRE10].

#### 4.3.2.1 Gaya Arsitektural

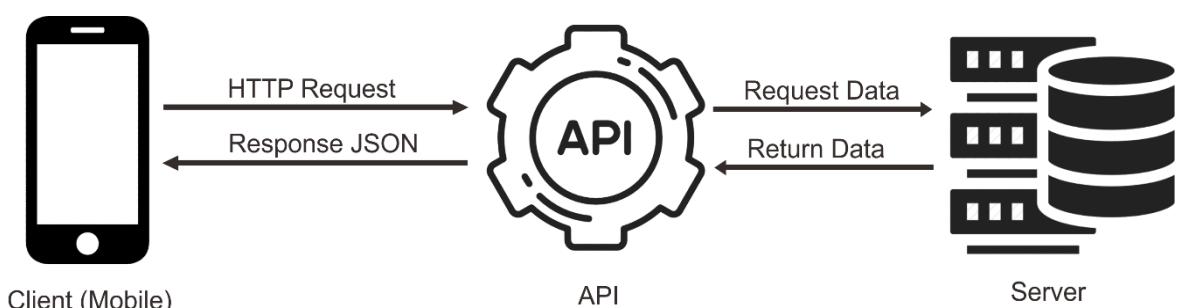
Arsitektural yang digunakan untuk aplikasi berbasis web dan *mobile* yaitu menggunakan gaya arsitektur berpusat pada data. Sebuah tempat penyimpanan data berapa pada pusat arsitektur dan diakses secara sering oleh komponen-komponen lainnya yang melakukan pembaruan, penambahan, penghapusan atau modifikasi data [PRE10]. Berikut ini adalah gaya arsitektur yang digunakan, dapat dilihat pada Gambar 4.35, Gambar 4.36, dan Gambar 4.37.



Gambar 4.35 Gaya Arsitektural Berpusat Pada Data



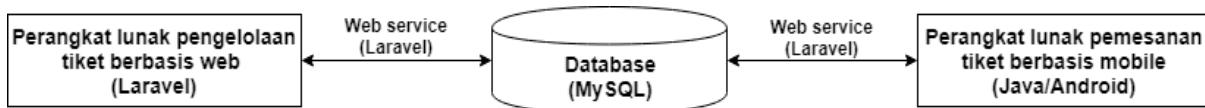
Gambar 4.36 Arsitektur Teknologi Web



Gambar 4.37 Arsitektur Teknologi Mobile

#### 4.3.2.2 Arsitektur Teknologi

Arsitektur teknologi adalah teknologi-teknologi yang digunakan dalam pembuatan aplikasi berbasis web dan aplikasi berbasis *mobile*. Berikut ini adalah teknologi yang digunakan pada pembuatan kedua aplikasi tersebut. Dapat dilihat pada Gambar 4.38.



Gambar 4.38 Arsitektur Teknologi

#### 4.3.3 Perancangan Antarmuka Pengguna

Perancangan antarmuka menciptakan media komunikasi yang efektif diantara manusia dan komputer. Mengikuti sejumlah prinsip perancangan antarmuka, perancangan antarmuka pengguna bekerja dengan cara mengidentifikasi objek-objek antarmuka dan aksi-aksi dan kemudian membuat tata letak layar yang membentuk basis untuk pembuatan suatu prototype antarmuka pengguna [PRE10]. Berikut ini adalah antarmuka pengguna dari aplikasi web dan aplikasi *mobile*.

##### 1. Antarmuka pemesanan tiket berbasis web

The screenshot shows a web browser window titled 'Antarmuka pemesanan tiket berbasis web'. The URL is 'http://failda-waterpark.com/pemesanan'. The page has a purple header bar with the 'Failda Waterpark' logo and a search bar. On the left is a sidebar with navigation links like Dashboard, Kelola Menu, Level Pengguna, Kelola Karyawan, Kelola Pengguna, Kelola Customer, Pemesanan Tiket, Konfirmasi Pembayaran, Konfigurasi Website, and Konfigurasi Mobile. The main content area is titled 'Halaman Pemesanan' and shows a table of 'Daftar Pemesanan Tiket' with columns: No, Kode, Batas Masuk, Jumlah Tiket, Total Pembayaran, Status, Jenis, and Aksi. The table contains 5 entries. At the bottom, it says 'Showing 1 to 6 of 6 entries' and has a 'search' button. The footer includes '2019 © Failda Waterpark Powered By Laravel 5.7' and 'Version 1.0'.

Gambar 4.39 Antarmuka pemesanan tiket berbasis web

##### Deskripsi

Gambar diatas merupakan tampilan untuk halaman pemesanan tiket untuk tampil data berdasarkan hari. Data yang ditampilkan pada tabel yaitu kode pemesanan, batas masuk, jumlah tiket, total pembayaran, status pemesanan dan jenis pemesanan.

2. Antarmuka tambah dan update data pemesanan tiket

Gambar 4.40 Antarmuka tambah dan update data pemesanan tiket berbasis web

**Deskripsi**

Gambar diatas merupakan tampilan untuk halaman tambah pemesanan tiket. Halaman ini bisa dijadikan untuk tambah data pemesanan tiket dan update data pemesanan tiket. Selain itu, untuk mempercepat dalam proses pemesanan disediakan empat buah button yang menunjukan seringnya pengunjung yang memesan tiket mulai dari satu tiket sampai empat tiket. Apabila lebih dari empat maka kasir harus menginput manual.

3. Antarmuka konfirmasi pembayaran berbasis web

Gambar 4.41 Antarmuka konfirmasi pembayaran berbasis web

#### Deskripsi

Gambar diatas merupakan tampilan untuk halaman konfirmasi pembayaran. Data yang ditampilkan pada tabel yaitu kode pemesanan, bukti pembayaran, batas pembayaran, pembayaran, nama customer, dan status pembayaran.

#### 4. Antarmuka update status pembayaran tiket berbasis web

##### Antarmuka update status pembayaran tiket berbasis web

The screenshot shows a web-based application for managing ticket payments. On the left, there's a sidebar with a search bar and a navigation menu including 'Dashboard', 'Kelola Menu', 'Level Pengguna', 'Kelola Karyawan', 'Kelola Pengguna', 'Kelola Customer', 'Pemesanan Tiket', 'Konfirmasi Pembayaran', 'Konfigurasi Website', and 'Konfigurasi Mobile'. The main content area has a title 'Update Data Pembayaran' and a sub-section 'Jumlah Tiket' with a dropdown menu set to 'Menunggu Pembayaran'. Below this is a table with the following data:

No	Bukti	Kode	Tanggal	Tanggal Masuk	Nama Customer	Status	Aksi
3	TRS-23-0003	2019-02-23		Rp. 52.000	Belum diverifikasi	Langsung	
4	TRS-23-0004	2019-02-23	1	Rp. 13.000	Sudah diverifikasi	Langsung	
5	TRS-23-0005	2019-02-23	1	Rp. 24.000	Sudah diverifikasi	Langsung	

At the bottom, there are buttons for 'Cancel' and 'Submit', and a footer note 'Showing 1 to 6 of 6 entries'.

Gambar 4.42 Antarmuka update status pembayaran tiket berbasis web

#### Deskripsi

Gambar diatas merupakan tampilan untuk halaman update status pembayaran pemesanan tiket dari customer. Untuk halaman ini yang berhak mengupdate yaitu admin saja.

#### 5. Antarmuka menampilkan notifikasi pembayaran tiket berbasis web

##### Antarmuka menampilkan notifikasi pembayaran tiket berbasis web

The screenshot shows a web-based application for managing payment notifications. On the left, there's a sidebar with a search bar and a navigation menu including 'Dashboard', 'Kelola Menu', 'Level Pengguna', 'Kelola Karyawan', 'Kelola Pengguna', 'Kelola Customer', 'Pemesanan Tiket', 'Konfirmasi Pembayaran', 'Konfigurasi Website', and 'Konfigurasi Mobile'. The main content area has a title 'Halaman Konfirmasi Pembayaran' and a sub-section 'Daftar Konfirmasi Pembayaran' with a 'Tambah' button. Below this is a table with the following data:

No	Notifikasi Masuk	Nama Customer	Tanggal	Aksi
1		TR5-23-0001	2019-02-23	
2		TR5-23-0002	2019-02-23	
3				
4				
5				

At the bottom, there are buttons for 'Tambah' and 'Refresh', and a footer note 'Showing 1 to 6 of 6 entries'.

Gambar 4.43 Antarmuka notifikasi pembayaran tiket berbasis web

#### Deskripsi

Gambar diatas merupakan tampilan untuk halaman notifikasi pemesanan yang masuk dari customer yang melakukan pemesanan melalui aplikasi mobile.

## 6. Antarmuka halaman kelola pengguna berbasis web

**Antarmuka kelola pengguna berbasis web**

A Web Page  
http://failda-waterpark.com/pemesanan

**Failda Waterpark**

Search...  
Navigation  
Dashboard, Kelola Menu, Level Pengguna, Kelola Karyawan, Kelola Pengguna, Kelola Customer, Pemesanan Tiket, Konfirmasi Pembayaran, Konfigurasi Website, Konfigurasi Mobile

**Halaman Kelola**  
Halaman pengguna digunakan untuk mengelola data user

**Tambah**

**Daftar Pemesanan Tiket**

Show 10 entries

No	Nama Lengkap	username	email	Level	Aksi
1	Faisal	faisal27	faisal@gmail.com	Administrator	[Edit, Delete]
2	Faisal	faisal271	faisal2@gmail.com	Kasir	[Edit, Delete]
3	Faisal	faisal272	faisal3@gmail.com	Administrator	[Edit, Delete]
4	Faisal	faisal273	faisal4@gmail.com	Piminanan	[Edit, Delete]
5	Faisal	faisal274	faisal5@gmail.com	Administrator	[Edit, Delete]

Showing 1 to 6 of 6 entries

2019 © Failda Waterpark Powered By Laravel 5.7 Version 1.0

Gambar 4.44 Antarmuka kelola pengguna berbasis web

### Deskripsi

Gambar diatas merupakan tampilan untuk halaman kelola pengguna. Data yang ditampilkan pada tabel yaitu nama lengkap, username, email, level, dan status.

## 7. Antarmuka tambah dan update kelola pengguna berbasis web

**Antarmuka tambah dan update kelola pengguna berbasis web**

A Web Page  
http://failda-waterpark.com/pemesanan

**Failda Waterpark**

Search...  
Navigation  
Dashboard, Kelola Menu, Level Pengguna, Kelola Karyawan, Kelola Pengguna, Kelola Customer, Pemesanan Tiket, Konfirmasi Pembayaran, Konfigurasi Website, Konfigurasi Mobile

**Tambah/Update Data User**

Nama  
- Pilih Karyawan -

Username  
Masukan username

Email  
Masukan email

password  
Masukan Password

Level  
Pilih level

Status  
Pilih Status

Cancel, Submit

2019 © Failda Waterpark Powered By Laravel 5.7 Version 1.0

Gambar 4.45 Antarmuka tambah dan update kelola pengguna berbasis web

### Deskripsi

Gambar diatas merupakan tampilan untuk halaman update data karyawan. Untuk halaman ini yang berhak mengupdate yaitu admin saja.

## 8. Antarmuka halaman cetak tiket

**Antarmuka cetak tiket berbasis web**

The screenshot shows a web page titled "Antarmuka cetak tiket berbasis web". The URL in the address bar is "http://failda-waterpark.com". The main content is a ticket for "Failda Waterpark" located at "Kp. Rawasari RT/RW 02/03 Kecamatan Sukanegara Desa Sukanegara Kabupaten Cianjur, Jawa Barat, 42364". The ticket details a purchase of "TRIS-23-0001" on "2019-02" for "13.000" with "1 buah". It includes a breakdown: "Total : Rp. 13.000", "Bayar : Rp. 20.000", and "Kembalian : Rp. 7.000". Below the details are four QR codes arranged in a 2x2 grid. At the bottom, there is a message: "Terima kasih telah berkunjung ke kolam renang kami. Telepon : 085624420395 Email : failda.waterpark@gmail.com".

Gambar 4.46 Antarmuka halaman cetak tiket berbasis web

Deskripsi
Gambar diatas merupakan tampilan untuk halaman mencetak tiket yang dipesan oleh customer dengan datang langsung ke tempat.

## 9. Antarmuka halaman scan QR Code mobile

**Antarmuka halaman scan QR Code mobile**

The image shows two screenshots of a mobile application. Both screens have a header showing "07:06 PM" and a battery icon. The left screen has a title "Failda Waterpark" and two buttons: "Scan QR Code" and "Kode Input". It features a large camera icon and a green "Scan" button at the bottom. The right screen also has a title "Failda Waterpark" and two buttons: "Scan QR Code" and "Kode Input". It has a text input field labeled "Kode Pemesanan" and a green "Submit" button below it.

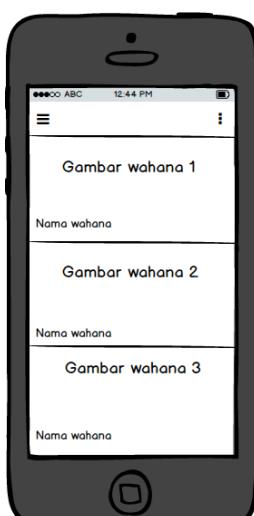
Gambar 4.47 Antarmuka halaman scan QR Code berbasis mobile

Deskripsi
Gambar diatas merupakan tampilan untuk halaman scan QR Code yang dilakukan oleh petugas penarik tiket yang melakukan pemindaian QR Code dari tiket yang telah dipesan customer.

10. Antarmuka halaman registrasi aplikasi *mobile*

<b>Antarmuka halaman registrasi aplikasi <i>mobile</i></b>	
	
Gambar 4.48 Antarmuka halaman registrasi aplikasi <i>mobile</i>	
<b>Deskripsi</b>	
<p>Gambar diatas merupakan tampilan untuk halaman registrasi yang dilakukan oleh customer sebelum melakukan pemesanan tiket di aplikasi <i>mobile</i>.</p>	

11. Antarmuka halaman awal aplikasi *mobile*

<b>Antarmuka halaman awal aplikasi <i>mobile</i></b>	
	
Gambar 4.49 Antarmuka halaman awal aplikasi berbasis mobile	
<b>Deskripsi</b>	
<p>Gambar diatas merupakan tampilan untuk halama awal ketika sudah login di aplikasi <i>mobile</i>.</p>	

12. Antarmuka sidebar aplikasi *mobile*.



Gambar 4.50 Antarmuka sidebar aplikasi berbasis mobile

**Deskripsi**

Gambar diatas merupakan tampilan untuk sidebar yang berisi menu-menu yang bisa digunakan oleh customer baik untuk memesan tiket, mengonfirmasi pembayaran, dan mengubah profil.

13. Antarmuka halaman pemesanan tiket aplikasi *mobile*.



Gambar 4.51 Antarmuka halaman pemesanan tiket aplikasi mobile

**Deskripsi**

Gambar diatas merupakan tampilan untuk halaman pemesanan tiket. Customer bisa harus mengisi tanggal masuk dan jumlah tiket yang akan dipesan, setelah itu akan dialihkan ke halaman detail pemesanan tiket untuk memastikan tiket yang dipesan sudah sesuai atau belum. Jika belum bisa merubahnya terlebih dahulu.

