

**MBKM**  
**Studi Independen**



**STUDI INDEPENDEN PEMBUATAN APLIKASI MASKAPAI  
PENERBANGAN BERBASIS WEBSITE DAN ANDROID  
DI PT LENTERA BANGSA BENDERANG**

**LAPORAN KULIAH KERJA PRAKTEK**

**Oleh:**

<b>NIM</b>	<b>NAMA</b>
<b>1911502217</b>	<b>Muhammad Ali Akbar</b>

**FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BUDI LUHUR**

**JAKARTA**

**SEMESTER GASAL**

**2022/2023**



LEMBAR PENGESAHAN



Nama : Muhammad Ali Akbar  
Nomor Induk Mahasiswa : 1911502217  
Program Studi : Teknik Informatika  
Jenjang Studi : Strata 1  
Judul : STUDI INDEPENDEN PEMBUATAN APLIKASI MASKAPAI  
PENERBANGAN BERBASIS WEBSITE DAN ANDROID  
DI PT LENTERA BANGSA BENDERANG

Laporan Kuliah Kerja Praktek ini telah disetujui, disahkan dan direkam secara elektronik sehingga tidak memerlukan tanda tangan tim penguji.

Tim Penguji:

Penguji : Ikhsan Rahdiana, S.Kom., M.Kom.  
Pembimbing : Ikhsan Rahdiana, S.Kom., M.Kom.  
Ketua Program Studi : Dr. Indra, S.Kom., M.T.I

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah Swt atas segala rahmat, hikmah dan karunia-Nya sehingga penulis dapat menyelesaikan Laporan Kuliah Kerja Praktek (KKP) dengan tepat waktu. Tidak lupa salawat dan salam selalu terucap kepada Nabi Muhammad saw. Adapun judul yang penulis pilih untuk laporan Kuliah Kerja Praktek (KKP) ini adalah **“Studi Independen Pembuatan Aplikasi Maskapai Penerbangan Berbasis Website dan Android Di PT Lentera Bangsa Banderang”**.

Dalam pembuatan laporan ini, tentu tidak bisa lepas dari pengarahan dan bimbingan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada semua pihak yang telah membantu penulis dalam pengerjaan laporan ini. Adapun pihak yang terkait diantaranya sebagai berikut:

1. Allah Swt atas segala nikmat dan kemudahan-Nya sehingga penulis berhasil menyelesaikan Laporan Kuliah Kerja Praktek (KKP).
2. Orang tua dan keluarga di rumah yang selalu memberikan doa, kasih sayang, serta perhatiannya kepada penulis.
3. Bapak Ir. Wendi Usino, MM., M.Sc., Ph.D selaku Rektor Universitas Budi Luhur.
4. Bapak Dr. Deni Mahdiana, S.Kom., M.M., M.Kom. selaku Dekan Fakultas Teknologi Informasi Universitas Budi Luhur.
5. Bapak Dr. Indra, S.Kom., M.T.I selaku Ketua Program Studi Teknik Informatika Universitas Budi Luhur.
6. Bapak Ikhsan Rahdiana S.Kom., M.Kom. selaku dosen pembimbing KKP.
7. Semua pihak yang terkait, baik secara langsung maupun tidak langsung. Termasuk seluruh dosen Universitas Budi Luhur, Fasiliator tempat penulis magang dan teman-teman.

Jakarta, 25 Januari 2022

Penulis

## DAFTAR TABEL

Tabel 3. 1 Tabel Jadwal Kegiatan Final Project .....	9
Tabel 3. 2 Tabel Task Responsibility .....	11
Tabel 4. 1 Tabel Aturan Bisnis GarudaNih .....	16
Tabel 4. 2 Tabel Notifies.....	23
Tabel 4. 3 Tabel Ticket .....	25
Tabel 4. 4 Tabel User .....	27
Tabel 4. 5 Tabel Transaction.....	28

## DAFTAR GAMBAR

Gambar 2. 1 Struktur Organisasi .....	4
Gambar 2. 2 Lokasi Google Maps Binar Academy .....	4
Gambar 3. 1 Diagram Alur Pengerjaan Full-Stack Web.....	iii
Gambar 3. 2 Timeline pengerjaan final .....	7
Gambar 4. 1 Activity Diagram Proses Bisnis GarudaNih.....	12
Gambar 4. 2 Bagan Aturan Bisnis GarudaNih.....	14
Gambar 4. 3 Use Case Superadmin GarudaNih.....	17
Gambar 4. 4 Use Case User GarudaNih .....	18
Gambar 4. 5 Class Diagram .....	20
Gambar 4. 6 LRS (Logical Record Structure) .....	21
Gambar 4. 7 Struktur Tampilan (menu) Website.....	28
Gambar 4. 8 Struktur Tampilan (menu) Mobile .....	29
Gambar 4. 9 Struktur Tampilan (menu) Admin.....	29
Gambar 4. 10 Rancangan Layar Login - Website.....	30
Gambar 4. 11 Rancangan Layar Register - Website.....	31
Gambar 4. 12 Rancangan Layar Forgot Password - Website.....	32
Gambar 4. 13 Rancangan layar Dashboard - Website .....	33
Gambar 4. 14 Rancangan Layar Edit Profile - Website.....	34
Gambar 4. 15 Rancangan Layar About - Website .....	35
Gambar 4. 16 Rancangan Layar Notification - Website .....	36
Gambar 4. 17 Rancangan Layar Wishlist - Website.....	37
Gambar 4. 18 Rancangan Layar History - Website .....	38
Gambar 4. 19 Rancangan Layar Find Ticket - Website.....	39
Gambar 4. 20 Rancangan Layar Booking Ticket - Website.....	40
Gambar 4. 21 Rancangan Layar Login - Android .....	41
Gambar 4. 22 Rancangan Layar Buat Akun - Android.....	42
Gambar 4. 23 Rancangan layar Ubah Password - Android.....	43
Gambar 4. 24 Rancangan Layar Profile - Android .....	44
Gambar 4. 25 Rancangan Layar Home - Android .....	45
Gambar 4. 26 Rancangan Layar Riwayat Order - Android.....	46
Gambar 4. 27 Rancangan Layar Order - Android.....	47
Gambar 4. 28 Rancangan Layar Detail Ticket - Android .....	48
Gambar 4. 29 Rancangan Layar Order Detail - Android .....	49
Gambar 4. 30 Rancangan Layar Status Booking - Android.....	50
Gambar 4. 31 Rancangan Layar Dashboard - Admin .....	51
Gambar 4. 32 Rancangan Layar Profile - Admin .....	52
Gambar 4. 33 Rancangan Layar Our Users - Admin .....	53
Gambar 4. 34 Rancangan Layar Create New Admin - Admin .....	54

Gambar 4. 35 Rancangan Layar List Tickets - Admin .....	55
Gambar 4. 36 Rancangan Layar Create New Ticket - Admin .....	56
Gambar 4. 37 Rancangan Layar Transactions - Admin.....	56
Gambar 4. 38 Sequence Diagram Login & Register - User.....	57
Gambar 4. 39 Rancangan Layar Sequence Diagram Find Ticket - User.....	58
Gambar 4. 40 Sequence Diagram Login - Admin .....	58
Gambar 4. 41 Boundary Class User .....	59
Gambar 4. 42 Boundary Class User .....	59
Gambar 4. 43 Boundary Class Admin.....	59
Gambar 4. 44 Control Class GarudaNih.....	59
Gambar 4. 45 Entity Class GarudaNih .....	60

## DAFTAR ISI

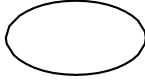
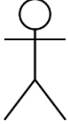

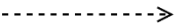
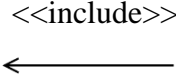
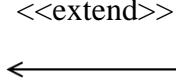
LEMBAR PENGESAHAN .....	i
KATA PENGANTAR .....	ii
DAFTAR TABEL.....	iii
DAFTAR GAMBAR.....	iv
DAFTAR ISI.....	vi
DAFTAR SIMBOL .....	ix
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang .....	1
1.2 Tujuan Pelaksanaan.....	2
1.3 Manfaat Program.....	2
BAB II GAMBARAN UMUM .....	3
2.1 Profil Perusahaan .....	3
2.2 Struktur Organisasi.....	4
2.3 Lokasi Tempat Mitra.....	4
BAB III METODE PELAKSANAAN .....	5
3.1 Tahapan Pelaksanaan .....	5
3.2 Pihak yang terlibat.....	9
BAB IV HASIL YANG DICAPAI .....	12
4.1 Proses Bisnis .....	12
4.1.1 <i>Activity Diagram</i> Proses Bisnis GarudaNih .....	12
4.1.2 Proses Bisnis GarudaNih.....	13
4.2 Aturan Bisnis.....	14
4.3 <i>Use Case Diagram dan Deskripsi Use Case</i> .....	16
4.4 Rancangan Basis Data.....	19
4.4.1 Class Diagram.....	20
4.4.2 LRS ( <i>Logical Record Structure</i> ).....	21
4.4.3 <i>Model Data Relational</i> .....	22
4.4.4 Spesifikasi Basisdata.....	22
4.5 Struktur Tampilan .....	28
4.6 Rancangan Layar (User Interface) .....	30

4.6.1	Rancangan Layar Login – Website.....	30
4.6.2	Rancangan Layar Register – Website .....	31
4.6.3	Rancangan Layar Lupa Password – Website.....	32
4.6.4	Rancangan Layar Dashboard – Website.....	33
4.6.5	Rancangan Layar Profile.Edit Profile – Website.....	34
4.6.6	Rancangan Layar Profile.About – Website .....	35
4.6.7	Rancangan Layar Profile.Notification – Website .....	36
4.6.8	Rancangan Layar Profile.Wishlist – website .....	37
4.6.9	Rancangan Layar History – Website .....	38
4.6.10	Rancangan Layar Ticket – Website .....	39
4.6.11	Rancangan Layar Booking Ticket – Website .....	40
4.6.12	Rancangan Layar Login – Android .....	41
4.6.13	Rancangan Layar Buat Akun – Android.....	42
4.6.14	Rancangan Layar Ubah Password – Android.....	43
4.6.15	Rancangan Layar Profile – Android .....	44
4.6.16	Rancangan Layar Home – Android .....	45
4.6.17	Rancangan Layar Riwayat Order – Android .....	46
4.6.18	Rancangan Layar Order – Android.....	47
4.6.19	Rancangan Layar Detail Ticket – Android.....	48
4.6.20	Rancangan Layar Order Detail – Android.....	49
4.6.21	Rancangan Layar Status Booking – Android .....	50
4.6.22	Rancangan Layar Dashboard – Admin.....	51
4.6.23	Rancangan Layar Profile – Admin .....	52
4.6.24	Rancangan Layar Our Users – Admin.....	53
4.6.25	Rancangan Layar Create New Admin – Admin .....	54
4.6.26	Rancangan Layar List Tickets – Admin .....	55
4.6.27	Rancangan Layar Create New Ticket – Admin .....	56
4.6.28	Rancangan Layar Transactions – Admin.....	56
4.7	Sequence Diagram .....	57
4.8	Class Diagram .....	59
BAB V PENUTUP .....		61
5.1	Kesimpulan .....	61


5.2	Saran.....	61
BAB VI REFLEKSI DIRI .....		62
DAFTAR PUSTAKA .....		63



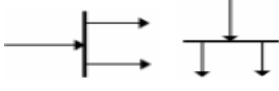

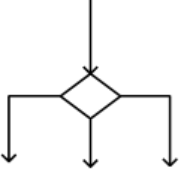
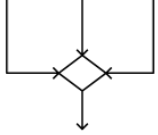
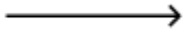
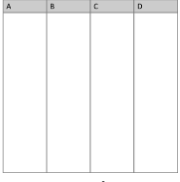
## DAFTAR SIMBOL

### 1. Simbol Use Case Diagram

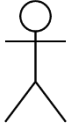
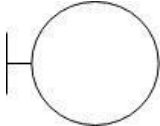

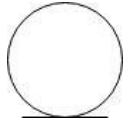


No	Gambar	Keterangan
1	 <i>Use Case</i>	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
2	 <i>Actor</i>	Aktor: Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
3	 <i>Association</i>	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
4	 <i>Generalisasi</i>	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
5	 <<include>>	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
6	 <<extend>>	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

### 2. Simbol Activity Diagram

No	Gambar	Keterangan
1	 <i>Start state</i>	<i>Start State</i> digunakan sebagai penanda tindakan awal kegiatan untuk setiap diagram aktivitas. Dengan tegas menunjukkan di mulainya suatu <i>workflow</i> pada sebuah <i>activity diagram</i> , Hanya ada satu Start State dalam sebuah <i>workflow</i>

2	 <p style="text-align: center;">End Point</p>	<p><i>End Point</i> merupakan penanda bagian akhir dari aktivitas. Bisa terdapat lebih dari satu <i>end point</i> pada sebuah <i>activity diagram</i></p>
3	 <p style="text-align: center;">Activity</p>	<p><i>Activity</i> merupakan aktivitas yang dilakukan atau yang sedang terjadi di dalam <i>activity diagram</i>. <i>Activity</i> menggambarkan sebuah pekerjaan/tugas dalam <i>workflow</i></p>
4	 <p style="text-align: center;">Fork</p>	<p><i>Fork</i> (percabangan) digunakan untuk memecah <i>behaviour</i> menjadi <i>activity</i> atau <i>action parallel</i>. Sebuah transisi dapat membuat cabang ke dua atau lebih percabangan. Mempunyai 1 transisi masuk dan 2 atau lebih transisi keluar</p>
5	 <p style="text-align: center;">Join</p>	<p><i>Join</i> (penggabungan) yang digunakan untuk menggabungkan atau mempersatukan kembali <i>activity</i> atau <i>action</i> yang paralel tersebut. Mempunyai 2 atau lebih transisi masuk dan hanya 1 transisi keluar</p>
6	 <p style="text-align: center;">Decision Node</p>	<p><i>Decision Node</i> adalah suatu titik point pada <i>activity diagram</i> yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi, merupakan suatu titik (<i>point</i>) di dalam <i>activity diagram</i> aktivitas, jika lebih dari 1, dengan mengindikasikan adanya kemungkinan perbedaan transisi</p>
7	 <p style="text-align: center;">Merge Event</p>	<p><i>Merge event</i> digunakan untuk menggabungkan <i>flow</i>, yang tadinya dipecah oleh <i>decision</i> atau keputusan. <i>Merge Event</i> hanya ada 1 alur keluar namun bisa mempunyai &gt; 1 alur masuk</p>
8	 <p style="text-align: center;">Action Flow</p>	<p><i>Action flow</i> digunakan sebagai transisi atau penghubung dari tindakan yang satu ke tindakan yang lain, menunjukkan kegiatan apa berikutnya setelah suatu kegiatan sebelumnya.</p>
9	 <p style="text-align: center;">Swimlane</p>	<p><i>Swimlane</i> dipergunakan untuk memecah <i>activity diagram</i> menjadi beberapa baris atau kolom, yang fungsinya untuk membagi tanggung jawab atau tugas antar objek yang sedang melakukan aktivitas tertentu. <i>Swimlane</i> digambar secara <i>vertical</i>, walaupun kadang-kadang digambar secara <i>horizontal</i></p>

### 3. Simbol Activity Diagram

Gambar	Keterangan
	<p><b>Aktor</b></p> <p>Menggambarkan seseorang yang menyediakan atau menerima informasi.</p>
	<p><b>Boundary</b></p> <p>Menggambarkan hubungan <i>user</i> dengan sistem.</p>
	<p><b>Control</b></p> <p>Digunakan untuk mengontrol aktivitas yang dilakukan dalam suatu kegiatan.</p>
	<p><b>Entity</b></p> <p>Memiliki atribut yang memiliki data yang bisa direkam.</p>
	<p><b>Message</b></p> <p>Digambarkan dengan garis berpanah yang menunjukkan arah <i>message</i>.</p>
	<p><b>Lifeline</b></p> <p>Menggambarkan eksekusi obyek selama <i>sequence</i>.</p>

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan teknologi sangat pesat di semua sektor. Hal ini bisa terjadi karena manusia sebagai aktor utamanya. Tingginya tingkat ketergantungan manusia terhadap teknologi dan meningkatnya persaingan antar perusahaan teknologi menjadi salah satu sebab utama mengapa teknologi semaju ini [1]. Kemajuan inilah yang kemudian dimanfaatkan beberapa perusahaan untuk membuat sebuah layanan aplikasi perjalanan, dimana para pengguna dapat memesan tiket tanpa keluar rumah dengan hanya bermodalkan *internet*.

Di era *post-pandemic* seperti ini, orang-orang mulai disibukkan kembali kepada rutinitas awal seperti bekerja dan bersosialisasi. Begitupun dengan melakukan perjalanan dan liburan. Untuk itulah dikembangkan aplikasi maskapai penerbangan berbasis *web* dan *android* untuk memudahkan para masyarakat yang ingin berlibur, melepaskan penat dan stress setelah mengalami pandemic. Selain itu, aplikasi ini dikembangkan sebagai syarat kelulusan kegiatan Studi Independen Kampus Merdeka.