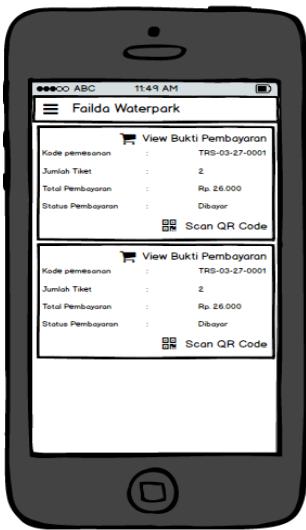
14. Antarmuka halaman konfirmasi pembayaran aplikasi *mobile*

<b>Antarmuka halaman konfirmasi pembayaran aplikasi <i>mobile</i></b>	
	
Gambar 4.52 Antarmuka halaman konfirmasi pembayaran aplikasi <i>mobile</i>	
<b>Deskripsi</b>	
<p>Gambar diatas merupakan tampilan untuk halaman konfirmasi pembayaran apabila customer sudah melakukan pemesanan tiket di aplikasi <i>mobile</i>.</p>	

15. Antarmuka halaman bukti pembayaran

<b>Antarmuka halaman bukti pembayaran</b>	
	
Gambar 4.53 Antarmuka halaman bukti pembayaran	
<b>Deskripsi</b>	
<p>Gambar diatas merupakan tampilan untuk halaman bukti pembayaran ketika customer sudah melakukan pemesanan.</p>	

16. Antarmuka halaman list pemesanan tiket

Antarmuka halaman bukti pembayaran											
 <p>The image shows a smartphone displaying two identical payment receipt screens side-by-side. The screen is titled 'Failda Waterpark' and shows the following details:</p> <table border="1"><thead><tr><th colspan="2">View Bukti Pembayaran</th></tr></thead><tbody><tr><td>Kode pemesanan</td><td>: TRS-03-27-0001</td></tr><tr><td>Jumlah Tiket</td><td>: 2</td></tr><tr><td>Total Pembayaran</td><td>: Rp. 26.000</td></tr><tr><td>Status Pembayaran</td><td>: Dibayar</td></tr></tbody></table> <p>Below each receipt is a 'Scan QR Code' button.</p>		View Bukti Pembayaran		Kode pemesanan	: TRS-03-27-0001	Jumlah Tiket	: 2	Total Pembayaran	: Rp. 26.000	Status Pembayaran	: Dibayar
View Bukti Pembayaran											
Kode pemesanan	: TRS-03-27-0001										
Jumlah Tiket	: 2										
Total Pembayaran	: Rp. 26.000										
Status Pembayaran	: Dibayar										
<p>Gambar 4.54 Antarmuka halaman list pemesanan tiket</p>											
<p><b>Deskripsi</b></p>											
<p>Gambar diatas merupakan tampilan untuk halaman list pemesanan tiket ketika customer sudah melakukan pemesanan.</p>											

## **BAB 5**

### **KONSTRUKSI DAN DEPLOYMENT**

Bab ini menjelaskan proses pembangunan perangkat lunak, testing perangkat lunak dan deployment perangkat lunak. Yang terdiri dari tahapan konstruksi perangkat lunak dan deployment perangkat lunak.

#### **5.1 Konstruksi Perangkat Lunak**

Konstruksi perangkat lunak mengacu pada pembuatan detail kerja, perangkat lunak yang berarti melalui kombinasi koding, verifikasi, pengujian unit, integrasi, dan *debugging*. Konstruksi perangkat lunak mengacu pada perancangan-perancangan yang sudah dibuat sebelumnya.

##### **5.1.1 Kebutuhan Perangkat dan Kakas Untuk Pengembangan**

Berikut ini adalah kebutuhan perangkat dan kakas untuk pengembangan perangkat lunak yang akan dibangun :

- a. Spesifikasi Laptop

Table 5.1 Spesifikasi Laptop

No.	Nama Perangkat	System Requirement
1.	Processor	Intel Core i5-6200U CPU @ 2.30 GHz (4 CPUs)
2.	RAM (Random Access Memory)	8 GB DDR3L
3.	Harddisk	SSD 512 GB Samsung
4.	Operating System	Windows 10 Pro 64 bit
5.	GPU (Graphics Processing Unit)	AMD Radeon(TM) R5 M330 (PS5.0/VS5.0)

- b. Software yang digunakan dalam pengembangan

Table 5.2 Software yang digunakan dalam pengembangan

No.	Nama Perangkat	Sumber	Deskripsi
1.	PhpStorm	<a href="https://www.jetbrains.com/phpstorm/">https://www.jetbrains.com/phpstorm/</a>	Software yang digunakan untuk web developer dalam membangun aplikasi berbasis web khususnya bahasa pemrograman Php.
2.	ERDPlus	<a href="https://erdplus.com/#/">https://erdplus.com/#/</a>	Software berbasis web yang digunakan untuk membuat Entity Relationship Diagram (ERD)
3.	StarUML	<a href="http://staruml.io/">http://staruml.io/</a>	Software yang digunakan untuk membuat pemodelan yang mendukung UML (Unified Modeling Language)
4.	Laragon	<a href="https://laragon.org/">https://laragon.org/</a>	Software yang digunakan untuk web server Apache yang didalamnya sudah tersedia database server MYSQL untuk mendukung PHP programming.
5.	Balsamiq Mockup	<a href="https://balsamiq.com/">https://balsamiq.com/</a>	Software yang digunakan untuk mendesain tampilan prototype dari perangkat lunak yang akan dibangun
6.	Android Studio	<a href="https://developer.android.com/studio">https://developer.android.com/studio</a>	Software yang digunakan untuk membangun aplikasi mobile berbasis android.

No.	Nama Perangkat	Sumber	Deskripsi
7.	GIT	<a href="https://git-scm.com/">https://git-scm.com/</a>	Software yang digunakan untuk version control system.
8.	Postman	<a href="https://www.getpostman.com/">https://www.getpostman.com/</a>	Software yang digunakan untuk testing API.

### 5.1.2 Pengkodean

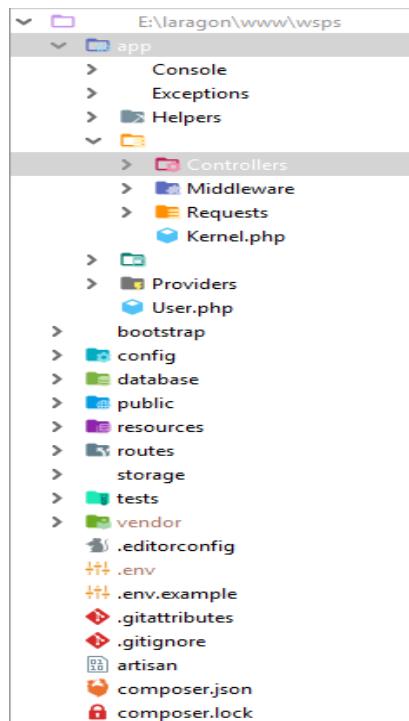
Tahap pengkodean adalah tahap pengimplementasian dari hasil model-model desain ditahap perancangan menjadi kode-kode program. Pada tahap pengkodean ini ada tiga tahap yang akan dibangun yaitu tahap pengkodean aplikasi *web*, *web service*, dan aplikasi *mobile*.

#### 5.1.2.1 Tahap Pengkodean Aplikasi Web dan Web Service

Tahap pengkodean aplikasi *web* dan *web service* yang akan dibangun menggunakan bahasa pemrograman PHP dengan framework Laravel serta database MYSQL.

##### 1. Struktur Aplikasi Web dan Web Service

Struktur yang dibangun menggunakan pola arsitektur MVC untuk memisahkan antara data (model) dengan user interface (view) serta logika (controller). Untuk di *framework* Laravel bagian model dan controller ada di folder app serta views berada di folder resource. Berikut ini struktur direktori aplikasi web dapat dilihat pada Gambar 5.1:



Gambar 5.1 Struktur Directori Aplikasi *Web* dan *Web Service*

Dibutuhkan sebuah file inti yang berguna untuk mengatur alur kerja dari aplikasi *web* dan *web service*, untuk melihat kode program dari masing-masing file dapat dilihat pada repository yang ada di Gitlab. Untuk link repository terdapat pada lampiran B-1 . Berikut ini file-file yang ada pada controllers, models, dan views di aplikasi *web* dan *web service*.

## 2. Controller Aplikasi Web

Controllers pada aplikasi *web* berguna untuk melakukan pemrosesan data yang diminta oleh bagian client yaitu pada bagian views yang akan mengakses data ke *database*. Berikut ini adalah beberapa controllers pada aplikasi web dapat dilihat pada Table 5.3 :

Table 5.3 Daftar Controller Aplikasi Web

No	Nama Controller	Deskripsi
1.	UserController	Untuk mengelola data karyawan yang menggunakan aplikasi <i>web</i> .
2.	KonfirmasiPembayaranController	Untuk mengelola data pembayaran dari customer yang melakukan pemesanan tiket melalui aplikasi <i>mobile</i> untuk mengecek apakah customer sudah melakukan pembayaran atau belum.
3.	PemesananController	Untuk mengelola data pemesanan tiket yang dipesan oleh customer baik menggunakan aplikasi <i>web</i> atau aplikasi <i>mobile</i> .
4.	KonfigurasiMobileController	Untuk mengelola data konten berita yang akan ditampilkan di aplikasi <i>mobile</i> .

### a. Controller UserController

Controller UserController berfungsi untuk mengelola data karyawan yang menggunakan aplikasi *web*. Dalam UserController ini terdapat fungsi-fungsi yang dapat dilihat pada Gambar 5.2 Controller UserController :

```
<?php
namespace App\Http\Controllers;
use ...;

class UserController extends Controller
{
    public function index() {...}
    public function datatable() {...}
    public function store(UserRequest $request) {...}
    public function edit(Request $request) {...}
    public function update(Request $request) {...}
    public function destroy(Request $request) {...}
    public function resetPassword(Request $request) {...}
    public function cekUsername(Request $request) {...}
    public function cekEmail(Request $request) {...}
    public function cekNoHp(Request $request) {...}
}
```

Gambar 5.2 Controller UserController

Berikut ini adalah deskripsi dari setiap method yang ada pada controller UserController :

1. Fungsi index untuk menampilkan halaman kelola pengguna (user)
2. Fungsi datatable untuk menampilkan data user ke dalam bentuk table dengan menggunakan library datatable

3. Fungsi store untuk menyimpan data user ke dalam *database*.
4. Fungsi edit untuk mengambil data user per id untuk keperluan update data user
5. Fungsi update untuk mengubah data user berdasarkan id user
6. Fungsi destroy untuk menghapus data user dengan cara *softdelete*
7. Fungsi resetPassword untuk mengreset password user
8. Fungsi cekUsername untuk mengecek username sudah tersedia atau belum
9. Fungsi cekEmail untuk mengecek email sudah tersedia atau belum
10. Fungsi cekNoHP untuk mengecek nomor handphone sudah tersedia atau belum

**b. Controller KonfirmasiPembayaranController**

Controller KonfirmasiPembayaranController berfungsi untuk mengelola data pembayaran dari customer yang melakukan pemesanan tiket melalui aplikasi *mobile* untuk mengecek apakah customer sudah melakukan pembayaran atau belum. Dalam KonfirmasiPembayaranController ini terdapat fungsi-fungsi yang dapat dilihat pada Gambar 5.3 :

```

<?php

namespace App\Http\Controllers;

use ...

class KonfirmasiPembayaranController extends Controller
{
    public function index() {...}

    public function datatable(Request $request) {...}

    public function getKodePemesanan(Request $request) {...}

    public function edit(Request $request) {...}

    public function getBuktiPembayaran(Request $request) {...}

    public function update(KonfirmasiPembayaranRequest $request) {...}

    public function destroy(Request $request) {...}
}

```

Gambar 5.3 Controller KonfirmasiPembayaranController

Berikut ini adalah deskripsi dari setiap method yang ada pada controller KonfirmasiPembayaranController :

1. Fungsi index untuk menampilkan halaman konfirmasi pembayaran
2. Fungsi datatable untuk menampilkan data pembayaran ke dalam bentuk table dengan menggunakan library datatable
3. Fungsi getKodePemesanan untuk mencari pembayaran dari customer berdasarkan kode pemesanan yang sudah dipesan
4. Fungsi edit untuk mengambil data pembayaran per id untuk keperluan update data pembayaran
5. Fungsi getBuktiPembayaran untuk mengambil bukti pembayaran dari customer
6. Fungsi update untuk mengubah data pembayaran berdasarkan id pembayaran

7. Fungsi destroy untuk menghapus data pembayaran dengan cara *softdelete*.

#### c. Controller PemesananController

Controller PemesananController berfungsi untuk mengelola data pemesanan tiket yang dipesan oleh customer baik menggunakan aplikasi *web* atau aplikasi *mobile*. Dalam PemesananController ini terdapat fungsi-fungsi yang dapat dilihat pada Gambar 5.4 :

```

<?php

namespace App\Http\Controllers;

use ...

class PemesananController extends Controller
{
    public function index() {...}

    public function datatable(Request $request) {...}

    public function store(PemesananRequest $request) {...}

    public function show($id) {...}

    public function edit(Request $request) {...}

    public function update(Request $request) {...}

    public function destroy(Request $request) {...}

    public function printTicket($id) {...}
}

```

Gambar 5.4 Controller PemesananController

Berikut ini adalah deskripsi dari setiap method yang ada pada controller PemesananController :

1. Fungsi index untuk menampilkan halaman pemesanan tiket
2. Fungsi datatable untuk menampilkan data pemesanan ke dalam bentuk table dengan menggunakan library datatable
3. Fungsi store untuk menyimpan data pemesanan ke *database*.
4. Fungsi show untuk melihat detail pemesanan tiket per id
5. Fungsi edit untuk mengambil data pemesanan per id untuk keperluan update data pembayaran
6. Fungsi update untuk mengubah data pemesanan per id
7. Fungsi destroy untuk menghapus data pemesanan dengan cara *softdelete*
8. Fungsi printTicket untuk mencetak tiket

#### d. Controller KonfigurasiMobileController

Controller KonfigurasiMobileController berfungsi untuk mengelola data konten berita yang akan ditampilkan di aplikasi *mobile*. Dalam KonfigurasiMobileController ini terdapat fungsi-fungsi yang dapat dilihat pada Gambar 5.5 :

```

<?php

namespace App\Http\Controllers;

use ...

class KonfigurasiMobileController extends Controller
{
    public function index() {...}

    public function datatable() {...}

    public function store(KonfigurasiMobileRequest $request) {...}

    public function edit(Request $request) {...}

    public function update(Request $request) {...}

    public function destroy(Request $request) {...}
}

```

Gambar 5.5 Controller KonfigurasiMobileController

Berikut ini adalah deskripsi dari setiap method yang ada pada controller KonfigurasiMobileController :

1. Fungsi index untuk menampilkan halaman Konfigurasi *Mobile*
2. Fungsi datatable untuk menampilkan data konfigurasi mobile ke dalam bentuk table dengan menggunakan library datatable
3. Fungsi store untuk menyimpan data konfigurasi mobile ke *database*
4. Fungsi edit untuk mengambil data konfigurasi mobile per id untuk keperluann update data konfigurasi mobile
5. Fungsi update untuk mengubah data konfigurasi *mobile* per id
6. Fungsi destroy untuk menghapus data konfigurasi *mobile* dengan cara *softdelete*

### 3. Controller Web Service

Controllers pada *web service* berguna untuk melakukan pemrosesan data yang diminta oleh pengguna *web service* untuk nantinya dihubungkan ke database serta menampilkan data kepada pengguna dalam bentuk *json*. Berikut ini adalah beberapa controllers pada *web service* dapat dilihat pada Table 5.4 *Daftar Controller Web Service*:

Table 5.4 Daftar Controller Web Service

No	Nama Controller	Deskripsi
1.	CustomerController	Untuk menerima request dari client yang menggunakan aplikasi <i>mobile</i> untuk mengambil data customer.
2.	PembayaranController	Untuk menerima request dari client yang menggunakan aplikasi <i>mobile</i> untuk mengambil data customer.
3.	PemesananController	Untuk mengelola data pemesanan tiket yang dipesan oleh customer baik menggunakan aplikasi <i>web</i> atau aplikasi <i>mobile</i> .
4.	KonfigurasiMobileController	Untuk mengirim data konten berita yang akan ditampilkan di aplikasi <i>mobile</i> .