Dengan adanya kegiatan Studi Independen Kampus Merdeka, penulis mendapatkan kesempatan untuk mengikuti Studi Independen di salah satu mitra Kampus Merdeka yaitu Binar Academy selama 5 bulan untuk meningkatkan keterampilan Hard Skill dan Soft Skill. Dalam *final project* ini, penulis diharuskan membuat sebuah aplikasi maskapai penerbangan yang berbasis web dan mobile yang diberi nama GarudaNih. Untuk pengembangan versi *web*, aplikasi ini dibangun dengan 4 teknologi utama, yaitu: *HTML*, *CSS*, *JavaScript*, dan *PostgreSql*. Kemudian ditambahkan beberapa *framework* dan *library* seperti: *Bootstrap*, *ExpressJs* dan *ReactJs* atau yang biasa disebut dengan *PERN Stack* [2]. Sedangkan untuk pengembangan versi *android*, teknologi yang dipakai adalah *Native Android* dengan memakai bahasa pemrograman *Kotlin* [3].

## **1.2 Tujuan Pelaksanaan**

Tujuan dari pelaksanaan project ini adalah untuk membuat aplikasi maskapai penerbangan berbasis *web* dan *android*. Agar user yang ingin berlibur bisa dengan mudahnya mendapatkan tiket penerbangan sesuai kota atau negara tujuan. Aplikasi yang kami buat ini sangat memikirkan kemudahan penggunaan. Mulai dari desain, alur aplikasi yang mudah dan jelas sampai ke aplikasi yang interaktif, harapannya agar user benar-benar bisa merasakan kenyamanan selama menggunakan aplikasi dan merasakan betapa mudahnya memesan sebuah tiket perjalanan hanya dengan beberapa kali klik.

## **1.3 Manfaat Program**

Berikut manfaat program magang yang meliputi manfaat untuk Universitas, manfaat untuk mitra Magang, manfaat untuk Prodi, dan manfaat untuk mahasiswa pelaksana program:

1. Bagi Universitas, menghasilkan lulusan yang berkompetensi tinggi sesuai dengan standar industri. Universitas mendapatkan Feedback yang dilakukan mahasiswa dan mitra magang, sehingga Universitas dapat terus melakukan penyempurnaan kurikulum dan pembelajaran agar relevan dan sesuai dengan standar industri.
2. Bagi mitra Magang, meningkatkan citra perusahaan dan mendapatkan ide serta pandangan baru dari para peserta, yang kemudian ide tersebut dapat diaplikasikan untuk peserta baru di kegiatan magang berikutnya.
3. Bagi Prodi, meningkatkan kompetensi global dengan HardSkill dan SoftSkill yang didapatkan di mitra magang, serta menjadi bahan pertimbangan dalam mempersiapkan mahasiswa di dunia kerja.
4. Bagi Mahasiswa, meningkatkan kualitas diri mahasiswa dalam rangka mempersiapkan dunia kerja dan menerapkan ilmu-ilmu yang didapatkan dari mitra magang.

## **BAB II**

### **GAMBARAN UMUM**

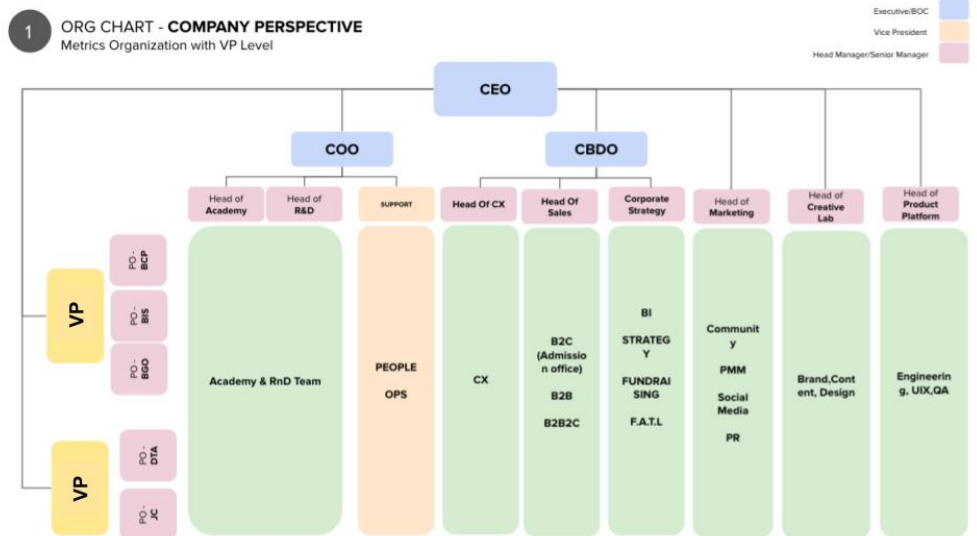
#### **2.1 Profil Perusahaan**

Binar Academy adalah sebuah perusahaan *edtech* yang berfokus pada pengembangan skill dan talenta digital dengan meningkatkan pengalaman belajar. Diajarkan oleh mentor berpengalaman, Binar Academy berfokus pada pemberdayaan calon talenta digital di seluruh Indonesia untuk mewujudkan potensi penuh mereka dalam hal pengetahuan teknis dan keterampilan kepemimpinan melalui kolaborasi dan pembelajaran berkelanjutan.

Binar Academy memiliki beberapa produk pembelajaran yaitu binar *insight* yang dapat dijadikan sebagai referensi untuk menambah skill dibidang digital, dan diharapkan lulusan dari binar *insight* mampu mengimplementasikan dan menjalin koneksi dengan teman-teman industri. Yang kedua ada binar bootcamp, suatu produk pembelajaran dengan kurikulumnya dirancang bersama fasilitator ternama yang sudah berpengalaman dan bekerja di industri digital. Binar juga telah berpartner dengan beberapa kampus ternama dan perusahaan terkenal di Indonesia. Dampaknya, Binar dapat menyalurkan para lulusan untuk langsung bekerja.

Sebagai perusahaan *edtech*, Binar Academy juga ikut berpartisipasi dalam program yang dikeluarkan oleh KEMENDIKBUD yaitu Kampus Merdeka yang bertujuan untuk membangun generasi Indonesia agar memiliki persiapan yang matang sebelum terjun ke dunia kerja.

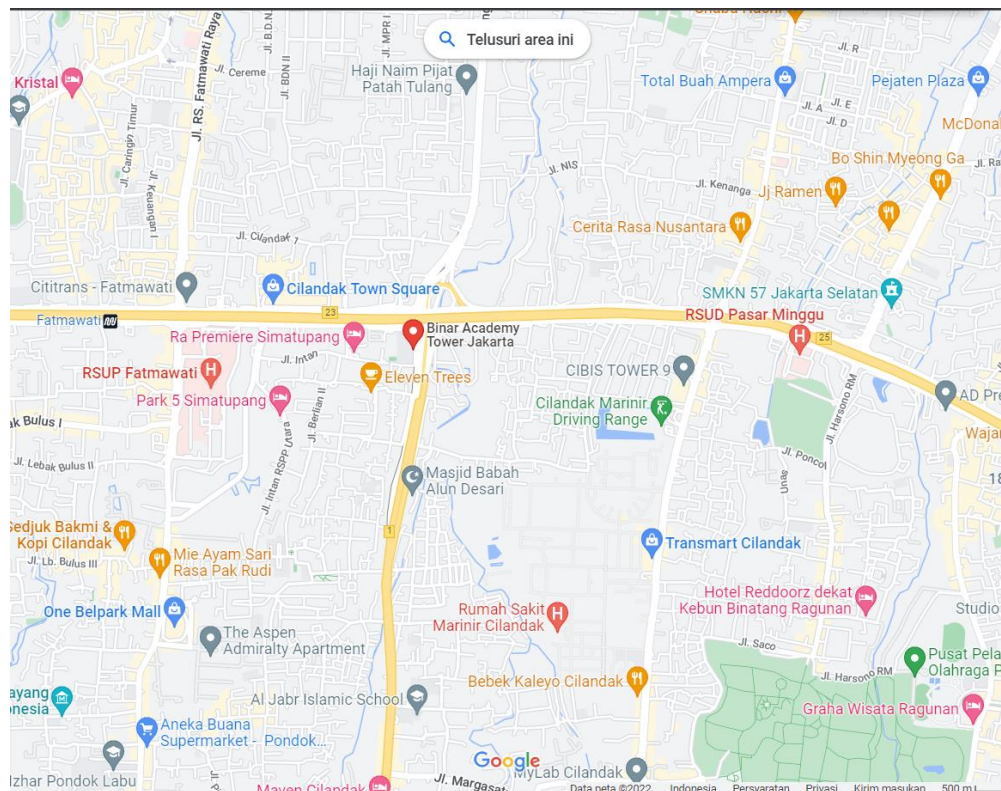
## 2.2 Struktur Organisasi



Gambar 2. 1 Struktur Organisasi

## 2.3 Lokasi Tempat Mitra

Binar Academy bertempat di The Breeze (Batik 3 - L.30, Jl. BSD Grand Boulevard, BSD City, Kec. Cisauk, Kabupaten Tangerang, Banten 15345.



Gambar 2. 2 Lokasi Google Maps Binar Academy

## BAB III

### METODE PELAKSANAAN

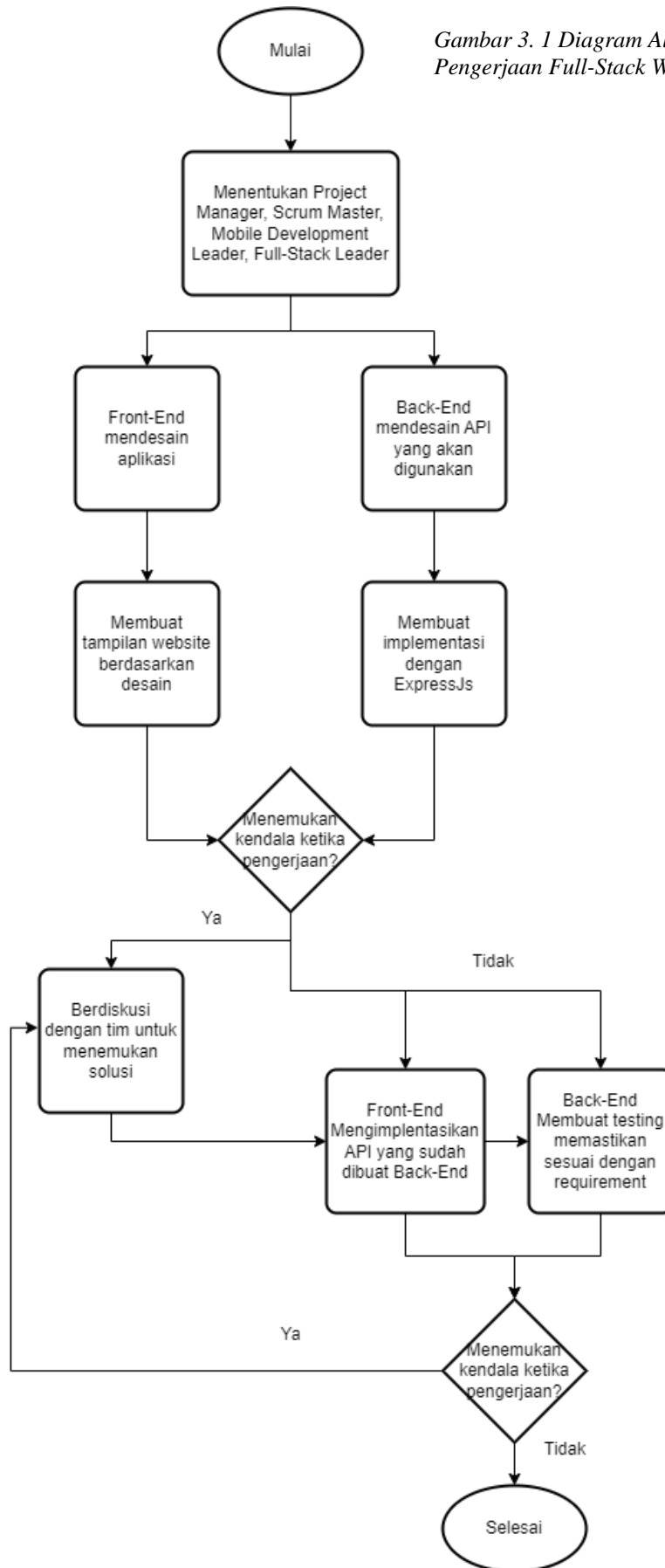
#### 3.1 Tahapan Pelaksanaan

*Project* atau aplikasi GarudaNih merupakan *final project* yang dibuat sebagai syarat kelulusan Studi Independen Binar Academy. Untuk mengerjakan *project* ini, dibentuklah sebuah tim yang terdiri dari *Full-stack Web Developer* dan *Mobile Development*. *Full-stack* dibagi lagi menjadi dua bagian, *Front-End* dan *Back-End* yang dibangun menggunakan ekosistem *JavaScript* [4].

Pekerjaan *Front-End* terkait mendesain *mock-up* aplikasi, membuat desain pada web, memastikan *User* mendapatkan pengalaman terbaik ketika menggunakan aplikasi. Pekerjaan *Back-End* terkait membuat *API* untuk dikonsumsi *Front-End* dan *Mobile*, mendesain dan membuat *database* dan melakukan testing untuk memastikan hal yang dikerjakan sesuai dengan *requirement* [5]. Pekerjaan *Mobile* terkait mengonversi apa yang ada di *web* menjadi *android*.

Tahapan atau proses pelaksanaan *final project* ini dibagi menjadi tiga tahapan besar, yaitu *pre-production*, *production*, *post-production*. Berikut diagram alur pengerjaan *project GarudaNih* beserta *timeline* pengerjaan:

Gambar 3. 1 Diagram Alur Pengerjaan Full-Stack Web





Waktu	Kegiatan
<b>Sprint Satu</b>	
21 – 24 Nov	<ul style="list-style-type: none"> <li>• Melakukan diskusi awal diskusi awal final project.</li> <li>• Self learning, mempersiapkan diri.</li> <li>• Melanjutkan diskusi final project, menentukan role.</li> <li>• Self learning, mempersiapkan perancangan database.</li> </ul>
25 Nov – 4 Des	<ul style="list-style-type: none"> <li>• Melakukan rancangan database.</li> <li>• Melanjutkan diskusi, menentukan alur aplikasi.</li> <li>• Self learning.</li> <li>• Melanjutkan menentukan alur.</li> <li>• Self learning, mengerjakan tugas yang diberi <i>Project Owner</i>.</li> <li>• Mendesain endpoint sesuai yang disepakati.</li> </ul>
<b>Sprint Dua &amp; Sprint Tiga</b>	
5 – 7 Des	<ul style="list-style-type: none"> <li>• Diskusi dengan kelompok, melakukan tracking progres pembuatan aplikasi.</li> <li>• Melanjutkan mendesain API.</li> <li>• Melakukan diskusi dengan kelompok terkait bug yang dialami.</li> </ul>
8 – 19 Des	<ul style="list-style-type: none"> <li>• Melanjutkan mendesain API.</li> <li>• Diskusi dengan teman sekelompok terkait progres dan masalah yang dialami saat mendesain API.</li> <li>• Melakukan tracking progres dan berdiskusi error.</li> <li>• Melanjutkan membuat api.</li> <li>• Melakukan diskusi terkait <i>MVP</i> final project.</li> <li>• Menyelesaikan fitur agar sesuai dengan <i>MVP</i>.</li> </ul>

	<ul style="list-style-type: none"> <li>• Membuat testing untuk API.</li> <li>• Berdiskusi dan melakukan tracking progres.</li> </ul>
20 Des – 28 Des	<ul style="list-style-type: none"> <li>• Melakukan diskusi dengan bagian <i>front-end</i> terkait penggabungan.</li> <li>• Melanjutkan membuat testing untuk API.</li> <li>• Melakukan penggabungan antara <i>front-end</i> dengan <i>back-end</i>.</li> <li>• Diskusi terkait bug yang dialami ketika penggabungan.</li> <li>• Mencoba aplikasi untuk melihat apakah ada bug yang dialami ketika penggabungan.</li> <li>• Deployment menggunakan <i>railway</i>.</li> <li>• Testing API, testing <i>app</i>.</li> </ul>
30 – 31 Des	<ul style="list-style-type: none"> <li>• Melakukan testing &amp; deployment melalui <i>railway</i></li> </ul>

Tabel 3. 1 Tabel Jadwal Kegiatan Final Project

### 3.2 Pihak yang terlibat

Pengerjaan *final project* bernama *GarudaNih*, tentunya melibatkan beberapa rekan tim yang terdiri dari *Full-Stack Web Development* dan *Mobile Development*. Berikut adalah daftar pihak-pihak yang terlibat beserta tanggung jawab dari pihak-pihak tersebut:

Nama	Project Role	Task Responsibility
<i>Android Development</i>		
Farrel Ibrahim	<i>Project Owner</i> , bekerja di bagian Mobile Development.	<ol style="list-style-type: none"> <li>1. Slicing.</li> <li>2. Login Register.</li> <li>3. CRUD Admin.</li> <li>4. Transaksi, Order Tiket.</li> <li>5. Update Profile.</li> <li>6. History Tiket.</li> <li>7. Notification.</li> <li>8. Lint.</li> </ol>