### a. Controller CustomerController

Controller CustomerController berfungsi untuk menerima request dari client yang menggunakan aplikasi *mobile* untuk mengambil data customer. Dalam CustomerController ini terdapat fungsi-fungsi yang dapat dilihat pada Gambar 5.6 :

```
<?php
namespace App\Http\Controllers\api;

use ...

class CustomerController extends Controller
{
    public function show(Request $request){...}

    public function update(Request $request){...}
}
```

Gambar 5.6 Controller Web Service PemesananController

Berikut ini adalah deskripsi dari setiap method yang ada pada controller CustomerController :

1. Fungsi show untuk mengambil data customer per id
2. Fungsi update untuk mengubah data customer berdasarkan id

### b. Controller PembayaranController

Controller PembayaranController berfungsi untuk menerima request dari client yang menggunakan aplikasi *mobile* untuk mengupload bukti pembayaran. Dalam PembayaranController ini terdapat fungsi-fungsi yang dapat dilihat pada Gambar 5.7 *Controller Web Service PembayaranController* :

```
<?php
namespace App\Http\Controllers\api;

use ...

class PembayaranController extends Controller
{
    public function update(Request $request){...}
}
```

Gambar 5.7 Controller Web Service PembayaranController

Berikut ini adalah deskripsi dari setiap method yang ada pada controller PembayaranController :

1. Fungsi update untuk mengubah bukti pembayaran yang sebelumnya belum terisi

### c. Controller PemesananController

Controller PemesananController berfungsi untuk menerima request dari client yang menggunakan aplikasi *mobile* untuk melakukan pemesanan tiket dari aplikasi *mobile*. Dalam PemesananController ini terdapat fungsi-fungsi yang dapat dilihat pada Gambar 5.8 :



```
<?php

namespace App\Http\Controllers\api;

use ...

class PemesananController extends Controller
{
    public function show(Request $request) {...}

    public function getPemesananByIdCustomer(Request $request) {...}

    public function store(Request $request) {...}

    public function verifyDataByQRCode(Request $request) {...}
}
```

Gambar 5.8 Controller Web Service PemesananController

Berikut ini adalah deskripsi dari setiap method yang ada pada controller PemesananController :

1. Fungsi show untuk mengambil data pemesanan berdasarkan id
2. Fungsi getPemesananByIdCustomer untuk mengambil data pemesanan berdasarkan id customer
3. Fungsi store untuk menyimpan data pemesanan ke *database*.
4. Fungsi verifyDataByQRCode untuk memverifikasi data pemesanan berdasarkan qr code

#### d. Controller KonfigurasiMobileController

Controller KonfigurasiMobileController berfungsi untuk mengirim data konten berita yang akan ditampilkan di aplikasi *mobile*. Dalam KonfgurasiMobileController ini terdapat fungsi-fungsi diantaranya yang dapat dilihat pada Gambar 5.9 :



```
<?php

namespace App\Http\Controllers\api;

use ...

class KonfigurasiMobileController extends Controller
{
    public function index()
    {
        $kontenMobile = KonfigurasiMobileModel::all();
        $konfigurasiWeb = KonfigurasiModel::all();

        $data = [
            'harga' => (integer) $konfigurasiWeb[2]->nilai_konfig,
            'bank' => $konfigurasiWeb[9]->nilai_konfig,
            'norek' => $konfigurasiWeb[10]->nilai_konfig,
            'pemilik' => $konfigurasiWeb[11]->nilai_konfig,
        ];

        if ($kontenMobile) {
            return response()->json(['result' => $kontenMobile, 'konfigurasi' => $data, 'status' => 200]);
        } else {
            return response()->json(['msg' => "Data tidak ditemukan", 'status' => 500]);
        }
    }
}
```

Gambar 5.9 Controller Web Service KonfigurasiMobileController

Berikut ini adalah deskripsi dari setiap method yang ada pada controller KonfigurasiMobileController :

1. Fungsi index untuk mengirim data content berita ke aplikasi *mobile* untuk ditampilkan di halaman awal

#### 4. Model Aplikasi Web dan Web Service

Model pada aplikasi *web* dan *web service* berguna untuk menghubungkan aplikasi dengan *database* dimana nama table dan field-field pada *database* di definisikan di model. Berikut ini adalah beberapa model yang ada pada *web service* dan aplikasi *web* dapat dilihat pada Table 5.5 :

Table 5.5 Daftar Model Web Service dan Aplikasi Web

No	Nama Model	Deskripsi
1.	UserModel	Untuk mengambil data user dari <i>database</i> untuk digunakan di aplikasi <i>web</i> dan <i>mobile</i> .
2.	KaryawanModel	Untuk mengambil data karyawan dari <i>database</i> untuk digunakan di aplikasi <i>web</i> dan <i>mobile</i> .
3.	KonfirmasiPembayaranModel	Untuk mengambil data pembayaran customer dari <i>database</i> yang akan digunakan di aplikasi <i>web</i> .
4.	PemesananModel	Untuk mengambil data pemesanan customer dari <i>database</i> yang akan digunakan di aplikasi <i>web</i> .
5.	CustomerModel	Untuk mengambil data customer dari <i>database</i> yang akan digunakan di aplikasi <i>web</i> dan <i>mobile</i> .

##### a. Model UserModel

Model UserModel berfungsi untuk mengambil data user dari *database* untuk digunakan di aplikasi *web* dan *mobile*. Dalam UserModel ini terdapat fungsi-fungsi yang dapat dilihat pada Gambar 5.10 :

```
<?php
namespace App\Models;

use ...

class UserModel extends Authenticatable
{
    use Notifiable;
    use SoftDeletes;
    protected $table = "user_karyawan";
    protected $guard = 'api_karyawan';
    protected $primaryKey = "id";
    protected $dates = ['deleted_at', 'updated_at', 'created_at'];
    protected $fillable = ["id_karyawan", "username", "password", "images", "status", "api_token"];

    protected $hidden = [
        'password' , 'api_token'
    ];

    public function pemesanan() {...}

    public function karyawan() {...}

    public function karyawanRole() {...}
}
```

Gambar 5.10 Model UserModel

Berikut ini adalah deskripsi dari setiap method yang ada pada model UserModel :

1. Fungsi pemesanan untuk berelasi dengan kelas pemesanan tiket
2. Fungsi karyawan untuk berelasi dengan kelas karyawan
3. Fungsi karyawanRole untuk berelasi dengan kelas choose role

### b. Model KaryawanModel

Model KaryawanModel berfungsi untuk mengambil data karyawan dari *database* untuk digunakan di aplikasi *web* dan *mobile*. Berikut ini isi dari model KaryawanModel dapat dilihat pada Gambar 5.11 :

```
<?php

namespace App\Models;

use ...

class KaryawanModel extends Model
{
    use SoftDeletes;
    protected $table = "karyawan";
    protected $primaryKey = "id";
    protected $dates = ['deleted_at', 'updated_at', 'created_at'];
    protected $fillable = ['nama', 'email', 'no_hp'];
}
```

Gambar 5.11 Model KaryawanModel

### c. Model KonfirmasiPembayaranModel

Model KonfirmasiPembayaranModel berfungsi untuk mengambil data pembayaran customer dari *database* yang akan digunakan di aplikasi *web*. Dalam KonfirmasiPembayaranModel ini terdapat fungsi-fungsi yang dapat dilihat pada Gambar 5.12 :

```
<?php

namespace App\Models;

use ...

class KonfirmasiPembayaranModel extends Model
{
    use SoftDeletes;
    protected $table = "pembayaran";
    protected $primaryKey = "id";
    protected $dates = ['deleted_at', 'updated_at', 'created_at'];
    protected $fillable = ['kode_pemesanan', 'id_status', 'bukti_pembayaran', 'id_karyawan', "batas_pembayaran",
    "komentar"];

    public function pemesananTiket()
    {
        return $this->hasOne( related: PemesananModel::class, foreignKey: 'kode_pemesanan', localKey: 'kode_pemesanan');
    }

    public function statusPembayaran()
    {
        return $this->hasOne( related: StatusPembayaranModel::class, foreignKey: 'id', localKey: 'id_status');
    }
}
```

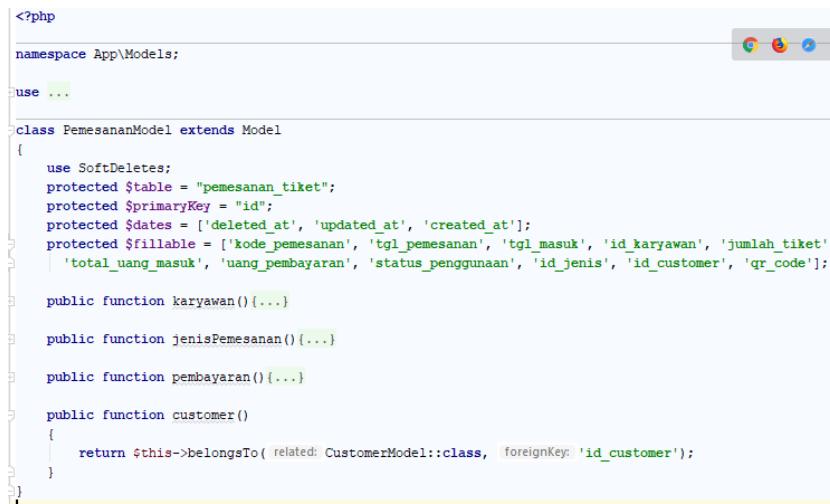
Gambar 5.12 Model KonfirmasiPembayaranModel

Berikut ini adalah deskripsi dari setiap method yang ada pada model KonfirmasiPembayaranModel :

1. Fungsi pemesananTiket untuk berelasi dengan kelas pemesanan tiket
2. Fungsi statusPembayaran untuk berelasi dengan kelas status pembayaran

### d. Model PemesananModel

Model KonfirmasiPembayaranModel berfungsi untuk mengambil data pemesanan customer dari *database* yang akan digunakan di aplikasi *web*. Dalam KonfirmasiPembayaranModel ini terdapat fungsi-fungsi yang dapat dilihat pada Gambar 5.13 :



```
<?php
namespace App\Models;

use ...;

class PemesananModel extends Model
{
    use SoftDeletes;
    protected $table = "pemesanan_tiket";
    protected $primaryKey = "id";
    protected $dates = ['deleted_at', 'updated_at', 'created_at'];
    protected $fillable = ['kode_pemesanan', 'tgl_pemesanan', 'tgl_masuk', 'id_karyawan', 'jumlah_tiket', 'total_uang_masuk', 'uang_pembayaran', 'status_penggunaan', 'id_jenis', 'id_customer', 'qr_code'];

    public function karyawan() {...}

    public function jenisPemesanan() {...}

    public function pembayaran() {...}

    public function customer()
    {
        return $this->belongsTo( related: CustomerModel::class, foreignKey: 'id_customer' );
    }
}
```

Gambar 5.13 Model PemesananModel

Berikut ini adalah deskripsi dari setiap method yang ada pada model PemesananModel :

1. Fungsi karyawan untuk berelasi dengan kelas karyawan
2. Fungsi jenisPemesanan untuk berelasi dengan kelas jenis pemesanan
3. Fungsi pembayaran untuk berelasi dengan kelas pembayaran
4. Fungsi customer untuk berelasi dengan kelas customer

#### e. Model CustomerModel

Model CustomerModel berfungsi untuk mengambil data customer dari *database* yang akan digunakan di aplikasi *web* dan *mobile*. Dalam CustomerModel ini terdapat fungsi-fungsi yang dapat dilihat pada yang dapat dilihat pada Gambar 5.14 :



```
<?php
namespace App\Models;

use ...;

class CustomerModel extends Authenticatable
{
    use SoftDeletes;
    protected $table = "user_customer";
    protected $guard = 'api_customer';
    protected $primaryKey = "id";
    protected $dates = ['deleted_at', 'updated_at', 'created_at'];
    protected $fillable = ['nama', 'email', "username", 'password', 'no_hp', 'images', 'status' , 'api_token', 'player_id'];

    protected $hidden = [
        'password' , 'api_token'
    ];

    public function pemesanan()
    {
        return $this->hasOne( related: PemesananModel::class, foreignKey: 'id_customer', localKey: 'id' );
    }
}
```

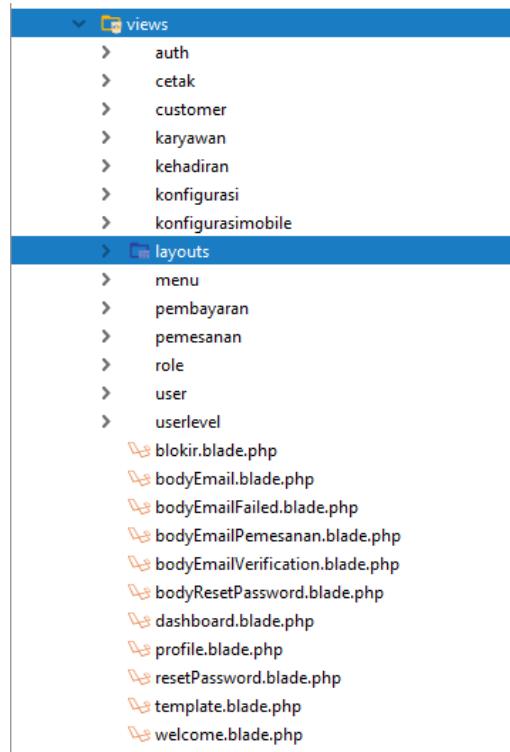
Gambar 5.14 Model CustomerModel

Berikut ini adalah deskripsi dari setiap method yang ada pada model CustomerModel :

1. Fungsi pemesanan untuk berelasi dengan kelas pemesanan

## 5. Struktur View Aplikasi Web

Berikut ini adalah view yang ada pada aplikasi web. Dapat dilihat pada Gambar 5.15.