Rifqi Padi Siliwangi	Leader dari Mobile Development.	<ol style="list-style-type: none"> <li>1. Slicing.</li> <li>2. Testing.</li> <li>3. Update Profile.</li> <li>4. Wishlist.</li> <li>5. Lint.</li> <li>6. Login Register.</li> </ol>
Willy Ramadani	Bekerja di bagian Mobile Development	<ol style="list-style-type: none"> <li>1. Slicing.</li> <li>2. History Tiket.</li> <li>3. Login Register.</li> <li>4. Localization.</li> <li>5. Notification.</li> <li>6. Bug Fixing.</li> </ol>
<i>Full-Stack Web Development Back-End</i>		
Yusron Arly Bazarah	Scrum Master dan Leader dari Full-Stack Web Development, bekerja di bagian Back-End.	<ol style="list-style-type: none"> <li>1. Design System.</li> <li>2. Development.</li> <li>3. Bug Fixing.</li> <li>4. Slicing.</li> <li>5. Notification.</li> <li>6. Wishlist.</li> </ol>
Muhammad Zerq Elvonda Graha Yuda	Sebagai anggota dari Full-Stack Web Development, bekerja di bagian Back-End.	<ol style="list-style-type: none"> <li>1. Development.</li> <li>2. Bug Fixing.</li> <li>3. Unit Testing.</li> <li>4. Oauth Google.</li> </ol>
Muhammad Ali Akbar	Sebagai anggota dari Full-Stack Web Development, bekerja di bagian Back-End.	<ol style="list-style-type: none"> <li>1. Development.</li> <li>2. Bug Fixing.</li> <li>3. Unit Testing.</li> <li>4. Login Register.</li> </ol>
<i>Full-Stack Web Development Front-End</i>		
Annisa Kayla Azzira M	Sebagai anggota dari Full-Stack Web Development, bekerja di bagian Front-End.	<ol style="list-style-type: none"> <li>1. Slicing.</li> <li>2. Integration.</li> <li>3. Bug Fixing.</li> <li>4. Mock-up.</li> </ol>
Riandi Ahmad	Sebagai anggota dari Full-Stack Web Development, bekerja di bagian Front-End.	<ol style="list-style-type: none"> <li>1. Slicing.</li> <li>2. Integration.</li> <li>3. Bug Fixing.</li> <li>4. Mock-up.</li> </ol>

Arya Affanda	Sebagai anggota dari Full-Stack Web Development, bekerja di bagian Front-End.	<ol style="list-style-type: none"><li>1. Slicing.</li><li>2. Integration.</li><li>3. Bug Fixing.</li><li>4. Testing.</li></ol>
--------------	---	--

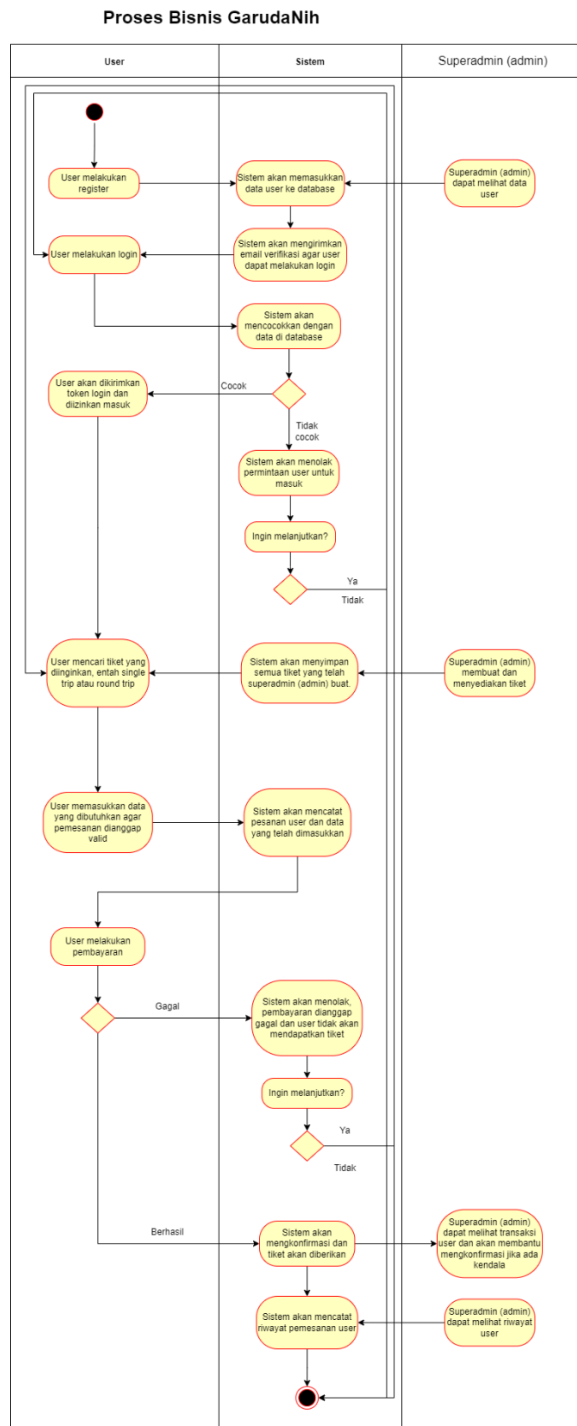
*Tabel 3. 2 Tabel Task Responsibility*

# BAB IV

## HASIL YANG DICAPAI

### 4.1 Proses Bisnis

#### 4.1.1 Activity Diagram Proses Bisnis GarudaNih



*Gambar 4. 6 Activity Diagram Proses Bisnis GarudaNih*

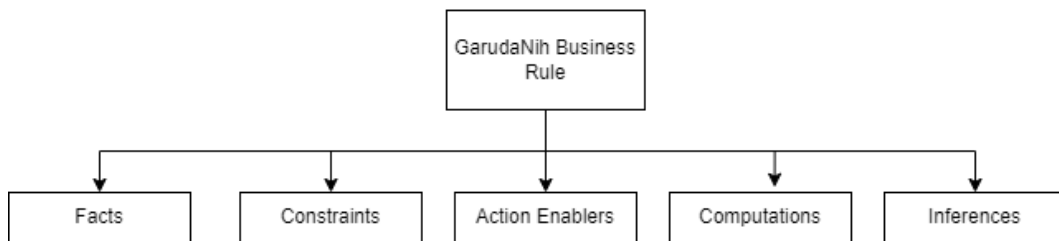
#### **4.1.2 Proses Bisnis GarudaNih**

Proses bisnis adalah serangkaian langkah yang saling terkait yang ditugaskan kepada setiap pemangku kepentingan untuk pekerjaan tertentu untuk memberikan produk atau layanan kepada pelanggan [6]. GarudaNih merupakan aplikasi maskapai penerbangan yang dibuat agar memudahkan masyarakat dalam melakukan perjalanan dengan pesawat. Proses bisnis GarudaNih dimulai dengan user melakukan registrasi pada website. Setelah itu sistem akan mencatat data yang di-inputkan user. Meliputi email, password, tanggal akun dibuat. Superadmin (admin) dapat melihat semua data yang dimasukkan user. Setelah itu jika user ingin melakukan pemesanan tiket, user melakukan login. Sebelum user diizinkan masuk ke halaman dashboard, sistem akan mencocokkan data login dengan data pada database. Jika cocok, maka user akan diizinkan masuk ke halaman dashboard. Jika tidak, maka user akan menerima pesan bahwa data yang dimasukkan tidak ditemukan. Superadmin menyediakan semua tiket perjalanan, lalu sistem mencatat semua tiket yang sudah disediakan oleh superadmin (admin). Jika sudah masuk ke dalam dashboard, user bisa langsung mencari tiket yang diinginkan. Terdapat menu ticket pada bagian atas untuk mencari tiket yang diinginkan user. Cukup tentukan keberangkatan dan tujuan, tanggal keberangkatan, single trip atau round trip dan terakhir pilih kelas dan harga. Jika sudah dapat tiket penerbangan yang diinginkan, maka selanjutnya masukkan beberapa data sebagai validasi. Data yang dibutuhkan seperti identify number (NIK, Passport, Visa), nama pembeli dan terakhir masukkan nomor kursi penerbangan. Jika yang dipilih round trip, maka tentukan juga kursi untuk melakukan penerbangan pulang. Sistem akan mencatat pesanan user dan data yang telah dimasukkan. Selanjutnya jika user melakukan pembayaran, maka sistem akan mengkonfirmasi dan tiket akan diberikan. Jika tidak maka sistem akan menolak dan pembayaran dianggap gagal. Sistem akan mencatat semua riwayat pemesanan user.

## 4.2 Aturan Bisnis

Aturan bisnis merupakan salah satu sumber utama kebutuhan fungsional perangkat lunak. Hal ini dikarenakan aturan bisnis bisa mendikte/memaksa sistem untuk memiliki kemampuan/fungsi/fitur tertentu yang juga digunakan untuk mengkonfirmasi aturan-aturan itu sendiri [7]. Aturan bisnis yang diusung GarudaNih sebagai aplikasi diklasifikasikan menjadi 5 bagian, yaitu *facts*, *constraint*, *action enablers*, *computations*, *inferences*.

### Aturan Bisnis GarudaNih



Gambar 4. 7 Bagan Aturan Bisnis GarudaNih

Tipe	Aturan
Facts	<ul style="list-style-type: none"> <li>• setiap user memiliki id</li> <li>• setiap user dapat melihat riwayat transaksi, membuat wishlist, memesan tiket (penerbangan).</li> <li>• setiap user dapat melakukan login dengan akun google atau dengan halaman register.</li> <li>• setiap user dapat melihat riwayat transaksi ascending maupun descending.</li> <li>• setiap admin dapat membuat, menghapus, dan mengee.</li> <li>• setiap admin dapat melihat semua profile user yang sudah mendaftar, tidak termasuk password.</li> <li>• setiap admin dapat melihat semua riwayat transaksi user.</li> <li>• setiap admin dapat melakukan penghapusan user.</li> </ul>

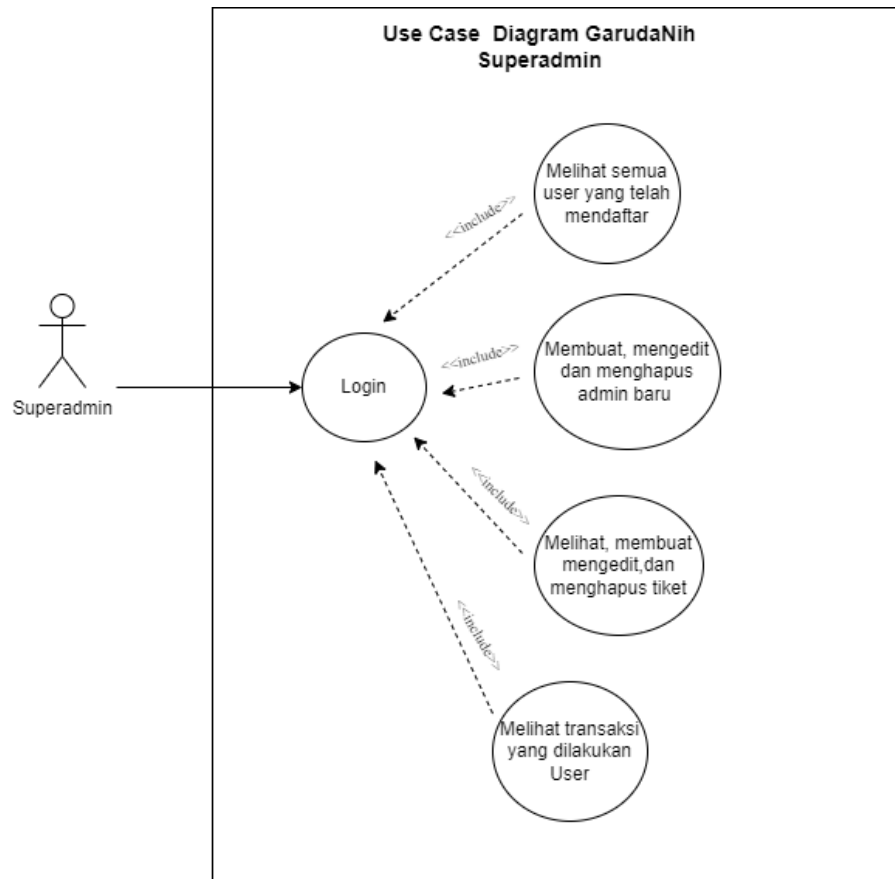
	<ul style="list-style-type: none"> <li>• setiap admin dapat melihat user, riwayat transaksi user ascending maupun descending.</li> <li>• setiap admin dapat melakukan pengeditan profile user, tidak termasuk password.</li> <li>• setiap melakukan pemesanan round trip, maka riwayat akan ditambahkan dua tiket.</li> </ul>
Constraints	<ul style="list-style-type: none"> <li>• hanya user yang sudah melakukan verifikasi yang dapat melakukan login dan memesan tiket.</li> <li>• hanya ada satu superadmin.</li> <li>• hanya superadmin yang dapat membuat admin baru.</li> <li>• Ukuran file attachment untuk profile picture hanya boleh 2mb.</li> <li>• user hanya bisa memilih tempat duduk yang belum dipilih.</li> </ul>
Action Enablers	<ul style="list-style-type: none"> <li>• jika user mendaftarkan email yang sama, maka keluar pesan "this email already exist".</li> <li>• jika kolom email tidak memuat simbol "@" pada saat register atau login, maka keluar pesan "please include an @".</li> <li>• jika kolom password dan confirm password pada halaman register tidak sama, maka keluar pesan "Password and Confirm Password does not match".</li> <li>• jika kolom email dan password kosong pada saat register atau login, maka keluar pesan "Email or password cannot empty".</li> <li>• jika saat login user memasukkan email atau password yang salah, maka akan keluar pesan "User not found".</li> <li>• jika nomor tempat duduk sudah dipilih maka akan diberi warna hijau dan tidak dapat di klik.</li> </ul>

	<ul style="list-style-type: none"> <li>• jika file attachment untuk profile picture bukan format gambar, maka keluar pesan "Format file must be PNG, JPG, or JPEG".</li> <li>• jika file attachment yang diunggah lebih besar dari 2 mb, maka keluar pesan "File size is bigger than 2 MB"</li> </ul>
Computations	<ul style="list-style-type: none"> <li>• id user = user + 1</li> <li>• ticketcode = flight type + year + "TO" + departure date + "AR" + 0 arrive date + total ticket</li> <li>• transaction code = 0 + userid + ticketstring + departure code + destination code + transaction list</li> </ul>
Inferences	<ul style="list-style-type: none"> <li>• jika user melakukan pembayaran, maka pesanan sukses dipesan.</li> <li>• jika user tidak melakukan pembayaran, maka pesanan gagal dipesan.</li> <li>• jika user tidak mengisi nama pada saat register, register dianggap valid.</li> <li>• jika user tidak mengisi name, birth, city, phone, dan profile picture, maka akun tetap dianggap valid.</li> <li>• jika user register dengan google, maka email, profile picture dan nama akan otomatis terisi.</li> </ul>

Tabel 4. 1 Tabel Aturan Bisnis GarudaNih

### 4.3 Use Case Diagram dan Deskripsi Use Case

Use Case diagram adalah satu dari berbagai jenis diagram UML (Unified Modelling Language) yang menggambarkan hubungan interaksi antara sistem dan aktor [8]. Dan berikut adalah Use Case diagram dari aplikasi GarudaNih beserta deskripsi dari Use Case yang terdiri dari bagian *Admin* dan *User*.