Gambar 5.15 List View Aplikasi Web

## 6. Struktur Route Aplikasi Web

Berikut ini adalah route yang ada pada aplikasi web. Dapat dilihat pada

```

Route::get('auth', 'AuthController@index')->name('auth');
Route::get('blokir', 'AuthController@blokir');
Route::get('logout', 'AuthController@logout');
Route::get('role', 'ChooseRoleController@index');
Route::get('link', 'ChooseRoleController@linkDashboard');
Route::post('auth/checklogin', 'AuthController@checkLogin');

// modul profile
Route::group(['prefix' => 'profile'], function () {
    Route::get('/', 'ProfileController@index');
    Route::get('cekUsername', 'UserController@cekUsername');
    Route::post('update', 'ProfileController@update');
    Route::put('changepassword', 'ProfileController@changePassword');
});

Route::middleware(['checkRole'])->group(function () {
    Route::get('/', 'DashboardController@index');

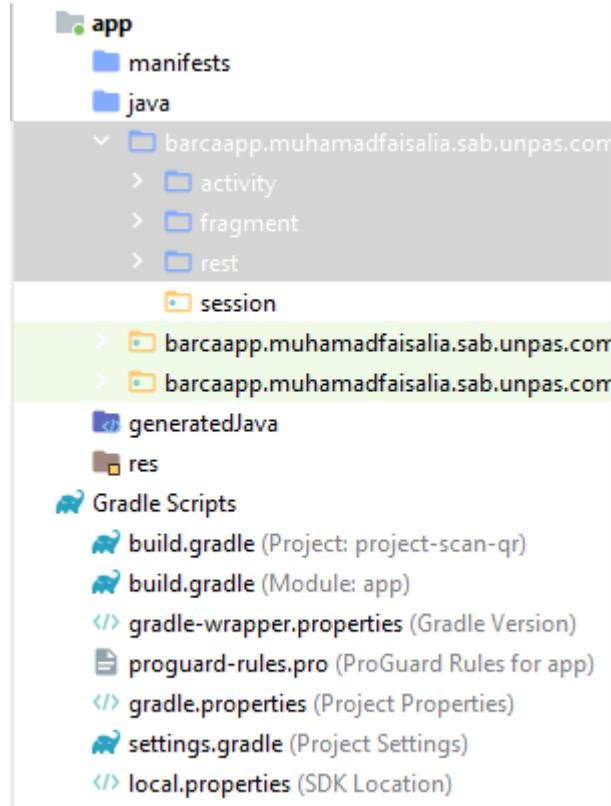
    // modul dashboard
    Route::group(['prefix' => 'dashboard'], function () {
        Route::get('/', 'DashboardController@index');
        Route::get('chart', 'DashboardController@chartForDataByDay');
        Route::get('notif', 'DashboardController@getNotification');
        Route::get('chart1', 'DashboardController@chartForDataByMonth');
        Route::put('update', 'DashboardController@updateNotif');
    });
});

```

Gambar 5.16 List Route Aplikasi Web

## 7. Struktur Aplikasi Mobile

Struktur yang dibangun menggunakan pola arsitektur MVP untuk memisahkan antara data (model) dengan user interface (view) serta logika (presenter). Untuk link repository terdapat pada lampiran B-1. Berikut ini adalah struktur pada aplikasi *mobile* dapat dilihat pada Gambar 5.17 :



Gambar 5.17 Struktur Directori Aplikasi Mobile

## 8. Koneksi Aplikasi *Mobile* dengan *Web Service*

Pada aplikasi *mobile* untuk terhubung dengan bagian server harus menggunakan *web service*. Berikut ini cara menghubungkan aplikasi *mobile* dengan bagian server bisa dilihat pada Gambar 5.18.

```
package barcaapp.muhamadfaisia.sab.unpas.com.qrcodescanner.rest;

import ...

public class ApiClient {
    public static final String BASE_URL = "https://demo.faidha-waterpark.com/api/";
    private static Retrofit retrofit = null;
    public static Retrofit getClient() {
        HttpLoggingInterceptor interceptor = new HttpLoggingInterceptor();
        interceptor.setLevel(HttpLoggingInterceptor.Level.BODY);
        OkHttpClient client = new OkHttpClient.Builder()
            .retryOnConnectionFailure(false)
            .connectTimeout( 30, TimeUnit.SECONDS)
            .writeTimeout( 30, TimeUnit.SECONDS)
            .readTimeout( 60, TimeUnit.SECONDS)
            .addInterceptor(interceptor)
            .build();
        if (retrofit==null) {
            retrofit = new Retrofit.Builder()
                .baseUrl(BASE_URL)
                .addConverterFactory(GsonConverterFactory.create())
                .build();
        }
        return retrofit;
    }
}
```

Gambar 5.18 Koneksi ke Web Service

## 9. Presenter Aplikasi *Mobile*

Presenter pada aplikasi *mobile* adalah sebagai controller yang berguna untuk melakukan pemrosesan data yang meminta data ke *database* melalui *web service*. Berikut ini adalah beberapa presenter pada aplikasi *mobile* yang dapat dilihat pada Table 5.6 :

Table 5.6 Daftar Presenter Aplikasi Mobile

No	Nama Model	Deskripsi
1.	DetailPemesananActivity	Untuk memesan tiket yang dipesan oleh customer menggunakan aplikasi <i>mobile</i> .
2.	PembayaranActivity	Untuk mengupload bukti pembayaran pemesanan tiket yang customer telah pesan.
3.	ScanActivity	Untuk memindai QR Code pada tiket yang dibawa customer.
4.	RegisterActivity	Untuk membuat akun customer jika ingin memesan tiket.

### a. Presenter DetailPemesananActivity

Presenter DetailPemesananActivity berfungsi untuk memesan tiket yang dipesan oleh customer menggunakan aplikasi *mobile*. Dalam DetailPemesananActivity ini terdapat fungsi-fungsi yang dapat dilihat pada Gambar 5.19 :

```
package com.example.faidawaterpark.activity;
import ...
public class DetailPemesananActivity extends AppCompatActivity {
    ...
    @Override
    protected void onCreate(Bundle savedInstanceState) {...}
    public void increment() {...}
    public void decrement() {...}
    private void display(int number) {...}
    public void prosesInput() {...}
    @Override
    public void onBackPressed() {...}
}
```

Gambar 5.19 Presenter DetailPemesananActivity

Berikut ini adalah deskripsi dari setiap method yang ada pada presenter DetailPemesananActivity :

1. Fungsi onCreate untuk menciptakan acitivity
2. Fungsi prosesInput untuk untuk merequest data pemesanan ke *web service*
3. Fungsi increment untuk menambah jumlah tiket
4. Fungsi decrement untuk mengurangi jumlah tiket
5. Fungsi display untuk menampilkan jumlah tiket
6. Fungsi onBackPressed untuk pindah ke activity yang dituju

### b. Presenter PembayaranActivity

Presenter PembayaranActivity berfungsi untuk mengupload bukti pembayaran pemesanan tiket yang customer telah pesan. Dalam PembayaranActivity ini terdapat fungsi-fungsi yang dapat dilihat pada Gambar 5.20 :

```

package com.example.faidawaterpark.activity;

import ...

public class PembayaranActivity extends AppCompatActivity {

    ...
    @Override
    protected void onCreate(Bundle savedInstanceState) {...}

    private void openGallery() {...}

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {...}

    @Override
    public void onBackPressed() {...}

    @Override
    public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {...}

    private String getRealPathFromUri(Uri uri) {...}

    public void prosesInput() {...}

    @NotNull
    private RequestBody createPartFromString(String descriptionString) {...}
}

```

Gambar 5.20 Presenter PembayaranActivity

Berikut ini adalah deskripsi dari setiap method yang ada pada presenter PembayaranActivity :

1. Fungsi onCreate untuk menciptakan acitivity
2. Fungsi prosesInput untuk untuk merequest data pemesanan ke *web service*
3. Fungsi openGallery untuk membuka folder gambar
4. Fungsi onRequestPermissionsResult untuk membuat request permission
5. Fungsi getRealPathFromUri untuk menampilkan gambar
6. Fungsi onBackPressed untuk pindah ke activity yang dituju
7. Fungsi createPartFromString untuk mengirim kode pemesanan ke *web service*

### c. Presenter RegisterActivity

Presenter RegisterActivity berfungsi untuk membuat akun customer jika ingin memesanan tiket.

Dalam RegisterActivity ini terdapat fungsi-fungsi yang dapat dilihat pada Gambar 5.21 :

```

package com.example.faidawaterpark.activity;

import ...

public class RegisterActivity extends AppCompatActivity {

    ...
    @Override
    protected void onCreate(Bundle savedInstanceState) {...}

    @Override
    public void onBackPressed() {...}

    private void prosesInput() {...}
}

```

Gambar 5.21 Presenter RegisterActivity

Berikut ini adalah deskripsi dari setiap method yang ada pada presenter RegisterActivity :

1. Fungsi onCreate untuk menciptakan acitivity
2. Fungsi prosesInput untuk untuk merequest data pemesanan ke *web service*
3. Fungsi onBackPressed untuk pindah ke activity yang dituju

#### d. Presenter ScanActivity

Presenter ScanActivity berfungsi untuk memindai QR Code pada tiket yang dibawa customer. Dalam ScanActivity ini terdapat fungsi-fungsi yang dapat dilihat pada Gambar 5.22 :

```
package barcaapp.muhammadfaisalia.sab.unpas.com.qrcodescanner.fragment;

import ...

/**
 * A simple {@link Fragment} subclass.
 */
public class ScanFragment extends Fragment {

    public ScanFragment() {}

    public static ScanFragment newInstance() { return new ScanFragment(); }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
}
```

Gambar 5.22 PresenterScanActivity

Berikut ini adalah deskripsi dari setiap method yang ada pada presenter ScanActivity :

1. Fungsi scanFragment sebagai constructor
2. Fungsi onCreateView untuk menciptakan fragment

#### 5.1.3 Implementasi Antarmuka Pengguna

Implementasi antarmuka pengguna mengacu pada perancangan antarmuka yang telah dibuat pada tahap sebelumnya. Berikut ini adalah implementasi antarmuka pengguna dari aplikasi web dan aplikasi *mobile*.

#### 1. Implementasi Antarmuka Pemesanan Tiket

No	Kode	Batas Masuk	Jumlah Tiket	Pembayaran	Status	Jenis	Aksi
No data available in table							
Showing 0 to 0 of 0 entries							
<a href="#">Previous</a> <a href="#">Next</a>							

2019 © Failda Waterpark Powered By Laravel 5.7 Version 1.0

Gambar 5.23 Implementasi antarmuka pemesanan tiket berbasis web

Deskripsi
Gambar diatas merupakan tampilan untuk halaman pemesanan tiket untuk tampil data berdasarkan hari. Data yang ditampilkan pada tabel yaitu kode pemesanan, batas masuk, jumlah tiket, total pembayaran, status pemesanan dan jenis pemesanan.

## 2. Implementasi Antarmuka Tambah dan Update Pemesanan Tiket

**Implementasi Antarmuka tambah dan update pemesanan tiket berbasis web**

Gambar 5.24 Implementasi Antarmuka tambah dan update pemesanan tiket berbasis web

Deskripsi
Gambar diatas merupakan tampilan untuk halaman tambah pemesanan tiket. Halaman ini bisa dijadikan untuk tambah data pemesanan tiket dan update data pemesanan tiket. Selain itu, untuk mempercepat dalam proses pemesanan disediakan empat buah button yang menunjukkan seringnya pengguna yang memesan tiket mulai dari satu tiket sampai empat tiket. Apabila lebih dari empat maka kasir harus menginput manual.

## 3. Implementasi Antarmuka konfirmasi pembayaran berbasis web

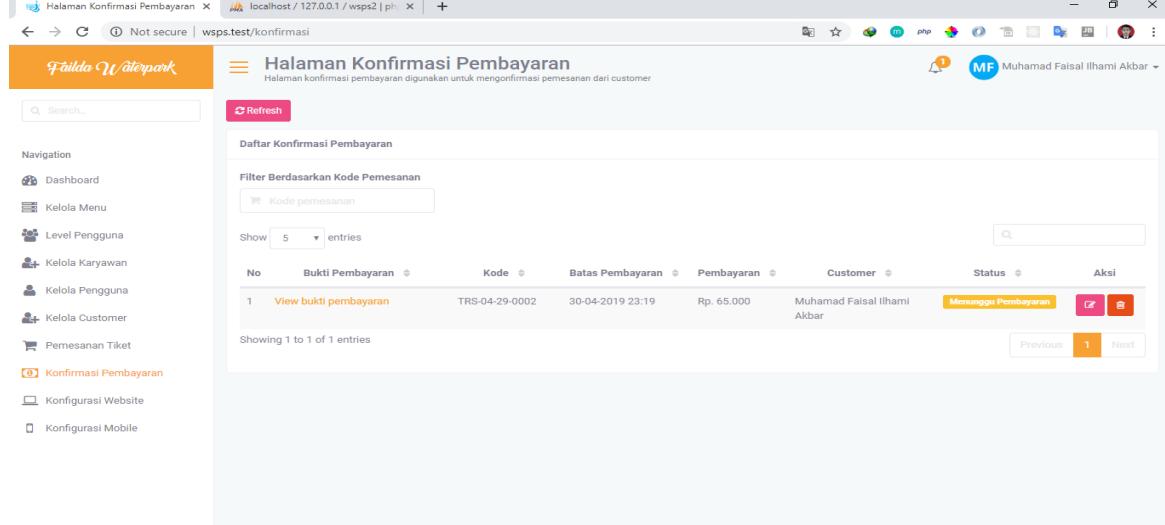
**Implementasi Antarmuka konfirmasi pembayaran berbasis web**

Gambar 5.25 Implementasi Antarmuka konfirmasi pembayaran berbasis web

Deskripsi
Gambar diatas merupakan tampilan untuk halaman konfirmasi pembayaran. Data yang ditampilkan pada tabel yaitu kode pemesanan, bukti pembayaran, batas pembayaran, pembayaran, nama customer, dan status pembayaran.