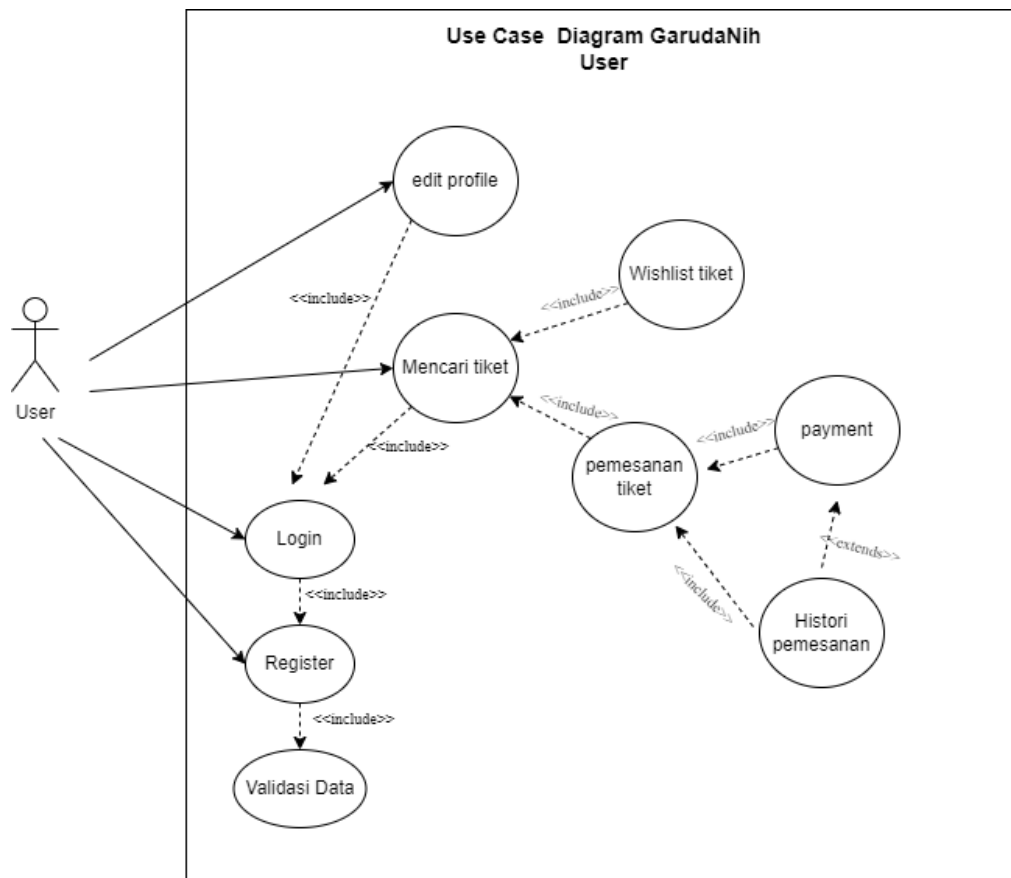


Gambar 4. 8 Use Case Superadmin GarudaNih

Berikut merupakan deskripsi dari *Use Case* Aplikasi GarudaNih Superadmin:

- *Superadmin (admin)*: orang yang dapat mengakses atau menggunakan keseluruhan sistem dari aplikasi GarudaNih. Sebagai orang dengan *role* tertinggi pada aplikasi.
- Melihat semua user yang mendaftar: Sebagai seseorang dengan *role* tertinggi, tentu saja tidak ada batasan untuk *superadmin (admin)*. User yang terdaftar dapat dilihat *by ascending* entah itu id, nama, email, dan tanggal lahir.
- Membuat dan menghapus admin atau user: Seorang *Superadmin* dapat membuat *admin* baru untuk membantunya dalam mengawasi *user*, dan seorang *admin* dapat membuat *user* baru. Hal ini sengaja dibuat dikarenakan untuk membantu seorang *user* yang belum mendapat KTP.

- Melihat, membuat, mengedit, dan menghapus tiket: Seorang *Superadmin* (*admin*) dapat Melihat berapa banyak tiket yang tersedia, membuat tiket baru untuk digunakan *user*, mengedit tiket sebelum *user* menggunakannya, dan menghapus tiket yang sudah usang.
- Melihat dan menghapus transaksi user: Seorang *Superadmin* (*admin*) dapat melihat semua riwayat transaksi yang user telah lakukan. Kemudian menghapus riwayat transaksi tersebut jika dirasa perlu.



Gambar 4. 9 Use Case User GarudaNih

Berikut merupakan deskripsi dari *Use Case* Aplikasi *GarudaNih Superadmin*:

- *User*: Orang yang dapat mengakses atau menggunakan aplikasi *GarudaNih*, mulai dari register dan login ke aplikasi hingga melakukan aksi terhadap aplikasi seperti *edit profile*, mencari tiket, *wishlist* tiket, memesan tiket, payment, dan melihat histori pemesanan.
- *Register*: Register merupakan langkah pertama yang dilakukan user ketika ia tidak mempunyai akses pada aplikasi *GarudaNih*. Mendaftarkan data diri

ke dalam aplikasi agar dikenali, bisa lewat halaman registrasi atau dengan akun google.

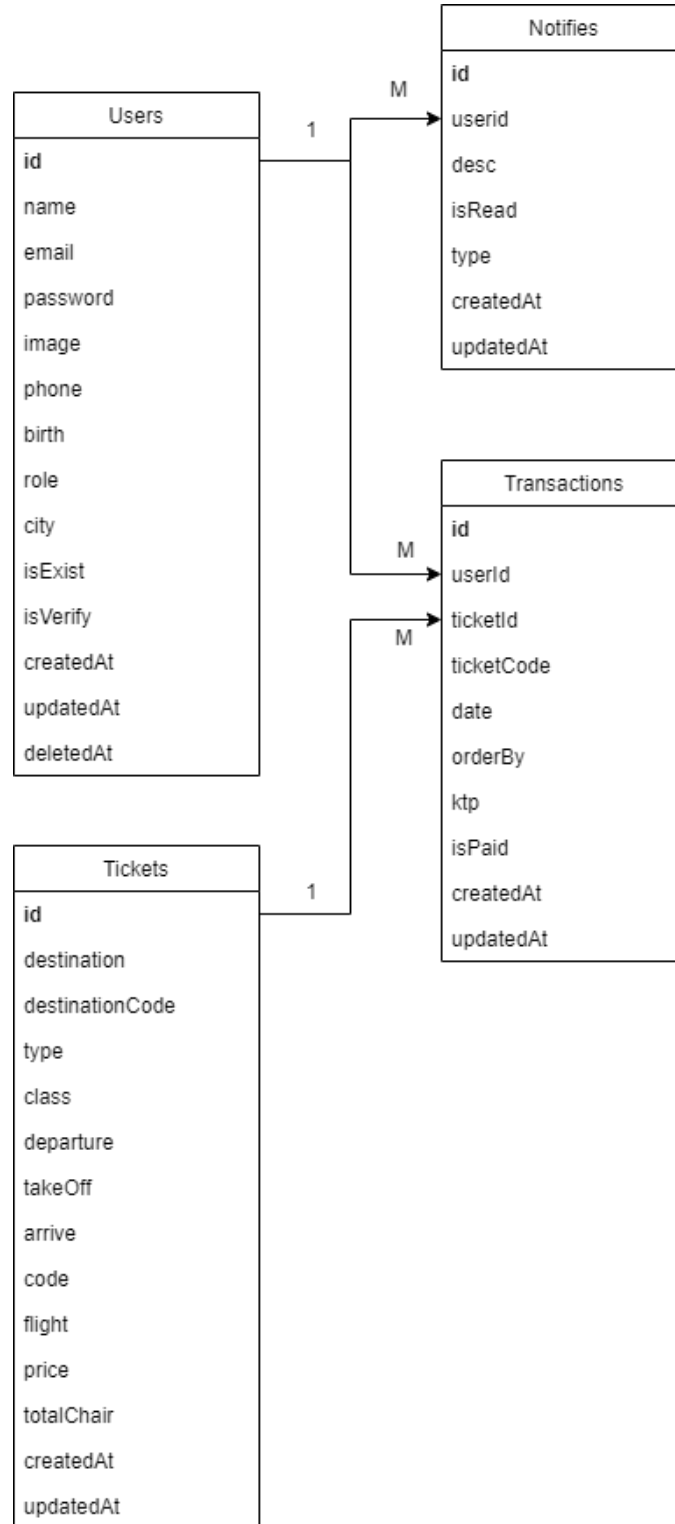
- *Login*: Setelah akun diaktifkan, user harus melakukan login agar dapat mengakses berbagai fitur aplikasi GarudaNih.
- *Edit Profile*: User disarankan melengkapi data diri pada menu ini. Seperti memasukkan nama, tanggal lahir, kota asal, nomor telepon, dan foto diri. Melengkapi data diri ini akan mempermudah validasi ketika user akan memesan tiket nantinya.
- *Mencari tiket*: Fitur utama dari aplikasi ini, user dapat mencari tiket yang diinginkan. Cukup lengkapi beberapa kolom seperti keberangkatan, tujuan, tanggal, serta kelas penerbangan. User juga perlu memilih *single trip* atau *round trip*.
- *Wishlist* tiket: Sebagai daftar keinginan user. Nantinya halaman ini menampilkan semua tiket yang menjadi keinginan user. User dapat memasukkan tiket keinginannya dan membeli tiketnya nanti.
- *Pemesanan tiket*: Suatu kegiatan dimana user nantinya akan memasukkan beberapa data sebagai bentuk validasi, nantinya user diminta memasukkan *identify number* (KTP), nama, dan nomor kursi.
- *Payment*: User nanti akan diarahkan ke halaman payment guna membayar tiketnya. Jika pembayaran berhasil, maka user berhak mendapatkan tiket dan tiket yang didapat akan dimasukkan ke dalam histori pembayaran. Jika gagal maka akan ada notifikasi pembayaran telah gagal.
- *Histori pemesanan*: Setelah memesan dan pembayaran dianggap berhasil, tiket yang dipesan akan dimasukkan ke dalam halaman ini. Nantinya user dapat melihat semua histori pemesanan yang telah dilakukannya.

#### **4.4 Rancangan Basis Data**

Dalam membuat aplikasi *GarudaNih*, dibutuhkan suatu perancangan basis data untuk menyimpan data user. Selain menyimpan data, perancangan ini juga berguna untuk mengambil data (*data retrieval*) yang kemudian akan ditampilkan melalui sisi pengguna (*client-side*).

#### 4.4.1 Class Diagram

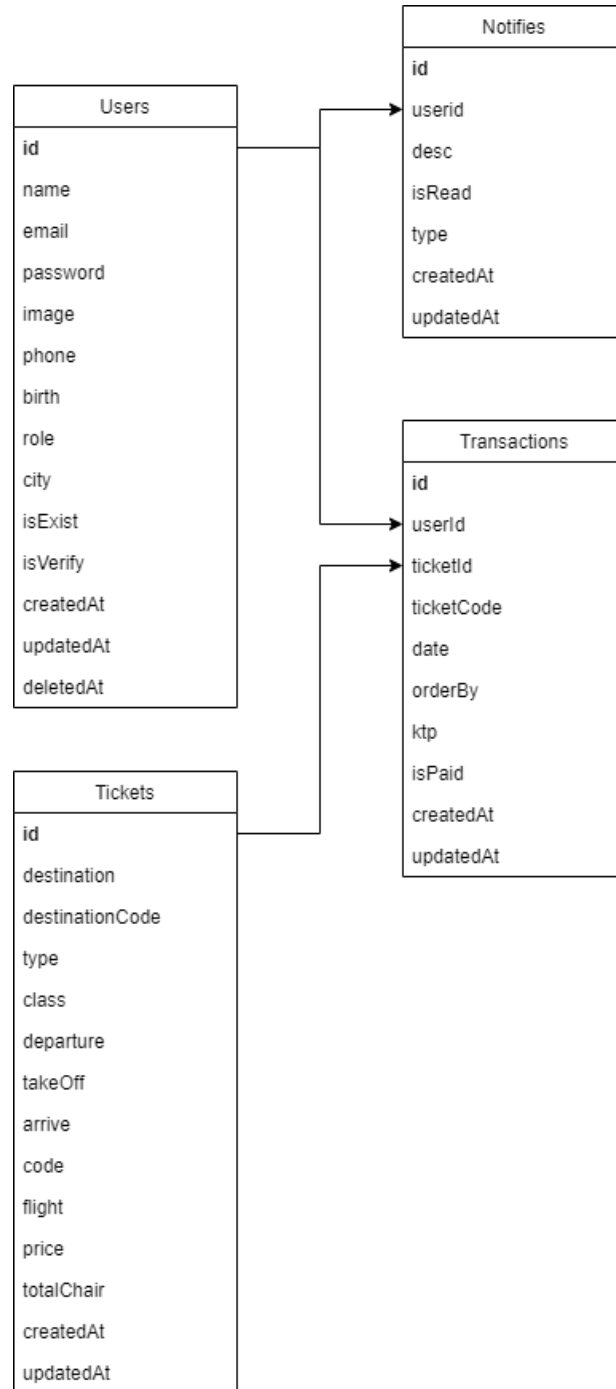
*Class Diagram* berikut akan memuat komponen-komponen entitas yang dilengkapi oleh atribut-atribut yang ada di dalam basis data.



Gambar 4. 10 Class Diagram

#### 4.4.2 LRS (*Logical Record Structure*)

Berikut adalah tampilan LRS (*Logical Record Structure*) untuk rancangan basis data aplikasi *GarudaNih*.



Gambar 4. 11 LRS (*Logical Record Structure*)

#### 4.4.3 Model Data Relational

Tahap ini merupakan tahap dimana Logical Record Structure diubah menjadi tabel yang menggambarkan basis data, Berikut merupakan model Data Relational dari aplikasi *GarudaNih*:

a. Tabel *Notifies*

id	userId	desc	isRead	type	createdAt	updatedAt
PK	FK					

b. Tabel *Tickets*

id	destination	destinationCode	type	class	departure	departureCode	takeOff	arrive	code	flight	price	totalChair	createdAt	updatedAt
PK														

c. Tabel *Users*

id	name	email	password	image	phone	birth	role	city	isExist	isVerify	createdAt	updatedAt	deletedAt
PK													

d. Tabel *Transactions*

id	userId	ticketId	ticketCode	date	orderBy	ktp	isPaid	createdAt	updatedAt
PK	FK								

#### 4.4.4 Spesifikasi Basisdata

Berikut adalah spesifikasi data yang dibutuhkan untuk rancangan basis data yang ada di dalam aplikasi *GarudaNih*.

- a. Nama Tabel : *Notifies*
- Media : *Server*
- Primary Key : *id*

<b>Nama Field</b>	<b>Jenis</b>	<b>Lebar</b>	<b>Keterangan</b>
id	int	11	id untuk <i>primary key</i> dalam tabel <i>notifies</i> .
userId	int	11	Untuk menyimpan id pengguna
desc	varchar	255	Deskripsi singkat dari notifikasi, misalnya ketika sukses melakukan pembayaran.
isRead	boolean	-	Penanda apakah user sudah membaca notifikasi tersebut atau belum.
type	varchar	255	Tipe dari notifikasi, misalnya sukses pembayaran, gagal pembayaran, voucher diskon.
createdAt	date	-	Data waktu dibuatnya notifikasi
updatedAt	date	-	Data waktu jika notifikasi ada yang diubah / diedit.

Tabel 4. 2 Tabel *Notifies*

- b. Nama Tabel : *Tickets*  
Media : *Server*  
*Primary Key* : id

<b>Nama Field</b>	<b>Jenis</b>	<b>Lebar</b>	<b>Keterangan</b>
Id	int	11	id untuk <i>primary key</i> dalam tabel <i>Tickets</i> .
destination	varchar	255	Deskripsi tujuan dari tiket. Memuat bandara tujuan dan kota tujuan.
destinationCode	varchar	255	Memuat kode singkat dari kota tujuan, misal tujuan Jakarta menjadi JKT.
type	varchar	255	Tipe yang tersedia. Terdapat dua pilihan, yaitu: adult dan child.
class	varchar	255	Kelas penerbangan yang tersedia. Terdapat tiga pilihan, yaitu: First Class, Business, dan Economy.
departure	varchar	255	Deskripsi keberangkatan. Memuat bandara keberangkatan dan kota keberangkatan.
departureCode	varchar	255	Memuat kode singkat dari kota keberangkatan, misal keberangkatan Malang menjadi MLG.
takeoff	date	-	Memuat waktu keberangkatan pesawat.
arrive	date	-	Memuat waktu tiba pesawat ke kota tujuan (estimasi).

code	varchar	255	Kode unik yang diberikan sebagai penanda bahwa tiket yang dibawa penumpang tidak sama, dan akan divalidasi ketika di bandara.
flight	varchar	255	Jenis penerbangan, tersedia lokal (domestic) dan luar negeri (internasional).
price	int	50	Harga satuan tiket.
totalChair	int	100	Banyaknya tempat duduk yang tersedia dalam satu penerbangan. Bisa dipakai untuk user memilih tempat duduk.
createdAt	date	-	Data waktu dibuatnya tiket.
updatedAt	date	-	Data waktu diubahnya tiket jika terjadi kesalahan pembuatan.

Tabel 4. 3 Tabel Ticket

- c. Nama Tabel : *Users*  
Media : *Server*  
Primary Key : *id*

Nama Field	Jenis	Lebar	Keterangan
Id	int	11	id untuk <i>primary key</i> dalam tabel <i>Users</i> .
name	varchar	255	Nama dari pengguna.
email	varchar	255	Email dari pengguna.

password	varchar	255	Password dari pengguna (sudah di-enkripsi).
image	varchar	255	Link dari gambar yang diunggah pengguna. Karena foto asli disimpan menggunakan layanan cloud (cloudinary).
phone	varchar	255	Nomor pribadi pengguna. Bisa digunakan sebagai remainder ketika pesawat akan lepas landas atau sebagai sarana promosi.
birth	date	-	Tanggal lahir pengguna, sebagai validasi bahwa pengguna telah cukup umur dalam memesan tiket/
role	varchar	255	<i>Role</i> atau jabatan dalam aplikasi. Semakin tinggi role, semakin leluasa menggunakan fitur dalam aplikasi
city	varchar	255	Kota asal / kelahiran dari pengguna.
isExist	boolean	-	Sebagai bentuk validasi apakah akun yang user gunakan masih aktif atau non-aktif.
isVerify	boolean	-	Sebagai bentuk validasi apakah user sudah melakukan verifikasi atau belum. Jika belum, user

			tidak akan diizinkan masuk aplikasi.
createdAt	date	-	Data waktu user membuat akun.
updatedAt	date	-	Data waktu user mengedit akun, bisa karena melengkapi data diri atau menambahkan foto.
deletedAt	date	-	Data waktu pe-nonaktifan akun pengguna.