#### 4. Implementasi Antarmuka update status pembayaran tiket berbasis web

**Implementasi Antarmuka update status pembayaran tiket berbasis web**



No	Bukti Pembayaran	Kode	Batas Pembayaran	Pembayaran	Customer	Status	Aksi
1	<a href="#">View bukti pembayaran</a>	TRS-04-29-0002	30-04-2019 23:19	Rp. 65.000	Muhammad Faisal Ilhami Akbar	<span style="background-color: orange; color: white; padding: 2px 5px;">Menunggu Pembayaran</span>	 

Showing 1 to 1 of 1 entries

Previous  1 Next 

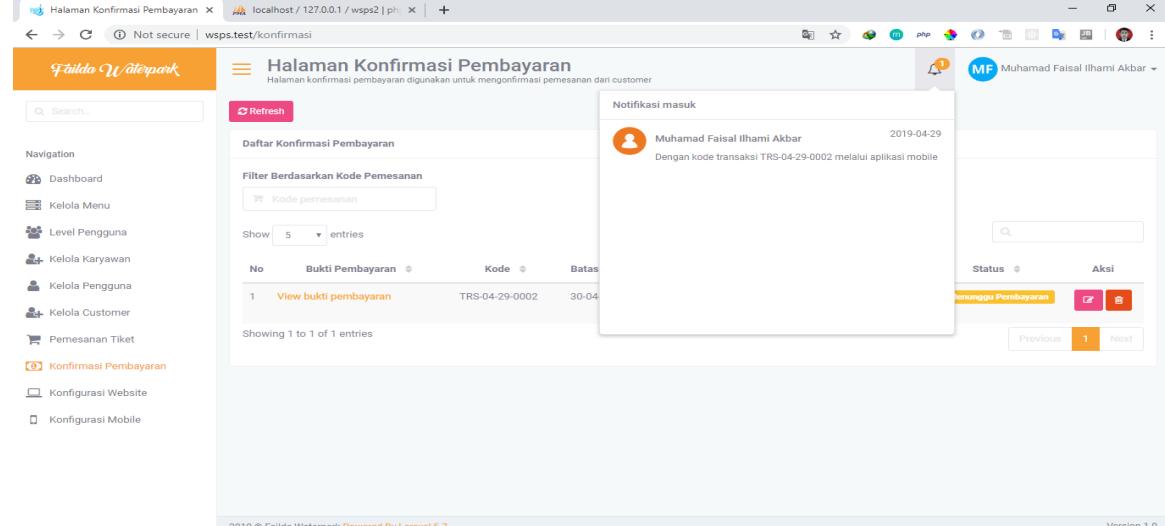
2019 © Failda Waterpark Powered By Laravel 5.7 Version 1.0

Gambar 5.26 Implementasi Antarmuka update status pembayaran tiket berbasis web

Deskripsi
Gambar diatas merupakan tampilan untuk halaman update status pembayaran pemesanan tiket dari customer. Untuk halaman ini yang berhak mengupdate yaitu admin saja.

#### 5. Implementasi Antarmuka menampilkan notifikasi pembayaran tiket berbasis web

**Implementasi Antarmuka menampilkan notifikasi pembayaran tiket berbasis web**



Notifikasi masuk

 Muhammad Faisal Ilhami Akbar
 2019-04-29

Dengan kode transaksi TRS-04-29-0002 melalui aplikasi mobile

Showing 1 to 1 of 1 entries

Previous  1 Next 

2019 © Failda Waterpark Powered By Laravel 5.7 Version 1.0

Gambar 5.27 Implementasi Antarmuka menampilkan notifikasi pembayaran tiket berbasis web

Deskripsi
Gambar diatas merupakan tampilan untuk halaman notifikasi pemesanan yang masuk dari customer yang melakukan pemesanan melalui aplikasi mobile.

## 6. Implementasi Antarmuka halaman kelola pengguna berbasis web

**Implementasi Antarmuka kelola pengguna berbasis web**

Gambar 5.28 Implementasi Antarmuka kelola pengguna berbasis web

No	Nama Lengkap	Username	Email	Level	Status	Aksi
1	Muhamad Faisal Ilhami Akbar	faisal27	faisal.ilhami1997@gmail.com	Administrator	Aktif	

**Deskripsi**

Gambar diatas merupakan tampilan untuk halaman kelola pengguna. Data yang ditampilkan pada tabel yaitu nama lengkap, username, email, level, dan status.

## 7. Implementasi Antarmuka tambah dan update kelola pengguna berbasis web

**Implementasi Antarmuka tambah dan update kelola pengguna berbasis web**

Gambar 5.29 Implementasi Antarmuka tambah dan update kelola pengguna berbasis web

**Deskripsi**

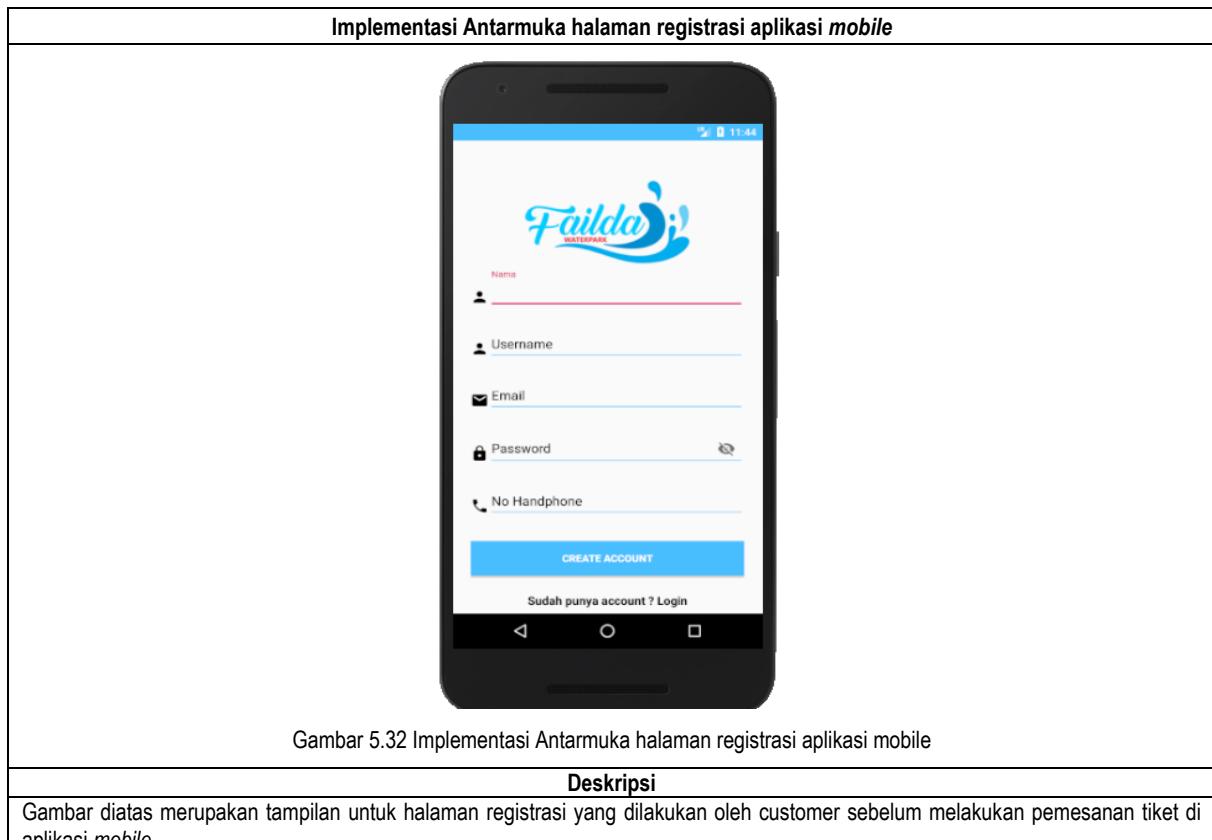
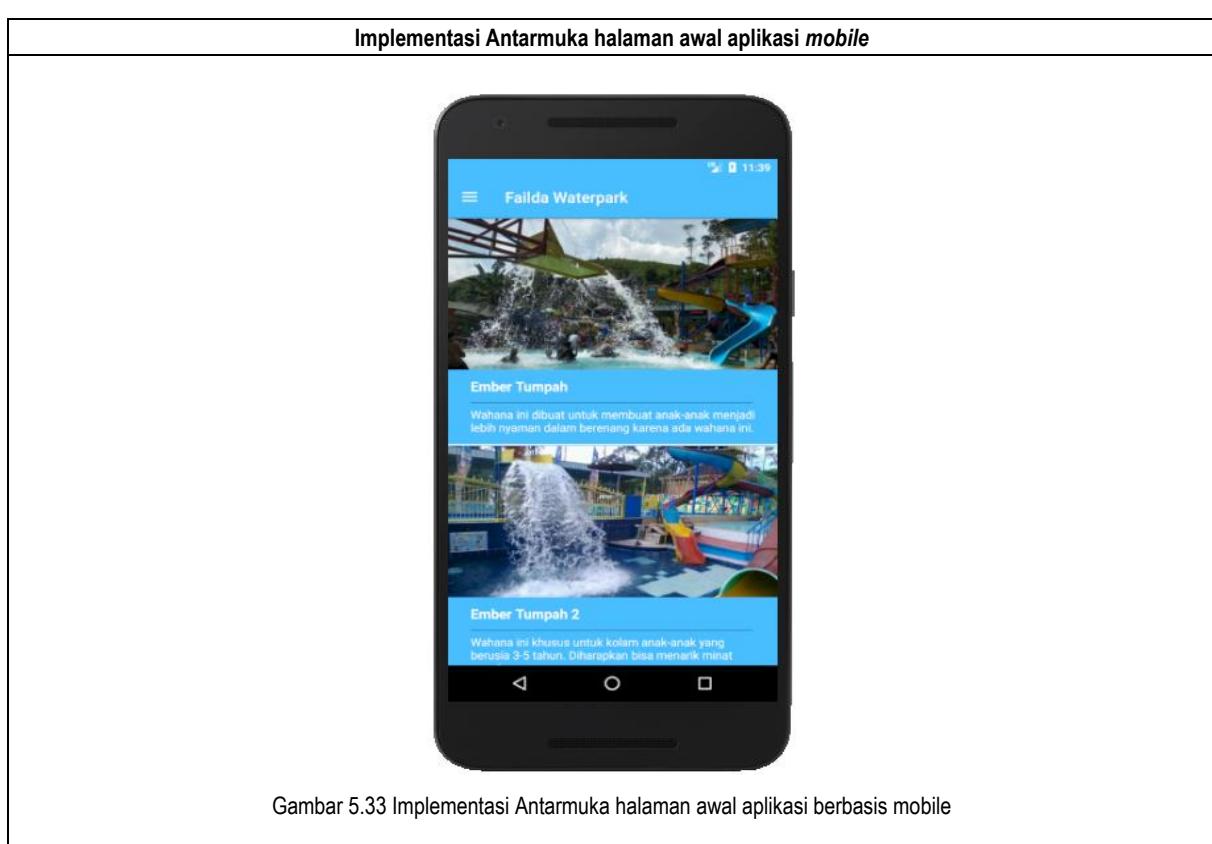
Gambar diatas merupakan tampilan untuk halaman update data karyawan. Untuk halaman ini yang berhak mengupdate yaitu admin saja.

## 8. Implementasi Antarmuka halaman cetak tiket

Implementasi Antarmuka cetak tiket berbasis web				
Gambar 5.30 Implementasi Antarmuka cetak tiket berbasis web				
<b>Deskripsi</b>				
Gambar diatas merupakan tampilan untuk halaman mencetak tiket yang dipesan oleh customer dengan datang langsung ke tempat.				

## 9. Implementasi Antarmuka halaman scan QR Code mobile

Implementasi Antarmuka halaman scan QR Code mobile	
Gambar 5.31 Implementasi Antarmuka halaman scan QR Code mobile	
<b>Deskripsi</b>	
Gambar diatas merupakan tampilan untuk halaman scan QR Code yang dilakukan oleh petugas penarik tiket yang melakukan pemindaian QR Code dari tiket yang telah dipesan customer.	

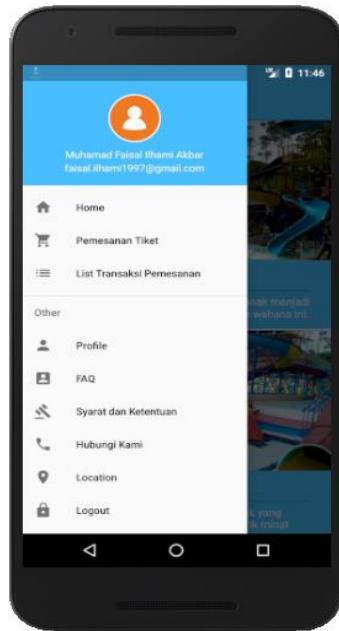
**10. Implementasi Antarmuka halaman registrasi aplikasi *mobile*****11. Implementasi Antarmuka halaman awal aplikasi *mobile***

<b>Deskripsi</b>
------------------

Gambar diatas merupakan tampilan untuk halama awal ketika sudah login di aplikasi mobile.
---

**12. Implementasi Antarmuka sidebar aplikasi mobile.**

<b>Implementasi Antarmuka sidebar aplikasi mobile</b>
---



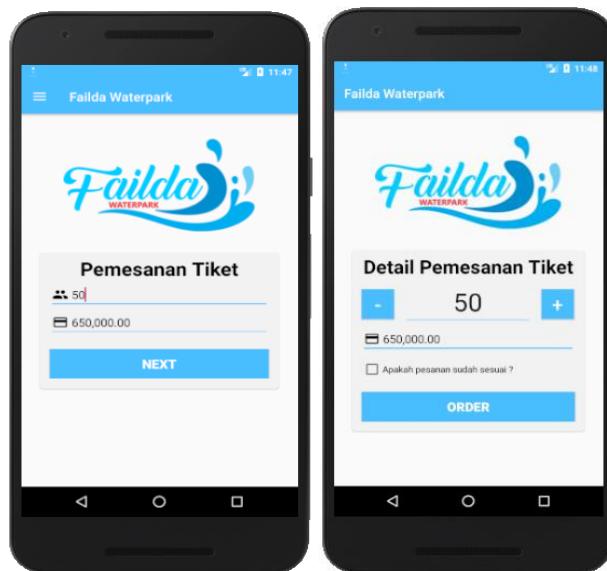
Gambar 5.34 Implemetnasi Antarmuka sidebar aplikasi berbasis mobile

<b>Deskripsi</b>
------------------

Gambar diatas merupakan tampilan untuk sidebar yang berisi menu-menu yang bisa digunakan oleh customer baik untuk memesan tiket, mengonfirmasi pembayaran, dan mengubah profil.
---