Tabel 4. 4 Tabel User

d. Nama Tabel : *Transactions*

Media : *Server*

Primary key : id

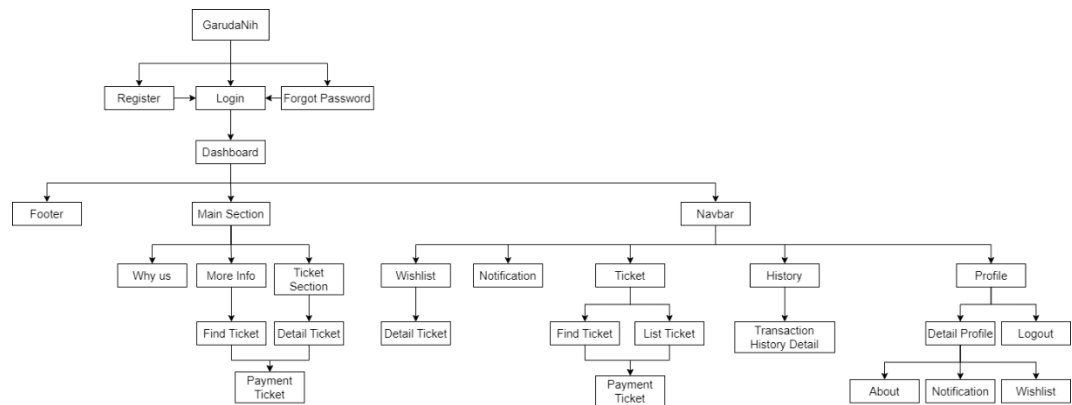
<b>Nama Field</b>	<b>Jenis</b>	<b>Lebar</b>	<b>Keterangan</b>
Id	int	11	id untuk <i>primary key</i> dalam tabel <i>Transactions</i> .
userId	int	11	Untuk menyimpan id pengguna.
ticketId	int	11	Untuk menyimpan id tiket.
ticketCode	varchar	255	Untuk menyimpan kode unik pada tiket yang pasti berbeda antara user dengan user lainnya.
date	date	-	Data transaksi dibuat.
orderBy	varchar	255	Nama pemesan tiket.
ktp	varchar	255	Nomor ktp yang diinputkan pengguna.

isPaid	boolean	-	Sebagai penanda apakah user sudah melakukan pembayaran atau belum.
createdAt	date	-	Data waktu dibuatnya transaksi, bisa karena berhasil atau gagal.
updatedAt	date	-	Data waktu transaksi diedit.

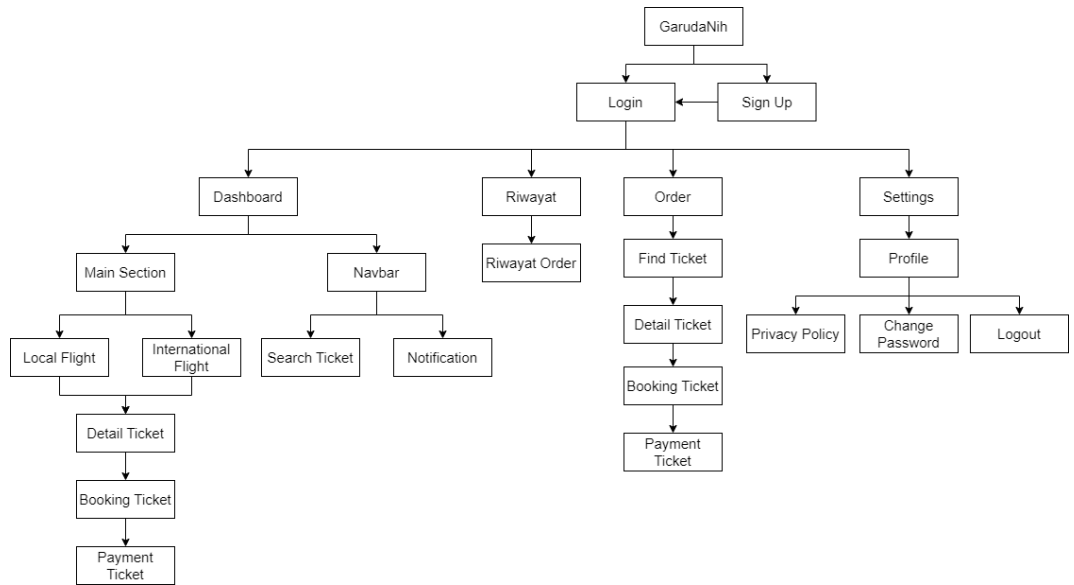
Tabel 4. 5 Tabel Transaction

#### 4.5 Struktur Tampilan

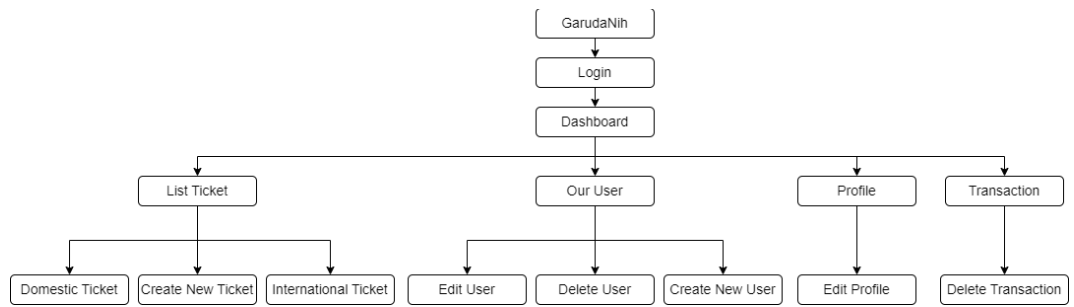
Sebelum membangun suatu aplikasi, struktur tampilan (menu) merupakan suatu hal yang penting dibuat sehingga pada saat pengerjaan aplikasi bisa lebih efisien. Berikut adalah struktur tampilan (menu) atau halaman-halaman yang akan ada di aplikasi *GarudaNih*, mencakup menu untuk website, mobile dan admin.



Gambar 4. 12 Struktur Tampilan (menu) Website



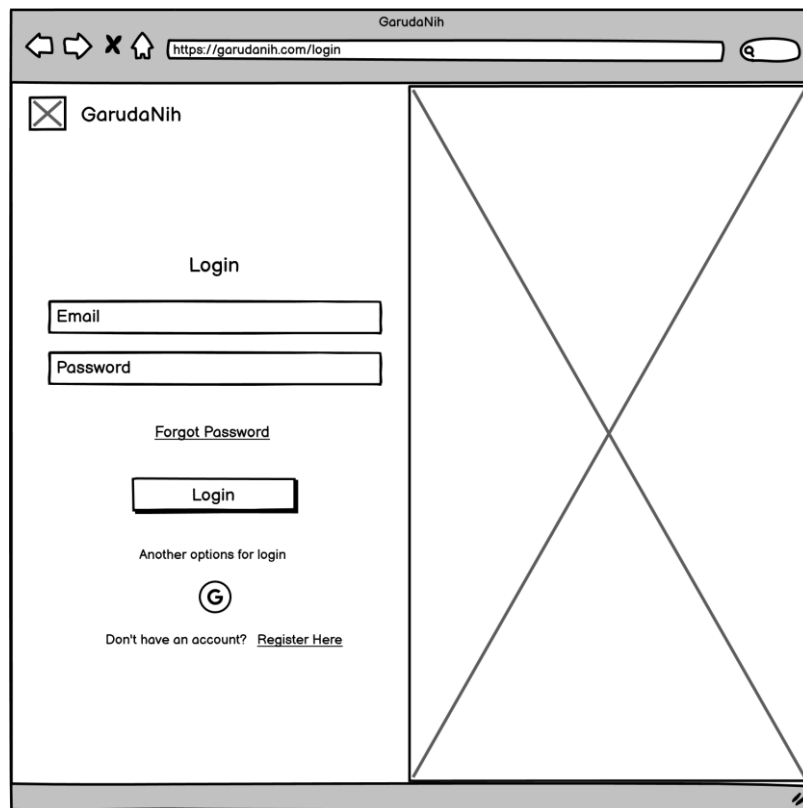
Gambar 4. 13 Struktur Tampilan (menu) Mobile



Gambar 4. 14 Struktur Tampilan (menu) Admin

## 4.6 Rancangan Layar (User Interface)

### 4.6.1 Rancangan Layar Login – Website



Gambar 4. 15 Rancangan Layar Login - Website

Ketika pertama kali membuka halaman website *GarudaNih*, pengguna diminta untuk login terlebih dahulu. Pengguna memasukkan email dan password yang dimasukkannya ketika mendaftar. Opsi lainnya yaitu menggunakan google login. Sistem hanya akan meminta izin untuk mengambil email, nama dan foto profil. Jika belum mempunyai akun, user bisa klik tombol *Register Here*. Pada halaman ini juga terdapat *link* untuk *forgot password* jika user lupa passwordnya.

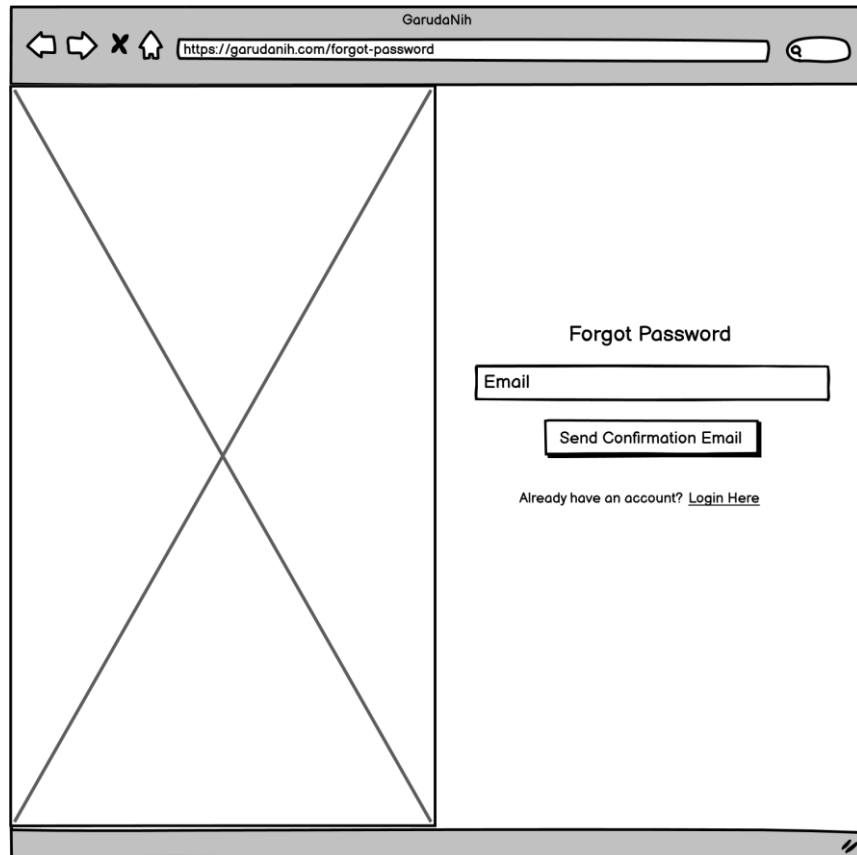
#### 4.6.2 Rancangan Layar Register – Website

The image shows a wireframe of a web browser window. The browser's address bar contains the URL "https://garudanih.com/register". The page content is divided into two main areas. On the left, there is a large rectangular area with a large 'X' drawn across it, representing a placeholder for a logo or image. On the right, there is a registration form titled "Register". The form consists of four input fields: "Name", "Email", "Password", and "Confirm Password". Below these fields is a "Register" button. At the bottom of the form area, there is a link that says "Already have an account? [Login Here](#)".

Gambar 4. 16 Rancangan Layar Register - Website

Halaman *Register* berisi 4 kolom yang user masukkan sebagai data awal untuk mendaftar pada aplikasi *GarudaNih*. Kolom nama bisa dikosongkan dan diisi nanti, tapi 3 kolom lainnya yaitu: email, password, dan confirm password wajib diisi. Jika sudah tombol register akan mengirim data ke server dan user bisa melanjutkan aktivitasnya dengan mengklik *login here*. Jika sistem mendeteksi bahwa ada data yang tidak valid pada kolom tertentu, misalnya pada email tidak terdapat simbol “@” dan kolom password dengan kolom confirm password tidak sama, maka ketika di klik tombol register akan ada pesan error.

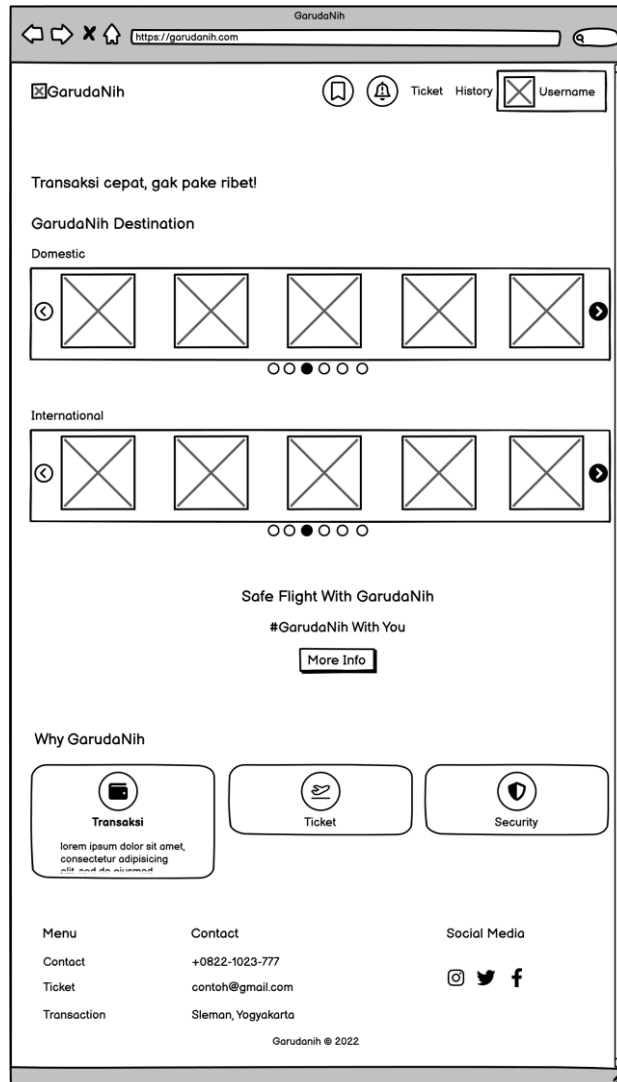
#### 4.6.3 Rancangan Layar Lupa Password – Website



Gambar 4. 17 Rancangan Layar Forgot Password - Website

Halaman *forgot password* digunakan ketika user lupa akan passwordnya. User hanya diminta mengisi kolom email sesuai dengan email yang didaftarkannya. Dengan mengklik tombol *send confirmation email*, maka sistem akan mengirimkan cara untuk *me-reset password* ke email tersebut. Selanjutnya klik *login here* agar user dapat melanjutkan aktivitas pada web *GarudaNih*.