**13. Implementasi Antarmuka halaman pemesanan tiket aplikasi mobile.**

<b>Implementasi Antarmuka halaman pemesanan tiket aplikasi mobile</b>
---



Gambar 5.35 Implementasi Antarmuka halaman pemesanan tiket aplikasi mobile

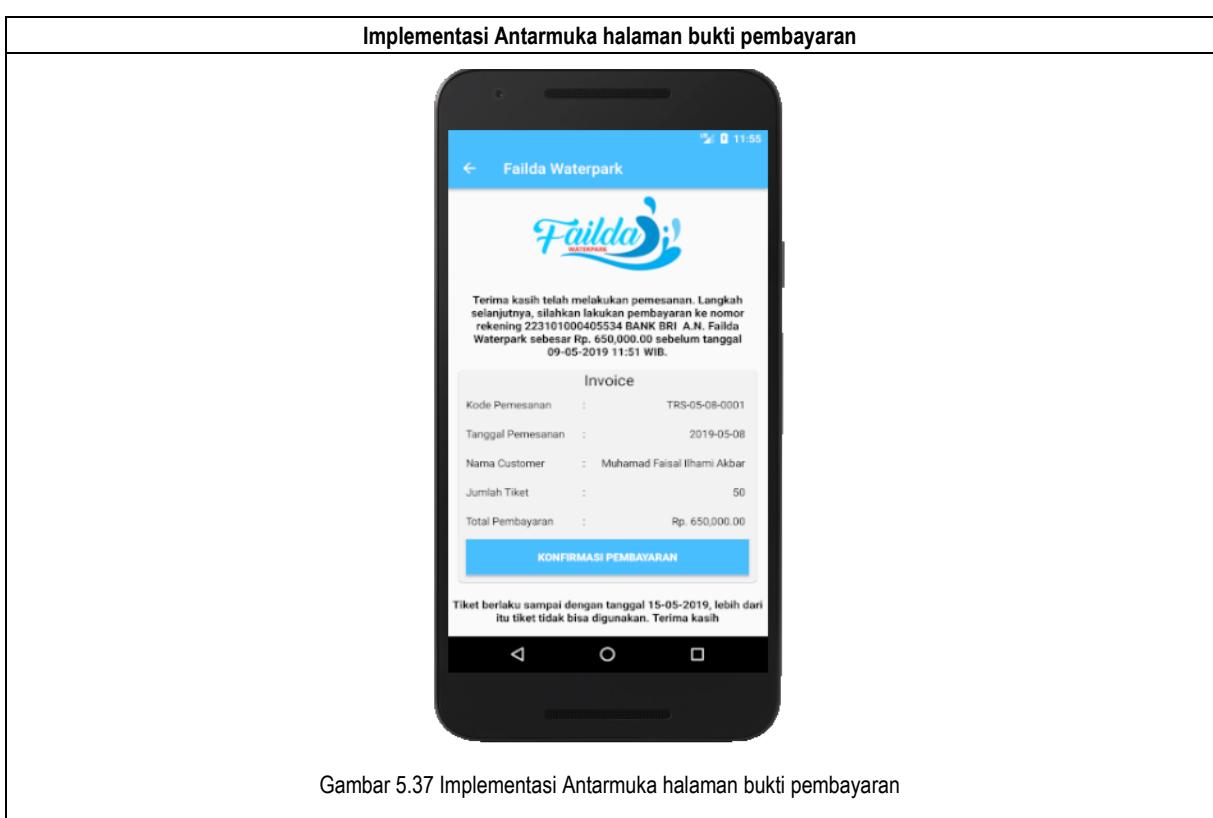
<b>Deskripsi</b>
------------------

Gambar diatas merupakan tampilan untuk halaman pemesanan tiket. Customer bisa harus mengisi tanggal masuk dan jumlah tiket yang akan dipesan, setelah itu akan dialihkan ke halaman detail pemesanan tiket untuk memastikan tiket yang dipesan sudah sesuai atau belum. Jika belum bisa merubahnya terlebih dahulu.

#### 14. Implementasi Antarmuka halaman konfirmasi pembayaran aplikasi mobile

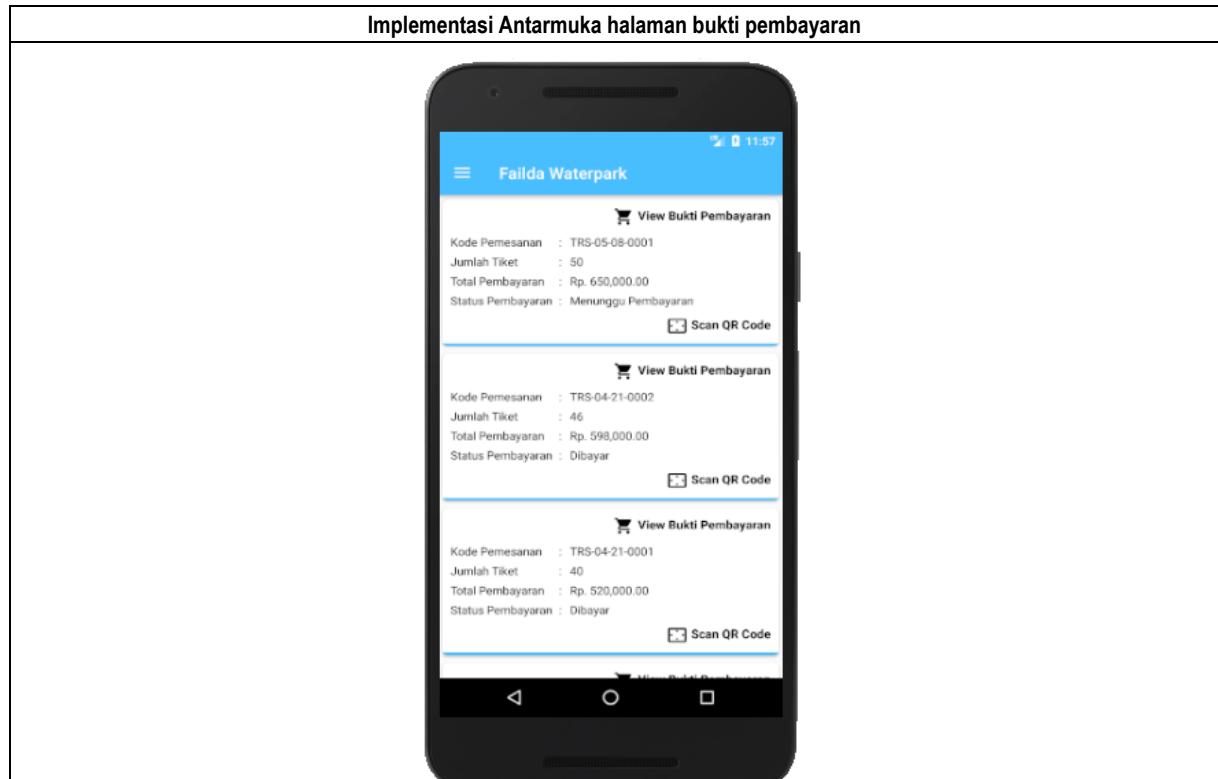


#### 15. Implementasi Antarmuka halaman bukti pembayaran



Deskripsi
Gambar diatas merupakan tampilan untuk halaman bukti pembayaran ketika customer sudah melakukan pemesanan.

#### 16. Implementasi Antarmuka halaman list pemesanan tiket



Gambar 5.38 Implementasi Antarmuka halaman list pemesanan tiket

Deskripsi
Gambar diatas merupakan tampilan untuk halaman list pemesanan tiket ketika customer sudah melakukan pemesanan.

#### 5.1.4 Pengujian Perangkat Lunak

Pengujian perangkat lunak yang akan dilakukan terhadap aplikasi *web* dan aplikasi *mobile* yaitu menggunakan metode *black box testing*. *Black box testing* adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak [PAS18]. Berikut ini merupakan pengujian perangkat lunak yang dilakukan.

##### 5.1.4.1 Pengujian Aplikasi Web

Berikut ini adalah hasil pengujian yang dilakukan pada aplikasi web. Diantaranya sebagai berikut :

###### 1. Cek Fungsional Login

Cek fungsional login ini bertujuan untuk menguji akun dari user apakah valid ketika login atau tidak. Tabel pengujian cek fungsional login memiliki satu test skenario dan empat test case, untuk table pengujian ada pada lampiran Table W-01.

## **2. Cek Fungsional Kelola Menu**

Cek fungsional kelola menu ini bertujuan untuk menguji data-data menu yang akan ditampilkan di aplikasi web. Tabel pengujian cek fungsional kelola menu memiliki satu test skenario dan sebelas test case, untuk table pengujian ada pada lampiran Table W-02.

## **3. Cek Fungsional Level Pengguna**

Cek fungsional level pengguna ini bertujuan untuk menguji data-data level pengguna yang ada di aplikasi web. Tabel pengujian cek fungsional level pengguna memiliki satu test skenario dan tiga test case, untuk table pengujian ada pada lampiran Table W-03.

## **4. Cek Fungsional Kelola Karyawan**

Cek fungsional kelola karyawan ini bertujuan untuk menguji data-data karyawan yang ada di aplikasi web. Tabel pengujian cek fungsional kelola karyawan memiliki satu test skenario dan tujuh test case, untuk table pengujian ada pada lampiran Table W-04.

## **5. Cek Fungsional Kelola Pengguna**

Cek fungsional kelola pengguna ini bertujuan untuk menguji data-data pengguna yang ada di aplikasi web. Tabel pengujian cek fungsional kelola pengguna memiliki satu test skenario dan sebelas test case, untuk table pengujian ada pada lampiran Table W-05.

## **6. Cek Fungsional Pemesanan Tiket**

Cek fungsional pemesanan tiket ini bertujuan untuk menguji data-data pemesanan tiket dari aplikasi web atau aplikasi *mobile*. Tabel pengujian cek fungsional pemesanan tiket memiliki satu test skenario dan tigas test case, untuk table pengujian ada pada lampiran Table W-06.

## **7. Cek Fungsional Pembayaran**

Cek fungsional pembayaran ini bertujuan untuk menguji data pembayaran dari pemesanan tiket yang dilakukan oleh customer di aplikasi *mobile*. Tabel pengujian cek fungsional pembayaran memiliki satu test skenario dan tiga test case, untuk table pengujian ada pada lampiran Table W-07.

## **8. Cek Fungsional Konfigurasi Website**

Cek fungsional konfigurasi website ini bertujuan untuk menguji data-data perusahaan yang ada di aplikasi web. Tabel pengujian cek fungsional konfigurasi website memiliki satu test skenario dan sebelas test case, untuk table pengujian ada pada lampiran Table W-08.

## **9. Cek Fungsional Konfigurasi Mobile**

Cek fungsional konfigurasi *mobile* ini bertujuan untuk menguji data-data berita yang akan ditampilkan di aplikasi *mobile*. Tabel pengujian cek fungsional konfigurasi *mobile* memiliki satu test skenario dan sembilan test case, untuk table pengujian ada pada lampiran Table W-09.

### **5.1.4.2 Pengujian Web Service**

Berikut ini adalah hasil pengujian yang dilakukan pada *web service*. Diantaranya sebagai berikut :

## 1. Cek Fungsional Login

Cek fungsional login ini bertujuan untuk menguji akun dari user apakah valid ketika login atau tidak. Tabel pengujian cek fungsional login memiliki satu test skenario dan empat test case, untuk table pengujian ada pada lampiran Table WS-01.

## 2. Cek Fungsional Customer

Cek fungsional customer ini bertujuan untuk menguji data-data customer yang akan ditampilkan di aplikasi *mobile*. Tabel pengujian cek fungsional customer memiliki satu test skenario dan dua test case, untuk table pengujian ada pada lampiran Table WS-02.

## 3. Cek Fungsional Pemesanan Tiket

Cek fungsional pemesanan tiket ini bertujuan untuk menguji data pemesanan tiket yang akan digunakan di aplikasi *mobile*. Tabel pengujian cek fungsional pemesanan tiket memiliki satu test skenario dan dua test case, untuk table pengujian ada pada lampiran Table WS-03.

## 4. Cek Fungsional Pembayaran

Cek fungsional pembayaran ini bertujuan untuk menguji data pembayaran yang akan digunakan di aplikasi *mobile*. Tabel pengujian cek fungsional pembayaran memiliki satu test skenario dan dua test case, untuk table pengujian ada pada lampiran Table WS-04.

### 5.1.4.3 Pengujian Aplikasi *Mobile*

Berikut ini adalah hasil pengujian yang dilakukan pada aplikasi *mobile*. Diantaranya sebagai berikut :

## 1. Cek Fungsional Login

Cek fungsional login ini bertujuan untuk menguji akun dari user apakah valid ketika login atau tidak. Tabel pengujian cek fungsional login memiliki satu test skenario dan empat test case, untuk table pengujian ada pada lampiran Table M-01.

## 2. Cek Fungsional Customer

Cek fungsional customer ini bertujuan untuk menguji data-data customer yang ada di aplikasi *mobile*. Tabel pengujian cek fungsional customer memiliki satu test skenario dan empat test case, untuk table pengujian ada pada lampiran Table M-02.

## 3. Cek Fungsional Pemesanan Tiket

Cek fungsional pemesanan tiket ini bertujuan untuk menguji data pemesanan tiket yang dipesan di aplikasi *mobile*. Tabel pengujian cek fungsional pemesanan tiket memiliki satu test skenario dan empat test case, untuk table pengujian ada pada lampiran Table M-03.

## 4. Cek Fungsional Pembayaran

Cek fungsional pembayaran ini bertujuan untuk menguji data pembayaran dengan mengupload bukti pembayaran di aplikasi *mobile*. Tabel pengujian cek fungsional pembayaran memiliki satu test skenario dan empat test case, untuk table pengujian ada pada lampiran Table M-04.

## 5.2 Deployment

Tahap deployment ini adalah tahap yang dilakukan untuk memindahkan projek yang ada di local ke server yang sudah siap dijalankan. Server harus diinstalasi terlebih dahulu sebelum projek di upload ke server. Untuk konfigurasi server akan menggunakan cPanel yang digunakan untuk mengelola projek dan konfigurasi-konfigurasi yang dibutuhkan. Untuk domain aplikasi web yaitu <http://demo.failda-waterpark.com>. Berikut ini adalah spesifikasi server dan konfigurasi yang akan dilakukan di server.

### 5.2.1 Spesifikasi Server

Berikut ini adalah spesifikasi server yang akan digunakan :

Table 5.7 Spesifikasi Server

No.	Item	Detail
1.	Programming Language	PHP 7.2.17
2.	Web Server	LiteSpeed
3.	Perl Version	5.16.3
4.	Operating System	Linux
5.	MYSQL Version	10.2.24-MariaDB
6.	Architecture System	X86_64
7.	Server Name	Srv60
8.	Kernel Version	3.10.0-962.3.2.lve1.5.25.6.el7.x86_64
9.	cPanel Version	78.0 (build 21)

### 5.2.2 Konfigurasi

Tahap konfigurasi yaitu tahap untuk membuat settingan pada server. Berikut ini adalah settingan yang akan dilakukan untuk keperluan aplikasi yang akan diupload. Diantaranya sebagai berikut :

1. Masuk ke akun cPanel
2. Buat database terlebih dahulu
3. Setelah database dibuat lalu tentukan user yang berhak mengakses database tersebut
4. Tentukan privilege untuk user yang telah ditambahkan
5. Tentukan versi PHP yang akan digunakan apabila akan diubah
6. Upload projek aplikasi dalam bentuk zip atau clone project dari Github atau Gitlab jika menyimpan projek di repository git.
7. Extract projek tersebut atau clone projek di direktori public\_html
8. Daftarkan domain untuk di SSL supaya domain bisa menggunakan https
9. Setelah tahapan diatas selesai, aplikasi bisa digunakan



## **BAB 6**

### **KESIMPULAN DAN SARAN**

Bab ini berisi mengenai hasil penelitian serta pernyataan yang didapat berdasarkan identifikasi masalah yang dikemukakan, serta keterkaitan dari semua tahap yang dilakukan dalam penelitian. Di dalamnya terdapat pula saran dan rekomendasi yang diusulkan untuk penelitian selanjutnya.