#### 4.6.4 Rancangan Layar Dashboard – Website

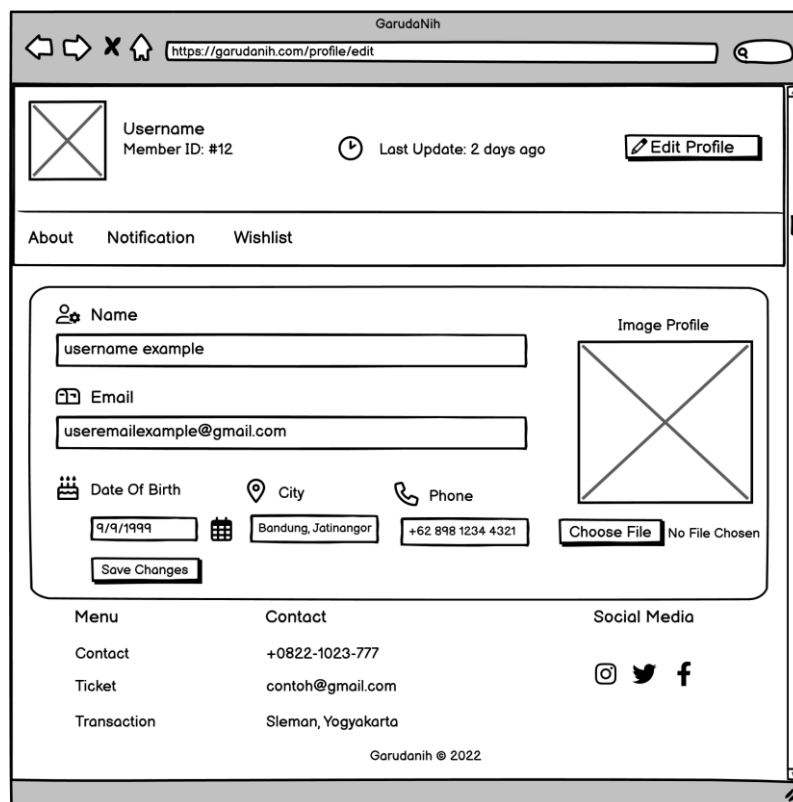


Gambar 4. 18 Rancangan layar Dashboard - Website

Halaman *dashboard* adalah halaman yang terbuka setelah user melakukan *login*. Pada bagian *navbar* terdapat 5 item yang bisa diinteraksi oleh user. Pada sisi kanan atas terdapat foto dan username yang sedang login. Bila di klik nanti akan muncul 2 menu tambahan *profile* dan *logout*. Lalu *history* digunakan untuk melihat semua riwayat transaksi user. *Ticket* digunakan untuk mencari tiket yang spesifik. Dua terakhir adalah notifikasi dan *wishlist*. Notifikasi menampilkan semua aksi yang dilakukan oleh user, seperti pemesanan tiket, pembayaran tiket berhasil dilakukan. *Wishlist* digunakan untuk menyimpan dan menandai tiket yang

ingin dibeli user nanti. Masuk main section, terdapat semua tiket yang tersedia, baik domestic (lokal) dan internasional (luar negeri). Tombol *more info* digunakan untuk *redirect* ke halaman mencari tiket. Sedangkan *why GarudaNih* menjelaskan kelebihan website ini. Terakhir, bagian footer berisi informasi tambahan yang ada pada website, seperti menu yang bila di klik akan menuju *section* yang sesuai. Terdapat juga kontak dan social media supaya user mudah dalam terhubung dengan pihak GarudaNih.

#### 4.6.5 Rancangan Layar Profile.Edit Profile – Website

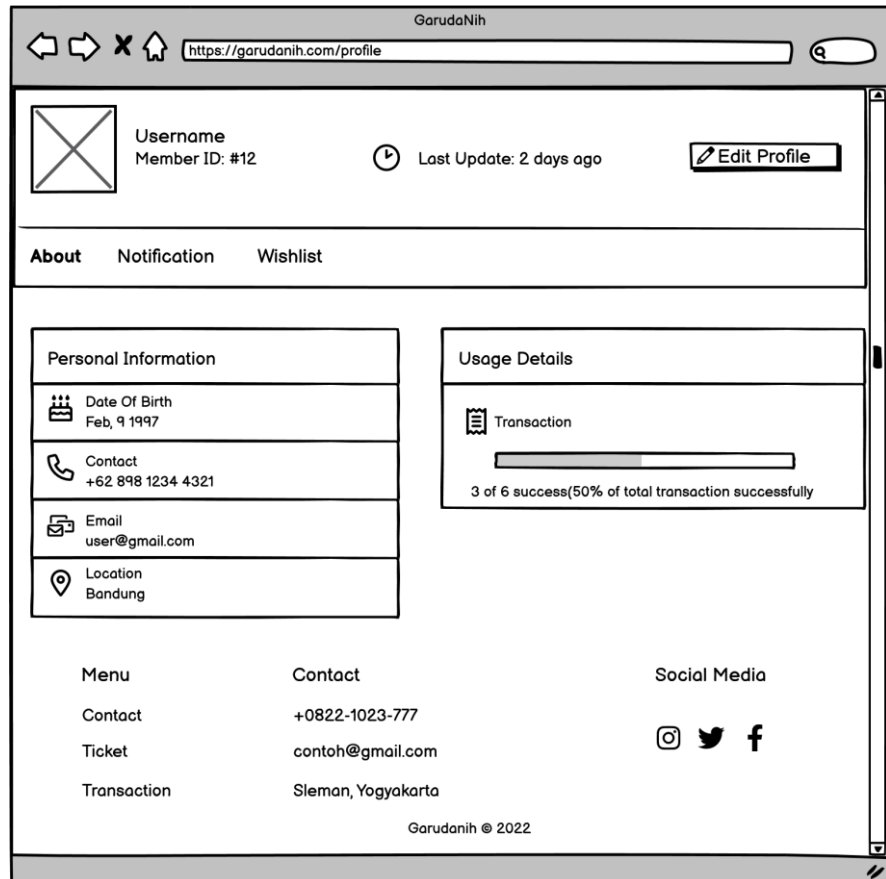


Gambar 4. 19 Rancangan Layar Edit Profile - Website

Halaman *profile-edit profile* digunakan user untuk melengkapi data diri. Data diri yang dimaksud mencakup nama (jika belum mengisi pada register), tanggal lahir, kota asal, nomor telepon dan foto profil. Pelengkapan data diri ini dimaksudkan agar admin dapat memverifikasi bahwa user yang mendaftar aktif. Selain itu halaman ini sesuai dengan

namanya, yaitu *edit profile*. Untuk mengedit data diri yang tidak sesuai ketika proses registrasi sebelumnya. Tombol *save changes* di klik apabila data diri sudah sesuai dan untuk memperbarui data diri yang sudah diisi.

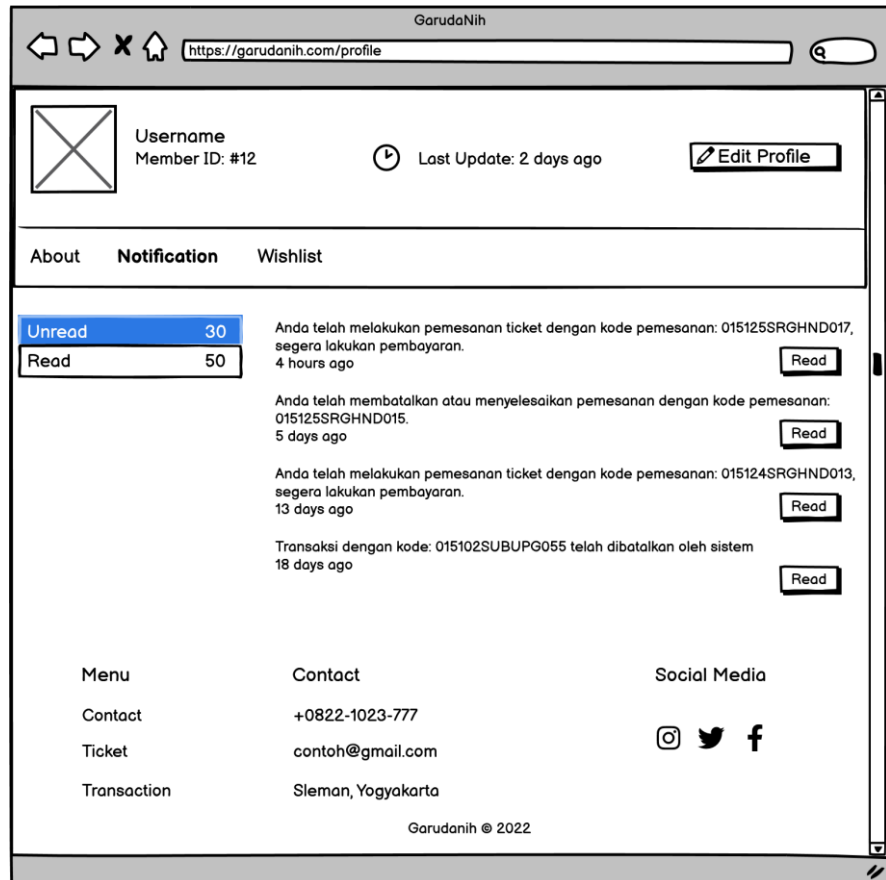
#### 4.6.6 Rancangan Layar Profile.About – Website



Gambar 4. 20 Rancangan Layar About - Website

Halaman *profile-about* berisi semua yang berkaitan dengan user, dimulai dari data diri yang sebelumnya sudah diisi pada halaman *edit profile* tentunya dengan tampilan yang lebih baik. Terdapat juga detail transaksi yang sudah dilakukan, ditampilkan dengan *bar progress* dan ditambahkan dengan persentase kesuksesan transaksi.

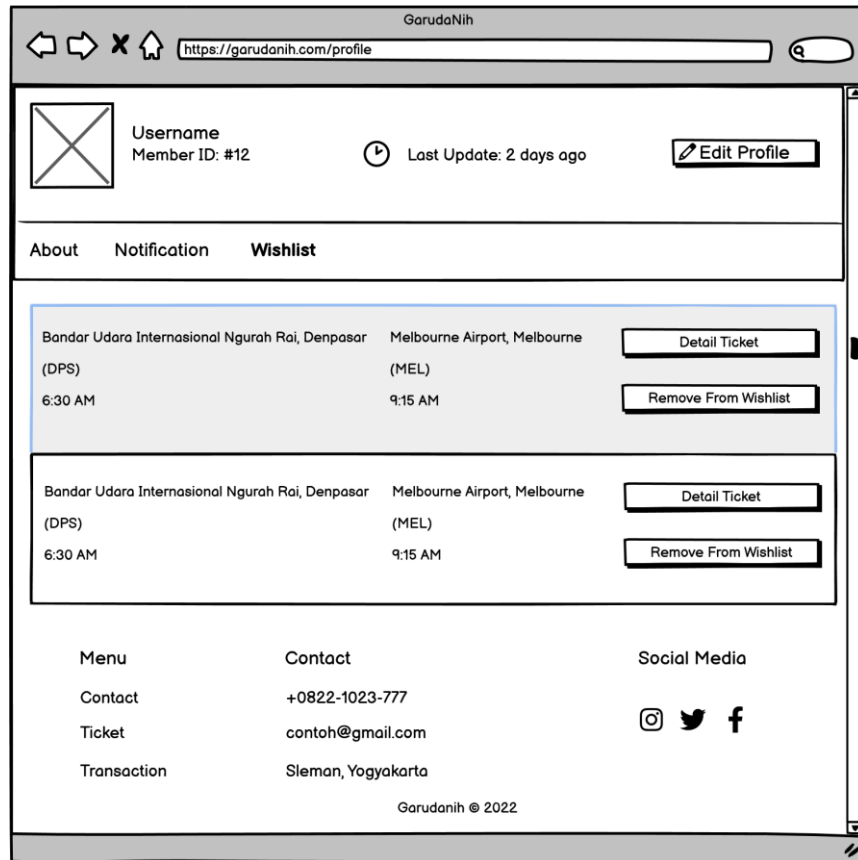
#### 4.6.7 Rancangan Layar Profile.Notification – Website



Gambar 4. 21 Rancangan Layar Notification - Website

Halaman *Profile-Notification* adalah halaman dimana user dapat melihat semua notifikasi yang ada. User dapat melihat notifikasi yang sudah dibaca maupun yang sudah dibaca. Tombol *read* digunakan untuk membaca notifikasi dan akan melempar notifikasi tersebut ke halaman *read* atau yang sudah dibaca.

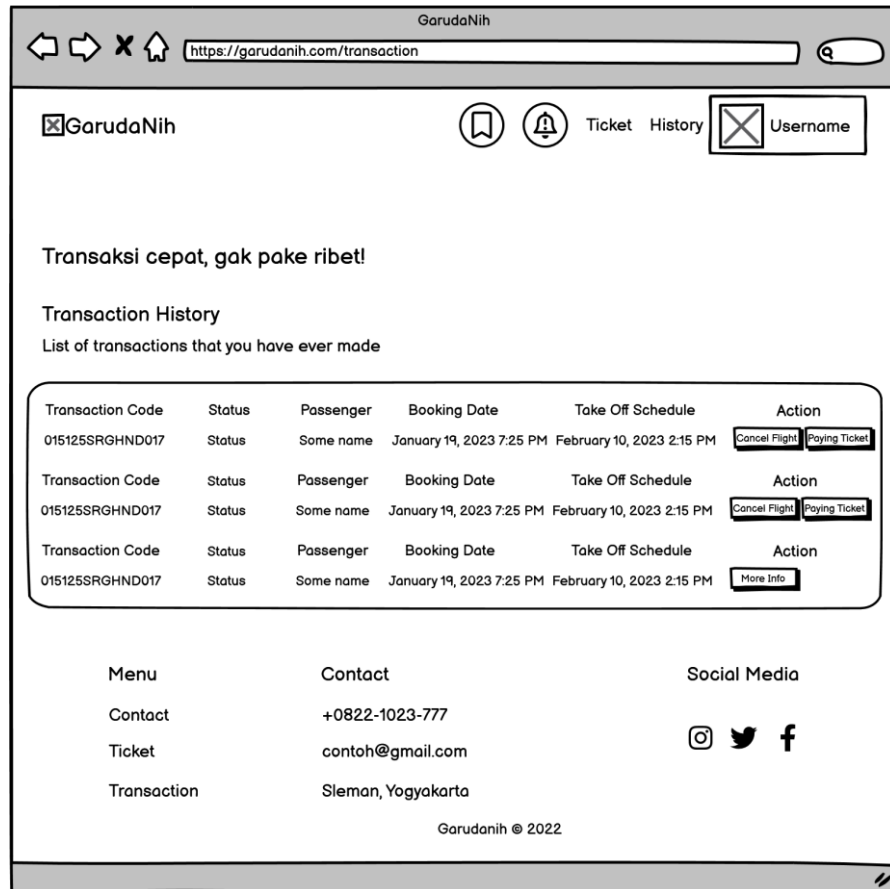
#### 4.6.8 Rancangan Layar Profile.Wishlist – website



Gambar 4. 22 Rancangan Layar Wishlist - Website

Halaman *Profile-Wishlist* adalah halaman dimana user dapat melihat semua tiket yang sudah di *wishlist*-nya. Terdapat tombol *Detail Ticket* yang berguna untuk melihat detail dari tiket. *Remove From Wishlist* adalah tombol yang digunakan untuk menghapus tiket dari daftar *wishlist*.

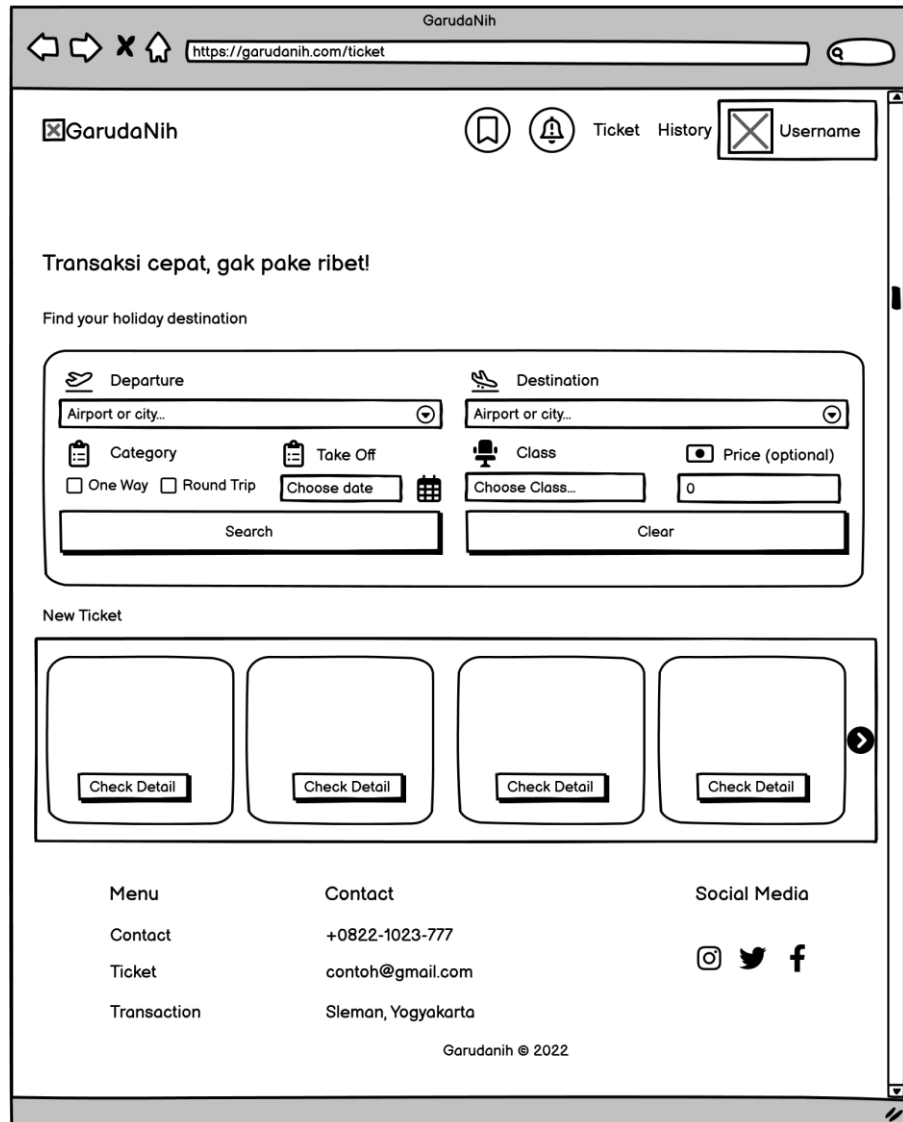
#### 4.6.9 Rancangan Layar History – Website



Gambar 4. 23 Rancangan Layar History - Website

Halaman *History* adalah dimana user dapat melihat semua riwayat transaksi yang pernah dilakukannya. Mulai dari data user memesan tiket pada jam berapa, transaksi kodenya, hingga status pembayarannya. Halaman ini juga menjadi cara untuk user mengkonfirmasi, apakah tiket yang sudah dipesan akan dibayar (dengan mengklik tombol *paying ticket*) atau justru dibatalkan (dengan mengklik tombol *cancel flight*). Sedangkan tombol *More info* hanya untuk menampilkan detail tiket yang sudah dibayar.

#### 4.6.10 Rancangan Layar Ticket – Website

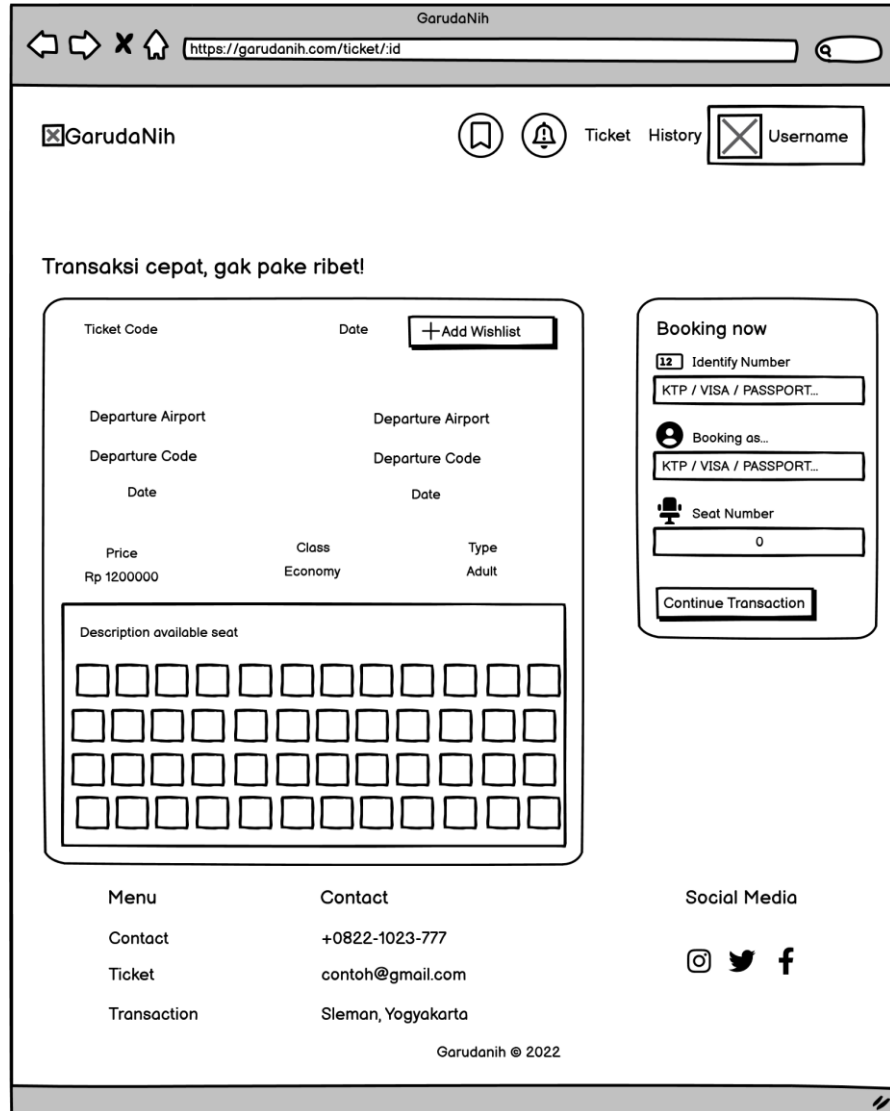


Gambar 4. 24 Rancangan Layar Find Ticket - Website

Halaman *Ticket* adalah halaman dimana user dapat mencari tiket spesifik yang diinginkannya. Kolom pertama yang diisi adalah *departure* atau keberangkatan, kedua adalah *destination* atau tujuan. Setelah itu pilih *one way* (1x perjalanan) atau *round trip* (2x perjalanan, pulang-pergi). Pilih jam keberangkatan serta class penerbangan. Terakhir isi *price* atau harga (optional), lalu tekan tombol *search*. Sistem akan menampilkan tiket yang sesuai dengan yang sudah diisikan tadi. Tombol *clear* untuk

menghapus hasil pencarian. Atau bisa pilih tiket pada bagian bawah untuk melihat-lihat tiket yang sudah ada sebelumnya.

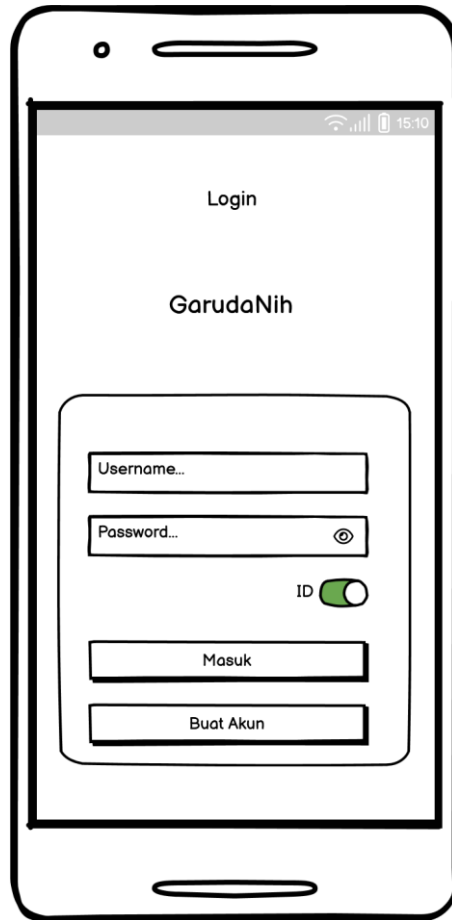
#### 4.6.11 Rancangan Layar Booking Ticket – Website



Gambar 4. 25 Rancangan Layar Booking Ticket - Website

Halaman *Booking Ticket* adalah halaman dimana user akan melakukan pemesanan tiketnya. Tetapi sebelum itu user diminta untuk mengisi beberapa data seperti yang tertera pada bagian kanan layar. User juga dapat menambahkan tiket ini ke dalam daftar *wishlist* jika dirasa tidak ingin memesan untuk saat ini. Pemilihan nomor kursi bisa dipilih di bagian bawah dengan cara di klik seperti biasa.

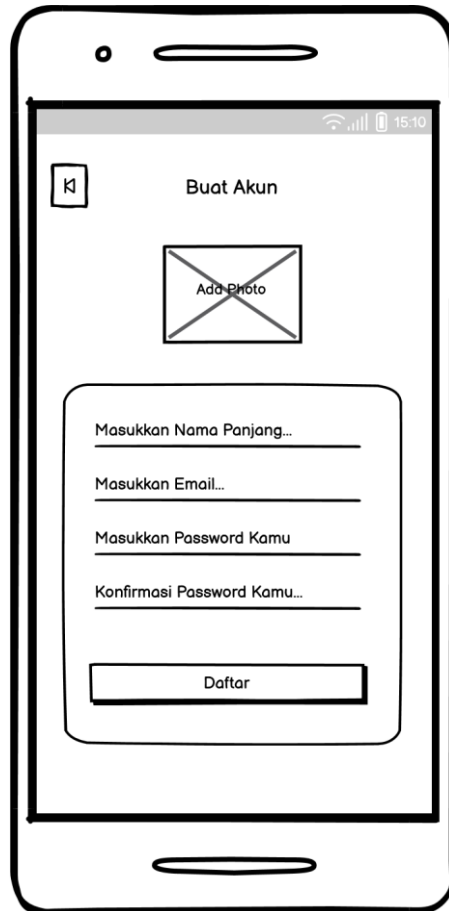
#### 4.6.12 Rancangan Layar Login – Android



Gambar 4. 26 Rancangan Layar Login - Android

Halaman *login* mobile (android), halaman yang pertama kali dijumpai. User diminta mengisi username dan password setelah itu klik tombol masuk. Tombol buat akun digunakan untuk membuat akun pertama kali.

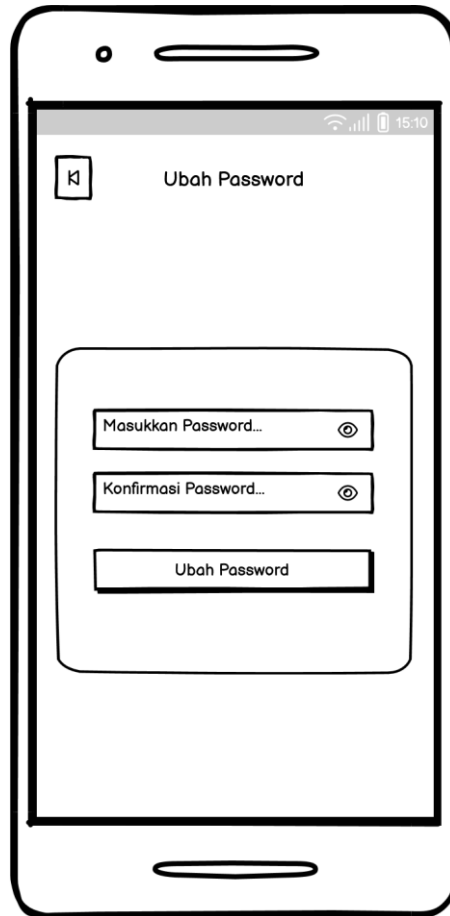
#### 4.6.13 Rancangan Layar Buat Akun – Android



*Gambar 4. 27 Rancangan Layar Buat Akun - Android*

Selanjutnya adalah halaman registrasi. Halaman ini merupakan untuk mendaftar pertama kali. Yang dibutuhkan ada nama panjang, email, password, dan konfirmasi password. Tidak lupa juga untuk mengunggah foto untuk dijadikan foto profil nantinya. Tombol daftar untuk mengirim data ke server, dan akun kamu siap digunakan.

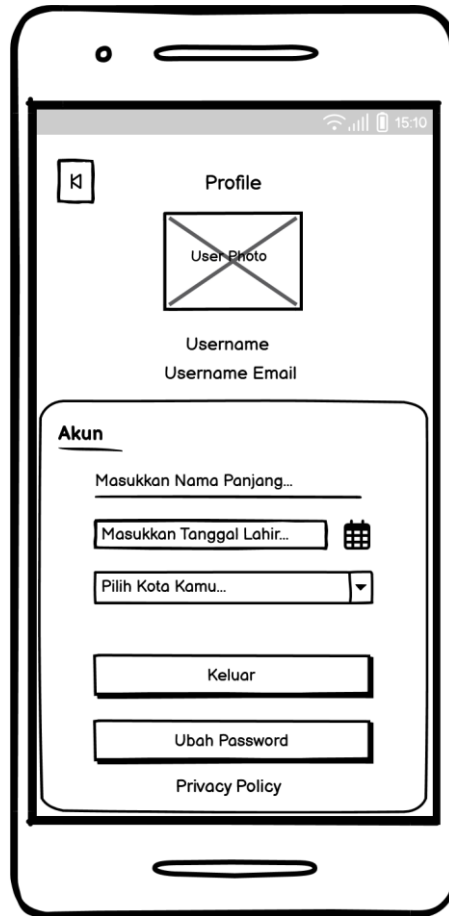
#### 4.6.14 Rancangan Layar Ubah Password – Android



*Gambar 4. 28 Rancangan layar Ubah Password - Android*

Halaman ini digunakan untuk mengubah password lama kamu, menjadi password baru. Silahkan masukkan password baru dan konfirmasi password baru kamu, setelah itu klik ubah password untuk mengubahnya.

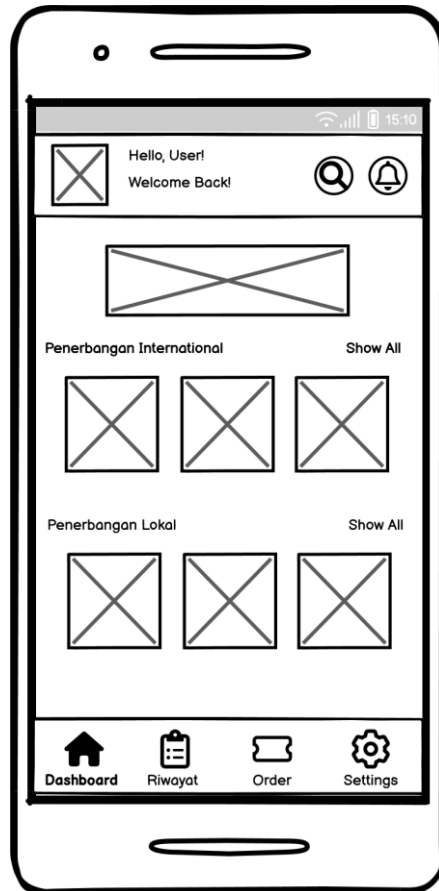
#### 4.6.15 Rancangan Layar Profile – Android



Gambar 4. 29 Rancangan Layar Profile - Android

Halaman *Profile* adalah halaman yang digunakan untuk melengkapi data diri dan mengedit data diri jika terdapat kesalahan pada registrasi sebelumnya. Semua kolom wajib diisi agar memudahkan sistem dalam memverifikasi. Tombol keluar untuk membatalkan aksi pengeditan dan ubah password untuk masuk ke halaman ubah password. Privacy Policy adalah kebijakan privasi aplikasi.

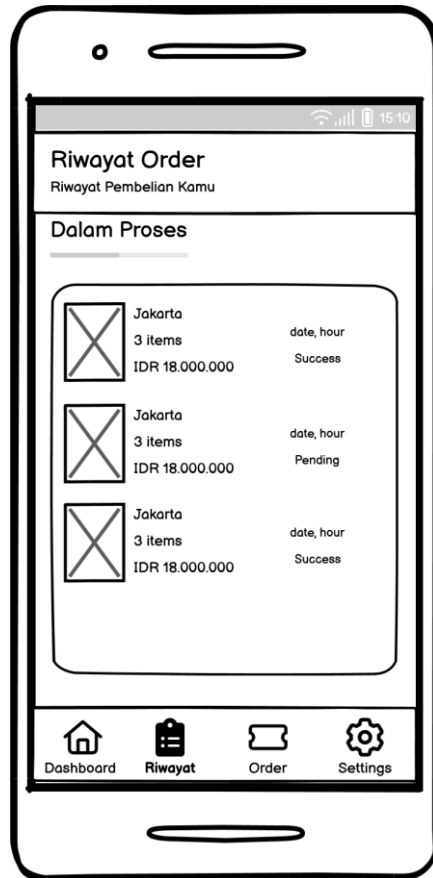
#### 4.6.16 Rancangan Layar Home – Android



Gambar 4. 30 Rancangan Layar Home - Android

Halaman *home* adalah halaman utama tempat user dapat memesan tiket. Di bagian *navigation bar* sebelah kanan terdapat tombol notifikasi dan tombol pencarian tiket. Lalu main section diisi dengan semua tiket penerbangan international dan domestic. Bagian bawah terdapat *Bottom Navigation* yang digunakan untuk berpindah halaman antar dashboard, riwayat, order dan settings.

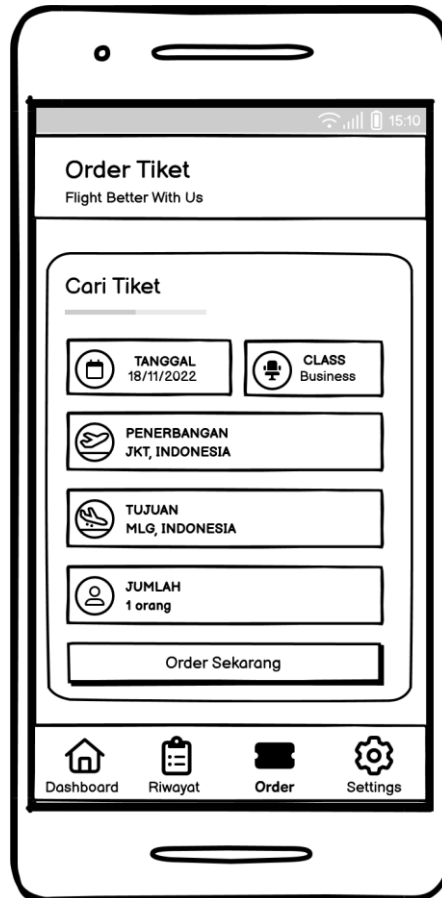
#### 4.6.17 Rancangan Layar Riwayat Order – Android



Gambar 4. 31 Rancangan Layar Riwayat Order - Android

Halaman riwayat digunakan untuk melihat semua riwayat transaksi yang sudah user lakukan. Terdapat pula informasi lain seperti tiket, harga, waktu pemesanan dan status dari transaksi tersebut.