#### **6.1 Kesimpulan**

Berdasarkan hasil implementasi dan pengujian aplikasi pemesanan tiket dengan mengacu pada analisis dan perancangan yang telah dibuat maka kesimpulan yang dapat ditarik, antara lain:

1. Aplikasi sudah mampu mengelola pemesanan tiket baik dari pemesanan secara langsung atau menggunakan aplikasi *mobile*.
2. Aplikasi sudah mampu menghandle pemesanan tiket berbasis *mobile*.
3. *Web Service* sudah dapat digunakan untuk bertukar data untuk aplikasi *mobile*.
4. Bahasa pemrograman PHP dengan menggunakan Laravel untuk membangun aplikasi web dan *web service* serta bahasa pemrograman Java untuk membangun aplikasi *mobile* sudah diterapkan.

#### **6.2 Saran**

Berdasarkan hasil penelitian, maka penulis menyarankan beberapa hal mengenai aplikasi pemesanan tiket dikolam renang Failda, antara lain :

1. Tampilan untuk aplikasi *mobile* dapat diperbaiki kembali sehingga menjadi lebih menarik.
2. Pengembangan lainnya yaitu menambahkan web untuk promosi kolam renang.
3. Pengembangan selanjutnya bisa ditambahkan fitur untuk mengelola pembayaran dengan adanya potongan harga.
4. Pengembangan selanjutnya bisa menambahkan testing non fungsional.

#### **6.3 Rekomendasi**

Berikut ini adalah rekomendasi yang dapat penulis berikan untuk pengembangan aplikasi pemesanan tiket di kolam renang Failda dan pihak kolam renang Failda, antara lain :

1. Pengembangan berikutnya dapat menambahkan fitur lain seperti kelola keuangan pada aplikasi web serta menambahkan *platform mobile* yang lain seperti ios.
2. Dapat terintegrasi dengan jasa pihak ketiga untuk *payment gateway* sehingga pembayaran otomatis terkonfirmasi ketika customer sudah melakukan pembayaran.
3. Memasang akses internet untuk memudahkan menggunakan aplikasi web dalam mengelola pemesanan tiket.



## DAFTAR PUSTAKA

- [AMI15] Aminudin, Cara Efektif Belajar Framework Laravel, Yogyakarta: Lokomedia, 2015.
- [AEP18] A. E. Prasetyadi, "Web 3.0: Teknologi Web Masa Depan," October 2011. [Online]. Available: <http://jurnal.unnar.ac.id/index.php/indept/article/view/78>. [Accessed 28 Agustus 2018].
- [ASU10] A. Sunyoto, "AJAX (Asynchronous JavaScript and XML)," *JURNAL DASI*, vol. 11, pp. 2 - 3, 2010.
- [ANA10] A. N. Akbar, "Perangkat Lunak Untuk Layanan Warga Berbasis Android Di Lingkungan Rukun Warga Menggunakan M-Goverment," 2017.
- [AAN14] A. ANWAR, "A Review of RUP (Rational Unified Process)," *International Journal of Software Engineering (IJSE)*, vol. 5, no. 2, p. 8, 2014.
- [ASU16] A. Sulihati, "Aplikasi Akademik Online Berbasis Mobile Android Pada Universitas Tama Jagakarsa," *Jurnal Sains dan Teknologi*, vol. XI, p. 20, 2016.
- [APB18] A. P. Beni, A. Hijriani and A. Junaidi, "Perancangan Application Programming Interface (Api) Berbasis Web Menggunakan Gaya Arsitektur Representational State Transfer (Rest) Untuk Pengembangan Sistem Informasi Administrasi Pasien Klinik Perawatan Kulit," *Jurnal Komputasi*, vol. 6, p. 33, 2018.
- [ADA17] A. Darmaanta, "Perancangan Aplikasi Marketplace Tiket Acara Online Berbasis Web di Kota Salatiga," p. 7, 2017.
- [BRA18] B. Raharjo, Belajar9 Otodidak Framework Codeigniter, Bandung: Penerbit Informatika, 2015.
- [CAM19] Cambridge, "Cambridge Dictionary," [Online]. Available: <https://dictionary.cambridge.org/dictionary/english/ticket>. [Accessed 4 February 2019].
- [DIG19] Digitiket, "Tiket Digital Wahana Rekreasi," PT. Digitiket, [Online]. Available: <https://digitiket.com/>. [Accessed 29 Juni 2019].
- [EHI15] E. Hidayat, "Web Service Untuk E-Voting (Studi Kasus Pemilu Raya HMTIF UNPAS)," p. 27, 2015.
- [FSA17] F. Salamun, "Pembangunan Aplikasi Sistem Informasi Bandung Nmax Community," 2017.
- [GNG06] G. Ng-Kruelle and P. A. Swatman, "E-Ticketing Strategy and Implementation in an Open Access System m: The case of Deutsche Bahn," in *reasearchgate.net*, 2006.

- [GTN19] G. T. Network, "Tiket.com," PT. Global Tiket Network, [Online]. Available: <https://www.tiket.com/>. [Accessed 29 Juni 2019].
- [HDE11] H. Deviana, "Penerapan XML Web service Pada Sistem Distribusi Barang," *Jurnal Generic*, vol. 6, p. 62, 2011.
- [HMU17] H. Mubarak, "Pembangunan Aplikasi Web Pengelolaan Nilai Siswa Smp Berbasis Kurikulum 2013 Menggunakan Web App Framework Codeigniter," 2017.
- [IBG16] I. B. G. & E. A. Dagan, "Delivery of QR Codes to Celluar Phones throught Data Embedding in Audio," in *International Conference on the Science of Electrical Engineering*, 2016.
- [JRO08] J. Rouillard, Contextual QR Codes, Proceedings of the Third International Multi - Conference on Computing in the Global Information Technology, Athens: ICCGI, 2008.
- [JSO00] JSON, "Introducing JSON", Diakses pada tanggal 03 September 2018 , Diambil dari json.org, 2000
- [KEL95] Kelleher, Kevin, Casey G., Lois D., et al, "Cause and Effect Diagram : Plain and Simple", Joiner Associates Inc USA, 1995
- [KRE01] Kreger, Heather., "Web services Conceptual Architecture", IBM Software Group, 2001.
- [LRA13] L. R. a. M. Amundsen, RESTful Web APIs, Farnham O'Reilly, 2013.
- [MAS17] M. A. Saputra, "Pengembangan Aplikasi Pemesanan Tiket Bus Pada PO. Puspa Jaya Berbasis Android," pp. 6-12, 2017.
- [MAR17] M. Arslan, "6 Library PHP untuk Web Scraping," 17 September 2017. [Online]. Available: <https://www.codepolitan.com/6-library-php-untuk-web-scraping-59bb3e168ceb1>. [Accessed 05 April 2019].
- [MKA13] M. Kalin, Java Web Services: Up and Running, Second edition., Sebastopol: California O'Reilly, 2013.
- [MTA18] M. T. A. L. F. Mochammad Syaifullah Ferryansyah, "Analisis Performansi HTTP Networking Library pada Android (Studi Kasus: Portal Berita)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, p. 2027, 2018.
- [MUT16] Mutmainah, "Sistem Informasi Pemesanan Tiket Kapal Laut Berbasis Web (Studi Kasus PT. Paradigma Baru Luwuk Banggai, Sulawesi Tengah)," p. 13, 2016.

- [NES19] N. E. Sejahtera, "Tix Id," PT. Nusantara Elang Sejahtera, [Online]. Available: <https://www.tix.id/>. [Accessed 29 Juni 2019].
- [PAS18] P. Astuti, "PENGGUNAAN METODE BLACK BOX TESTING (BOUNDARY VALUE ANALYSIS) PADA SISTEM AKADEMIK (SMA/SMK)," vol. 11, p. 188, 2018.
- [PHI17] P. Hidayatullah and J. K. Kawistara, Pemrograman Web, Bandung: Penerbit Informatika, 2017.
- [HMJ17] H. M. J. Rezki Orientani, "Kualitas Layanan Tiket Elektronik Terhadap Kepuasan Konsumen," *Jurnal Riset Bisnis dan Manajemen*, vol. 10, p. 42, 2017.
- [PRE10] Pressman, Roger. S, Software Engineering: A Practitioner's Approach, Seventh Edition, New York: McGraw-Hill, a business unit of The McGraw-Hill Companies, Inc, 2010.
- [NSA11] N. Safaat H, Pemrograman Aplikasi Mobile, 2011: Penerbit Informatika, Smartphone dan Tablet PC Berbasis Android.
- [SAR12] S. Arif, "Rancang Bangun Website Program Studi DIV Komputer Multimedia STIKOM Surabaya," p. 6, 2012.
- [SBD09] S. B. Doro Edi, "Analisis Data dengan Menggunakan ERD dan Model Konseptual Data Warehouse," *Jurnal Informatika*, vol. 5, p. 75, 2009.
- [SSU04] S. Surajino, "Pembelajaran Berbasis Web: Suatu Tujuan dari aspek Kognitif," in *Makalah Lokakarya metode Pembelajaran Berbasis Web-Departemen*, Bandung, 2004.
- [THE01] The Stencil Group., "Defining Web services", tersedia : 14 Oktober 2014, [www.perfectXML.com/Xanalysis/TSG/TSG\\_DefiningWebServices.pdf](http://www.perfectXML.com/Xanalysis/TSG/TSG_DefiningWebServices.pdf), 2001.
- [TIN18] T. Inc, "Mobile Application (Mobile App)," 2018. [Online]. Available: <https://www.techopedia.com/definition/2953/mobile-application-mobile-app>. [Accessed 20 November 2018].
- [TAN08] T. Anggraini, "Pelayanan Reservasi Sistem On Line Tiket Pesawat Domestik Lion Air Di Pt. Electra Duta Wisata Surakarta," p. 10, 2008.
- [UEY15] U. E. Y. N. Kunang and S. N. , "Implementasi Metode Rational Unified Process Pada Mobile Digital Library," *Jurnal Ilmiah MATRIK*, vol. 16, pp. 37-40, 2015.
- [W3C04] W3C Working Group., "Web services Architecture", tersedia : 14 Oktober 2014, <http://www.w3.org/TR/ws-arch/>, 2004

- [NKY08] N. K. Yuli Angkasa, "Peranan Teknologi Informasi Dalam E-Ticketing Pesawat Udara," *Jurnal Aviasi Langit*, vol. 5, p. 28, 2008.
- [YRD17] Y. R. Dawood and R. , "Rancang Bangun Layanan Web (Web Service) Untuk Aplikasi Rekam Medis Praktik Pribadi Dokter," *Jurnal Online Teknik Elektro*, vol. 2, pp. 2-3, 2017.
- [ZPJ16] Z. P. Juhara, Panduan Lengkap Pemrograman Android, Jakarta: Andi, 2016.

## **LAMPIRAN**



**LAMPIRAN 1**  
**BERITA ACARA WAWANCARA**

<b>BERITA ACARA WAWANCARA</b>		
No.	Daftar Pertanyaan	Jawaban
1.	Berapa banyak jumlah pengunjung setiap harinya ?	Untuk jumlah pengunjung setiap harinya tidak tentu, rata-rata di kisaran 100 – 150 orang untuk hari biasa, untuk hari libur sabtu-minggu bisa sampai 100 – 700 pengunjung.
2.	Berapa banyak jumlah pengunjung yang pernah datang ke kolam renang Failda ?	Untuk jumlah pengunjung yang paling banyak sampai saat ini kurang lebih sekitar 1600 pengunjung dalam satu hari. Itu pun ketika libur sekolah atau libur hari raya.
3.	Berapa harga tiket saat ini ?	Harga tiket saat ini Rp. 13.000.
4.	Apakah tiket pada saat hari biasa dan hari libur berbeda ?	Tidak, tiket untuk hari biasa dan hari libur sama Rp. 13.000
5.	Dari jam berapa mulai buka kolam renang Failda ?	Buka dari jam 7 pagi sampai jam 4 sore.
6.	Siapa saja yang terlibat dalam proses pemesanan tiket ?	Yang terlibat untuk proses pemesanan tiket yaitu bagian kasir yang melayani pengunjung yang akan masuk ke kolam renang dan bagian petugas penarik tiket yang bertugas menarik tiket dari pengunjung sebelum masuk ke kolam renang.
7.	Apa jenis tiket yang digunakan ?	Tiket yang digunakan yaitu menggunakan kertas biasa.
8.	Apakah sering terjadi antrian panjang ketika pengunjung sedang banyak yang datang ?	Sering, ketika hari libur.
9.	Bagaimana proses pemesanan tiket saat ini ?	Untuk proses pemesanan tiket saat ini mulai dari pengunjung datang lalu memesan tiket di bagian kasir lalu membayar tiket tersebut, lalu tiket tersebut akan ditarik kembali oleh petugas

		penarik tiket sebelum pengunjung masuk ke kolam renang.
10.	Bagaimana proses rekapitulasi data tiket ?	Prosesnya yaitu dengan menghitung tiket yang terjual dan menyesuaikan dengan uang yang masuk.
11.	Kapan dilakukan rekapitulasi data tiket ?	Setelah tutup kolam renang.
12.	Apakah ada kendala dalam proses pemesanan tiket ?	Ada, kendalanya yaitu sering terjadi ketidak sesuaian antara tiket yang terjual dan uang yang masuk.
13.	Apa yang menyebabkan hal itu terjadi ?	Seringnya tiket itu rusak dan pernah juga hilang.
14.	Apakah ada alat bantu seperti menggunakan aplikasi dalam proses pemesanan tiket ?	Tidak ada.

Mengetahui,

Pewawancara,

( Muhamad Faisal Ilhami Akbar )  
NRP: 153040044

Narasumber,

( Lili Kusnadi, S.I.P )

**LAMPIRAN 2**  
**LINK REPOSITORY APLIKASI**

1. Aplikasi Web dan web service : <https://gitlab.com/faisal27/laravel-project-ta-second.git>
2. Aplikasi *Mobile* : <https://gitlab.com/faisal27/project-order-ticket-android.git>
3. Aplikasi Scan QR Code : <https://gitlab.com/faisal27/project-scan-qr.git>



**LAMPIRAN 3**  
**TABLE PENGUJIAN PERANGKAT LUNAK**

**1. Pengujian Aplikasi Web**

Table Pengujian W-01 Fungsional Login

Code	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Pass /Fail
TS-001	Cek fungsional login	Username dan password valid	1. Username: faisal27 2. Password: 12345678	Masuk ke halaman dashboard	Sama seperti expected result	Pass
TS-002		Username tidak valid dan password valid	1. Username: failsa27 2. Password: 12345678	Muncul message username tidak valid/tidak terdaftar	Sama seperti expected result	Pass
TS-003		Username valid dan password tidak valid	1. Username: faisal27 2. Password: barca1899	Muncul message password salah	Sama seperti expected result	Pass
TS-004		Username dan password tidak valid	1. Username: failsa27 2. Password: barca1899	Muncul message Username& password salah	Muncul message Username tidak terdaftar	Pass

Table Pengujian W-02 Fungsional Kelola Menu

Code	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Pass /Fail
TS-001	Cek fungsional Kelola Menu	Menambahkan data pada menu jika user tidak menginputkan title	Title : URL : PROFILE Icon : icon icon-user Nomor urut : 1 Main Menu : Main Menu Status : Aktif	Muncul pesan field harus diisi	Sama seperti Expected Result	Pass
TS-002		Menambahkan data pada menu jika user tidak menginputkan URL	Title : PROFILE URL : Icon : icon icon-user Nomor urut : 1 Main Menu : Main Menu Status : Aktif	Muncul pesan field harus diisi	Sama seperti Expected Result	Pass
TS-003		Menambahkan data pada menu jika user tidak menginputkan Icon	Title : PROFILE URL : PROFILE Icon : Nomor urut : 1 Main Menu : Main Menu Status : Aktif	Muncul pesan field harus diisi	Sama seperti Expected Result	Pass
TS-004		Menambahkan data pada menu jika user tidak menginputkan Nomor Urut	Title : PROFILE URL : PROFILE Icon : icon icon-user Nomor Urut : Main Menu : Main Menu Status : Aktif	Muncul pesan field harus diisi	Sama seperti Expected Result	Pass
TS-005		Menambahkan data pada	Title : PROFILE URL : PROFILE	Muncul pesan field harus diisi	Sama seperti Expected Result	Pass

Code	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Pass /Fail
	TS-006	menu jika user tidak Memilih Main menu	Icon : icon icon-user Nomor Urut : Main Menu : Status : Aktif			
TS-006		Menambahkan data pada menu jika user tidak memilih status	Title : PROFILE URL : PROFILE Icon : icon icon-user Nomor Urut : Main Menu : Main Menu Status :	Muncul pesan field harus diisi	Sama seperti Expected Result	Pass
TS-007		Menambahkan data pada menu jika user mengisi semua form	Title : PROFILE URL : PROFILE Icon : icon icon-user Nomor Urut : 1 Main Menu : Main Menu Status : Aktif	Data Berhasil Ditambahkan	Sama seperti Expected Result	Pass
TS-008		Mengubah data menu jika memilih main menu Kelola Menu	Choose Main Menu : Kelola Menu	Dashboard menjadi sub menu pada Kelola Menu	Sama seperti Expected Result	Pass
TS-009		Mengubah nama title	Title : PROFIL	Muncul pesan berhasil	Sama seperti Expected Result	Pass
TS-010		Mengubah url	Title : profileAdmin	Muncul pesan berhasil	Sama seperti Expected Result	Pass
TS-011		Menghapus menu profile	-	Muncul pesan berhasil	Sama seperti Expected Result	Pass

Table Pengujian W-03 Fungsional Level Pengguna

Code	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Pass /Fail
TS-001	Cek fungsional level pengguna	Menambahkan level pengguna Direktur	Nama Level : Direktur	Nama Level Berhasil ditambahkan	error	Fail
TS-002		Menubah level pengguna	Nama Level : Direktur	Nama Level Berhasil diubah	Sama seperti Expected Result	Pass
TS-003		Menghapus level pengguna	-	Level pengguna berhasil dihapus	Sama seperti Expected Result	Pass

Table Pengujian W-04 Fungsional Level Pengguna

Code	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Pass /Fail
TS-001	Cek fungsional Kelola Karyawan	Menambahkan data karyawan jika nama lengkap tidak diisi	Email : <a href="mailto:dahliaannisa26@gmail.com">dahliaannisa26@gmail.com</a> Nomor Hp: 08985953069	Muncul pesan error	Sama seperti Expected Result	Pass
TS-002		Menambahkan data karyawan jika Email lengkap tidak diisi	Nama Lengkap: Dahlia Annisa Nomor Hp: 08985953069	Muncul pesan error	Sama seperti Expected Result	Pass
TS-003		Menambahkan data karyawan jika Nomor	Nama Lengkap : Dahlia Annisa	Muncul pesan error	Sama seperti Expected Result	Pass

Code	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Pass /Fail
	TS-004	Handphone tidak diisi	Email : <a href="mailto:dahliaannisa26@gmail.com">dahliaannisa26@gmail.com</a>			
		Menambahkan data karyawan jika semua form diisi	Nama Lengkap : Dahlia Annisa Email : <a href="mailto:dahliaannisa26@gmail.com">dahliaannisa26@gmail.com</a> Nomor Hp: 08985953069	Data Berhasil Ditambahkan	Sama seperti Expected Result	Pass
		Menambahkan data karyawan jika Nomor Handphone tidak semua berisi angka	Nama Lengkap : Dahlia Annisa Email : <a href="mailto:dahliaannisa26@gmail.com">dahliaannisa26@gmail.com</a> Nomor Hp: 0896hcsghacjajah	Invalid, harus memasukkan angka	Sama seperti Expected Result	Pass
		Mengubah Nomor Handphone Salah satu Karyawan	Nomor Hp: 08985953069	Data Berhasil Diubah	Sama seperti Expected Result	Pass
		Mnghapus Salah satu data Karyawan	-	Data Berhasil Dihapus	Sama seperti Expected Result	Pass

Table Pengujian W-05 Fungsional Kelola Pengguna

Code	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Pass /Fail
	Cek fungsional Kelola Pengguna	Menambahkan Data pengguna	Karyawan : Testing Username : dahlia26 Password : 12345678 Level : Administrator Status : Aktif	Data berhasil ditambahkan dan menampilkan data karyawan baru	Akun sudah ada pada system	
TS-001		Mereset Password	-	Password Berhasil di reset	Sama seperti Expected Result	Pass
TS-002		Menghapus salah satu Pengguna	-	Data Berhasil Dihapus	Sama seperti Expected Result	Pass
TS-003						

Table Pengujian W-06 Fungsional Pemesanan Tiket

Code	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Pass /Fail
TS-001	Cek fungsional pemesanan tiket	Update jumlah tiket	1. Jumlah tiket: 5 2. Jumlah tiket: 8	Muncul message data berhasil diubah Data diubah	Sama seperti expected result	Pass
TS-002		Tambah jumlah tiket	1. Jumlah tiket: 2 2. Jumlah uang : 30000	Data berhasil disimpan	Sama seperti expected results	Pass
TS-003		Hapus record	-	Data berhasil dihapus	Muncul message data berhasil dihapus Data terhapus	Pass

Table Pengujian W-07 Fungsional Pemesanan Tiket

Code	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Pass /Fail
TS-001	Cek fungsional Konfirmasi Pembayaran	Update status	Ubah status ke dibayar	Muncul message data berhasil diubah Data diubah	Sama seperti expected result	Pass
TS-002		Update status	Ubah status ke tidak sesuai	Muncul message data berhasil diubah Data diubah	Sama seperti expected result	Pass
TS-003		Hapus record	-	Data berhasil dihapus	Muncul message data berhasil dihapus Data terhapus	Pass

Table Pengujian W-08 Fungsional Konfigurasi Website

Code	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Pass /Fail
TS-001	Cek fungsional konfigurasi website	Update nama perusahaan	Nama Perusahaan: Failda Waterpark Update Nama Perusahaan: Failda Waterpark Reza	Data Berhasil diubah	Sama seperti expected result	Pass
TS-002		Update pemilik perusahaan	Pemilik Perusahaan: Lily Kusnadi, S.IP Update Pemilik Perusahaan: Lily Kusnadi, S.IP,S.T	Data Berhasil diubah	Sama seperti expected result	Pass
TS-003		Update email perusahaan	Email perusahaan: failda.waterpark06@gmail.com Update Email perusahaan: failda.waterpark06@gmail.co.id	Data Berhasil diubah	Sama seperti expected result	Pass
TS-004		Update Nomor Telepon	Nomor Telepon: 085624420395 Update Nomor Telepon: 085624420396	Data Berhasil diubah	Sama seperti expected result	Pass
TS-005		Update harga tiket	Harga tiket: 13.000 Update Harga tiket: 15.000	Data Berhasil diubah	Sama seperti expected result	Pass
TS-006		Update versi aplikasi	Versi aplikasi: 1.0 Update Versi aplikasi: 2.0	Data Berhasil diubah	Sama seperti expected result	Pass
TS-007		Update Nama Bank	Nama Bank: Bank BRI	Data Berhasil diubah	Sama seperti expected result	

Code	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Pass /Fail
	TS-008		Update Nama Bank: Bank BCA			Pass
		Update Nomor Rekening	Nomor Rekening: 22310100040553 4	Data Berhasil diubah	Sama seperti expected result	Pass
			Update Nomor Rekening: 22310100040553 5			
		Update Pemilik Rekening	Pemilik Rekening: Failda Waterpark Update Pemilik Rekening: Failda Waterpark Reza	Data Berhasil diubah	Sama seperti expected result	Pass
		Update Reset Password	Password: failda2006 Update Password: failda2008	Data Berhasil diubah	Sama seperti expected result	Pass
	TS-011	Update Logo Perusahaan	Pilih gambar Klik open	Data Berhasil diubah	Sama seperti expected result	Pass

Table Pengujian W-09 Fungsional Konfigurasi Mobile

Code	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Pass /Fail
TS-001	Cek fungsional konfigurasi mobile	Title,Deskripsi,image diisi	1. Title: testing 2. Deskripsi: testing 3. Images: menu.PNG	Message Data berhasil ditambahkan Data muncul dihalaman konfigurasi mobile	Sama seperti expected result	Pass
TS-002		Title diisi,Deskripsi dan Image kosong	1. Title: testing 2. Deskripsi: null 3. Image: null	Muncul message the given data was invalid	Sama seperti expected result	Pass
TS-003		Title kosong,Deskripsi diisi dan Image kosong	1. Title: null 2. Deskripsi: testing 3. Image: null	Muncul message the given data was invalid	Sama seperti expected result	Pass
TS-004		Title kosong,Deskripsi kosong dan Image diisi	1. Title: null 2. Deskripsi: null	Muncul message the given data was invalid	Sama seperti expected result	Pass

Code	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Pass /Fail
	TS-005		3. Image: menu.PNG			
			1. Title: null 2. Deskripsi: null 3. Image: null	Muncul message the given data was invalid	Sama seperti expected result	Pass
			1. Title: testing 2. Deskripsi: testing 3. Image: null	Muncul message the given data was invalid	Message Data berhasil ditambahkan Data muncul dihalaman konfigurasimobile	Pass
			1. Title: testing 2. Update Title: testing1	Data berhasil diubah	Sama seperti expected result	Pass
			1. Deskripsi: tetsing 2. Update Deskripsi: testingabcd	Data berhasil diubah	Sama seperti expected result	Pass
	TS-009	Update Images	1. Images: menu.PNG 2. Update Images: navigasi.PNG	Data berhasil diubah	Sama seperti expected result	Pass

## 2. Pengujian Web Service

Table Pengujian WS-01 Fungsional Login

Code	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Pass /Fail
TS-001	Cek fungsional login	Username dan password valid	3. Username: faisal27 4. Password: 12345678	Muncul pesan berhasil	Sama seperti expected result	Pass
TS-002		Username tidak valid dan password valid	3. Username: failsa27 4. Password: 12345678	Muncul message username tidak valid/tidak terdaftar	Sama seperti expected result	Pass
TS-003		Username valid dan password tidak valid	3. Username: faisal27 4. Password: barca1899	Muncul message password salah	Sama seperti expected result	Pass
TS-004		Username dan password tidak valid	3. Username: failsa27 4. Password: barca1899	Muncul message Username& password salah	Muncul message Username tidak terdaftar	Pass

Table Pengujian WS-02 Fungsional Customer

Code	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Pass /Fail
TS-001	Cek fungsional Customer	Menambahkan Data Customer	nama : Faisal Username : faisal27 Email : faisal@gmail.com	Data berhasil ditambahkan	Sama seperti Expected Result	Pass

Code	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Pass /Fail
TS-002		Update Data Customer	Password : 12345678 No hp : 085794118495			
			nama : Faisal I A Username : faisal27 Email : faisal@gmail.com Password : 12345678 No hp : 085794118495	Data berhasil di ubah	Sama seperti Expected Result	Pass

Table Pengujian WS-03 Fungsional Pemesanan Tiket

Code	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Pass /Fail
TS-001	Cek fungsional Pemesanan Tiket	Menambahkan Data Pemesanan	Jumlah tiket : 10 Jumlah uang : 130000 Uang pembayaran : 150000	Data berhasil ditambahkan	Sama seperti Expected Result	Pass
		Verifikasi kode pemesanan	QR Code : TRS-12-02-0001	Data berhasil di verifikasi	Sama seperti Expected Result	Pass

Table Pengujian WS-04 Fungsional Pembayaran

Code	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Pass /Fail
TS-001	Cek fungsional Pembayaran	Upload bukti pembayaran	Bukti : testing.PNG	Data berhasil di upload	Sama seperti Expected Result	Pass
		Mengosongkan bukti pembayaran	Bukti :	Muncul pesan field harus diisi	Sama seperti Expected Result	Pass

### 3. Pengujian Aplikasi Mobile

Table Pengujian M-01 Fungsional Login

Code	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Pass /Fail
TS-001	Cek fungsional login	Username dan password valid	5. Username: faisal27 6. Password: 12345678	Masuk ke halaman home	Sama seperti expected result	Pass
TS-002		Username tidak valid dan password valid	5. Username: failsa27 6. Password: 12345678	Muncul message username tidak valid/tidak terdaftar	Sama seperti expected result	Pass
TS-003		Username valid dan password tidak valid	5. Username: faisal27 6. Password: barca1899	Muncul message password salah	Sama seperti expected result	Pass
TS-004		Username dan password tidak valid	5. Username: failsa27 6. Password: barca1899	Muncul message Username& password salah	Muncul message Username tidak terdaftar	Pass

Table Pengujian WS-02 Fungsional Customer

Code	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Pass /Fail
TS-001	Cek fungsional Customer	Menambahkan Data Customer	nama : Faisal Username : faisal27 Email : faisal@gmail.com Password : 12345678 No hp : 085794118495	Data berhasil ditambahkan	Sama seperti Expected Result	Pass
TS-002		Update Data Customer	nama : Faisal I A No hp : 085794118495	Data berhasil diubah	Sama seperti Expected Result	Pass
TS-003		Mengosongkan nama	nama : No hp : 085794118495	Muncul pesan field harus diisi	Sama seperti Expected Result	Pass
TS-004		Mengosongkan nomor hp	nama : Faisal I A No hp :	Muncul pesan field harus diisi	Sama seperti Expected Result	Pass

Table Pengujian M-03 Fungsional Pemesanan Tiket

Code	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Pass /Fail
TS-001	Cek fungsional Pemesanan Tiket	Menambahkan Data Pemesanan	Jumlah tiket : 10 Pembayaran : 130000	Data berhasil ditambahkan	Sama seperti Expected Result	Pass
TS-002		Mengosongkan jumlah tiket	Jumlah tiket : Pembayaran : 130000	Muncul pesan field harus diisi	Sama seperti Expected Result	Pass
TS-003		Menambahkan Data Pemesanan	Jumlah tiket : 10 Pembayaran :	Muncul pesan field harus diisi	Sama seperti Expected Result	Pass
TS-004		Verifikasi kode pemesanan	QR Code : TRS-12-02-0001	Data berhasil diverifikasi	Sama seperti Expected Result	Pass

Table Pengujian M-04 Fungsional Pembayaran

Code	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Pass /Fail
TS-001	Cek fungsional Pembayaran	Upload bukti pembayaran	Bukti : testing.PNG	Data berhasil diupload	Sama seperti Expected Result	Pass
TS-002		Mengosongkan bukti pembayaran	Bukti :	Muncul pesan field harus diisi	Sama seperti Expected Result	Pass