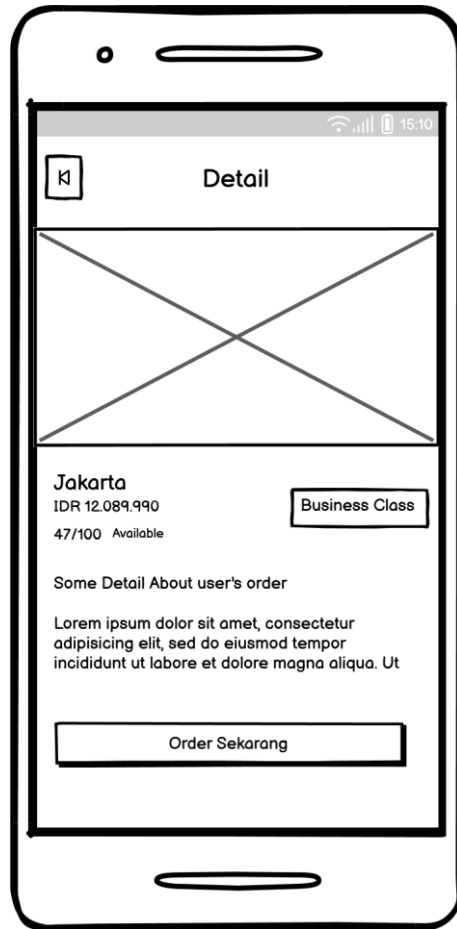
#### 4.6.18 Rancangan Layar Order – Android



Gambar 4. 32 Rancangan Layar Order - Android

Halaman order digunakan untuk mencari tiket. Kolom yang harus diisi user pun mencakup tanggal pergi, class, keberangkatan dan tujuan, serta jumlah tiket. Tombol order sekarang akan menampilkan semua list ticket sesuai data yang sudah user isikan.

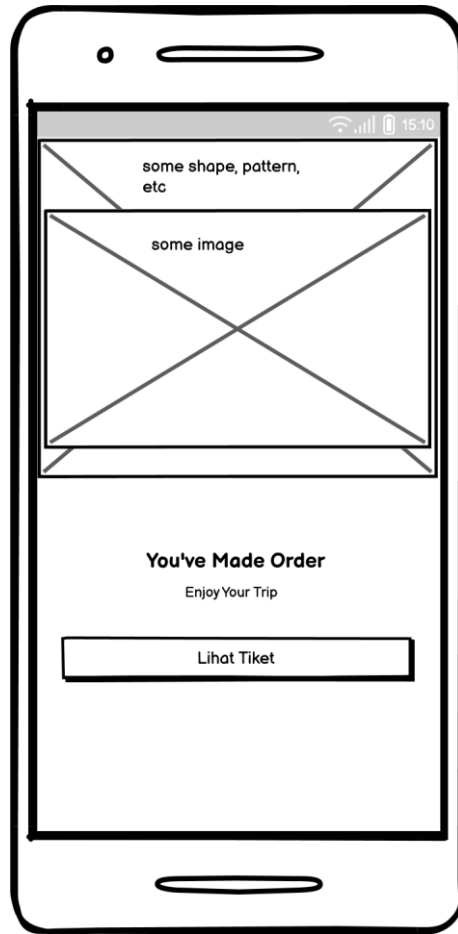
#### 4.6.19 Rancangan Layar Detail Ticket – Android



*Gambar 4. 33 Rancangan Layar Detail Ticket - Android*

Halaman detail ticket adalah halaman yang berisi informasi tiket yang akan dipesan oleh user. Informasi yang terkandung mencakup data harga, data user yang melakukan pemesanan, dan data kelas penerbangan pada tiket tersebut. Tombol order sekarang untuk melanjutkan transaksi tiket yang sudah dipilih.

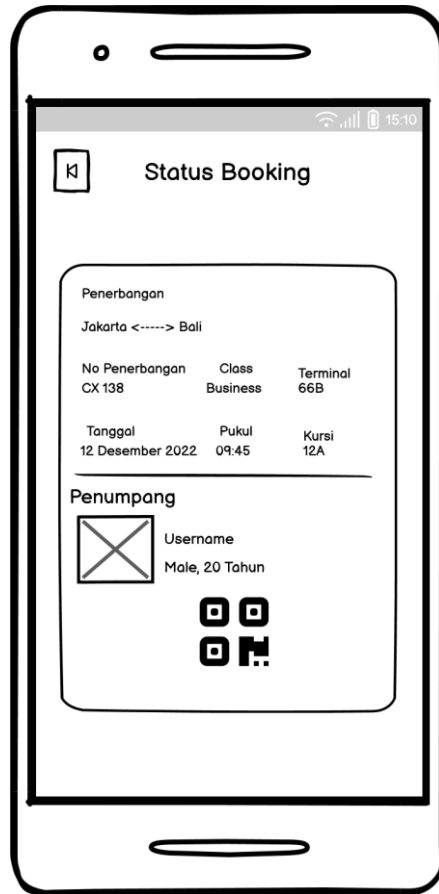
#### 4.6.20 Rancangan Layar Order Detail – Android



Gambar 4. 34 Rancangan Layar Order Detail - Android

Halaman order detail akan memberi tahu kalau tiket sudah berhasil dipesan, dan tombol lihat tiket akan *mendirect* kita untuk melihat detail ticket yang kita sudah pesan.

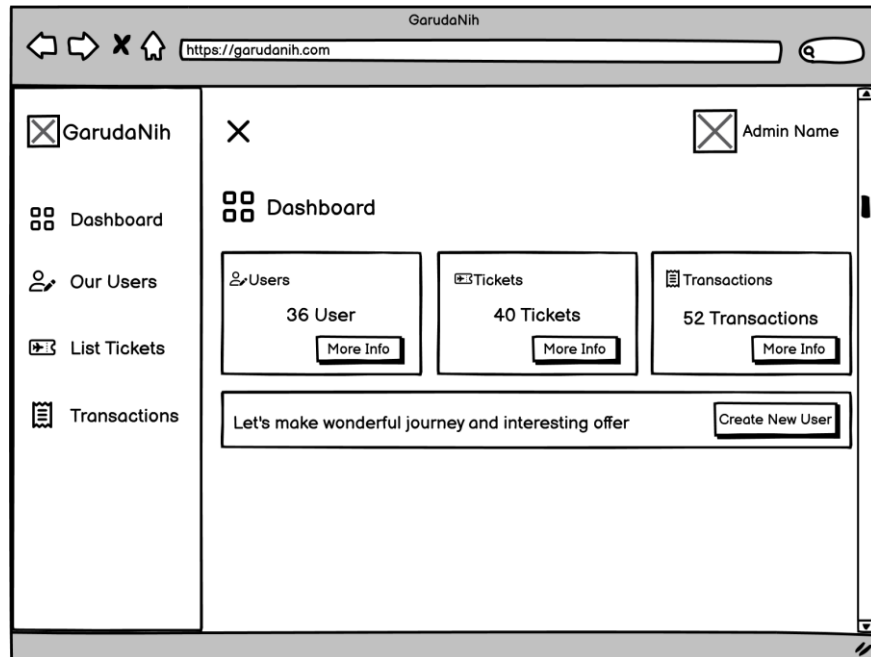
#### 4.6.21 Rancangan Layar Status Booking – Android



Gambar 4. 35 Rancangan Layar Status Booking - Android

Halaman terakhir yaitu *status booking*, di sini user akan diperlihatkan detail ticket dan status booking. Informasi yang ditampilkan pun seperti nomor penerbangan, terminal, class, kursi dan waktu dari keberangkatan tiket tersebut. Dibawahnya terdapat sedikit data penumpang dan *barcode* dari tiket tersebut.

#### 4.6.22 Rancangan Layar Dashboard – Admin



Gambar 4. 36 Rancangan Layar Dashboard - Admin

Halaman *dashboard admin* adalah halaman dimana para *superadmin (admin)* memantau data-data terkait user, tiket, dan transaksi. Pada halaman ini juga para *superadmin (admin)* dapat membuat *admin* baru dan tiket yang baru sesuai jadwal penerbangan yang sudah disediakan. Nantinya tiket yang dibuat dapat terlihat oleh user.

#### 4.6.23 Rancangan Layar Profile – Admin

The screenshot displays the Admin Profile page within the GarudaNih application. The browser address bar shows the URL `https://garudanih.com/admin`. The page layout includes a sidebar on the left with navigation links: Dashboard, Our Users, List Tickets, and Transactions. The main content area is titled 'Profile' and contains a form for editing the admin's information. The form fields are as follows:

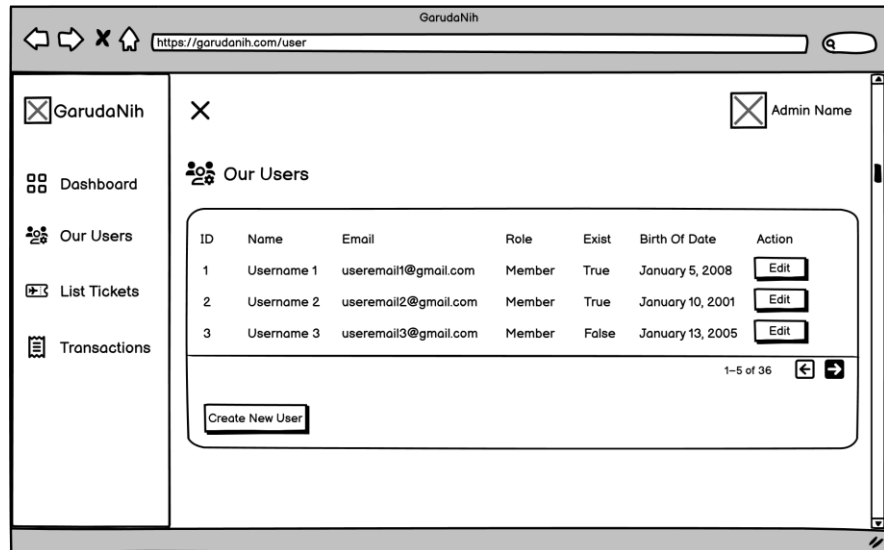
Field	Value
Name	admin name example
Email	adminemailexample@gmail.com
Date Of Birth	9/9/1999
City	Bandung, Jatinangor
Phone	+62 898 1234 4321

The 'Image Profile' section includes a 'Choose File' button and the text 'No File Chosen'. A 'Save Changes' button is located at the bottom of the form.

Gambar 4. 37 Rancangan Layar Profile - Admin

Halaman *profile* ini digunakan untuk mengedit data diri admin terkait. Pengeditan ini dilakukan seperti tanda pengenal admin. Hampir sama seperti user, admin wajib mengisi semua kolom yang sudah disediakan. Tombol *save changes* untuk menyimpan perubahan.

#### 4.6.24 Rancangan Layar Our Users – Admin



Gambar 4. 38 Rancangan Layar Our Users - Admin

Halaman *Our Users* adalah halaman dimana semua user yang telah mendaftar dapat dilihat. Pada kolom id, name, email, role, exist dan bod dapat dilakukan sorting berdasarkan *ascending* maupun *descending*. *Superadmin (admin)* juga dapat melakukan aksi untuk mengedit data user. Ini dilakukan agar data diri yang dimasukkan oleh user tidak terkesan asal dan mencegah pengunggahan foto tidak senonoh.

#### 4.6.25 Rancangan Layar Create New Admin – Admin

The screenshot displays a web browser window with the URL `https://garudanih.com/user/create`. The browser's title bar reads "GarudaNih". The page layout includes a sidebar on the left with the following menu items: GarudaNih, Dashboard, Our Users, List Tickets, and Transactions. The main content area is titled "Our Users / Create" and features a form for creating a new admin user. The form fields are as follows:

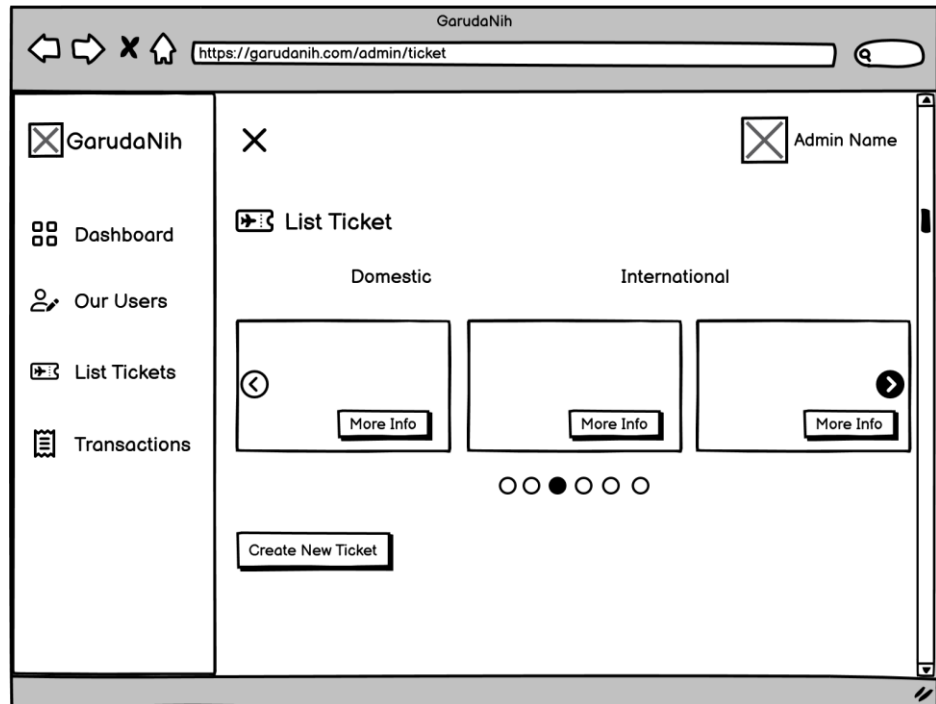
- Name:** A text input field with the placeholder "Enter fullname...".
- Email:** A text input field containing the example email "useremailexample@gmail.com".
- Password:** A text input field with the placeholder "Enter password...".
- Confirm Password:** A text input field with the placeholder "Re-type your password...".
- Date Of Birth:** A date picker field with the placeholder "dd/mm/yyyy".
- City:** A text input field with the placeholder "Enter city...".
- Phone:** A text input field with the placeholder "08xx xxxx xx93".

There is also an "Image Profile" section with a "Choose File" button and the text "No File Chosen". At the bottom right of the form, there are two buttons: "Cancel" and "Create New User".

Gambar 4. 39 Rancangan Layar Create New Admin - Admin

Halaman *create admin* adalah halaman dimana *superadmin* sebagai pemegang role tertinggi di sistem, dapat melakukan pembuatan admin baru untuk membantu tugasnya dalam memantau user-user.

#### 4.6.26 Rancangan Layar List Tickets – Admin



Gambar 4. 40 Rancangan Layar List Tickets - Admin

Halaman *List Ticket* berisi semua tiket-tiket yang sudah disediakan oleh admin. List ticket dapat dibagi menjadi dua yaitu domestic dan internasional.

#### 4.6.27 Rancangan Layar Create New Ticket – Admin

The screenshot shows a web browser window with the URL `https://garudanih.com/user/create`. The page title is 'GarudaNih'. The sidebar on the left contains the following menu items: 'GarudaNih', 'Dashboard', 'Our Users', 'List Tickets', and 'Transactions'. The main content area is titled 'Ticket / Create' and contains the following form fields:

- Departure: Airport name... (text input), ICAO (text input)
- Destination: Airport name... (text input), ICAO (text input)
- Take Off: 01/20/2023 04:58 (text input)
- Arrive: 01/21/2023 03:20 (text input)
- Type: Choose Type (dropdown menu)
- Class: Choose Class (dropdown menu)
- Price / Seat: 0 (text input)
- Total Seat Available: 100 (text input)
- Flight Type: Choose Flight (dropdown menu)

At the bottom right of the form, there are two buttons: 'Cancel' and 'Create New Ticket'.

Gambar 4. 41 Rancangan Layar Create New Ticket - Admin

Halaman *Create New Ticket* digunakan oleh *superadmin (admin)* untuk membuat tiket baru sesuai dengan jadwal yang sudah disediakan sebelumnya.

#### 4.6.28 Rancangan Layar Transactions – Admin

The screenshot shows a web browser window with the URL `https://garudanih.com/user`. The page title is 'GarudaNih'. The sidebar on the left contains the following menu items: 'GarudaNih', 'Dashboard', 'Our Users', 'List Tickets', and 'Transactions'. The main content area is titled 'Transactions all the time' and contains a table with the following data:

ID	Transaction Code	Passenger	Number Seat	Ticket Code	Number Identify	Booking Date	Action
1	015117BTJBPN009	user example 1	10	DO23TO01AR01025	192312421	January 5, 2023 9:33 PM	⊗
2	015117BTJBPN009	user example 2	60	DO23TO01AR01025	31730518912312	December 5, 2022 9:33 PM	⊗
3	015117BTJBPN009	user example 3	34	DO23TO01AR01025	3173052909812	November 29, 2022 7:35 AM	⊗

At the bottom right of the table, there is a pagination indicator '1-5 of 36' and navigation arrows.

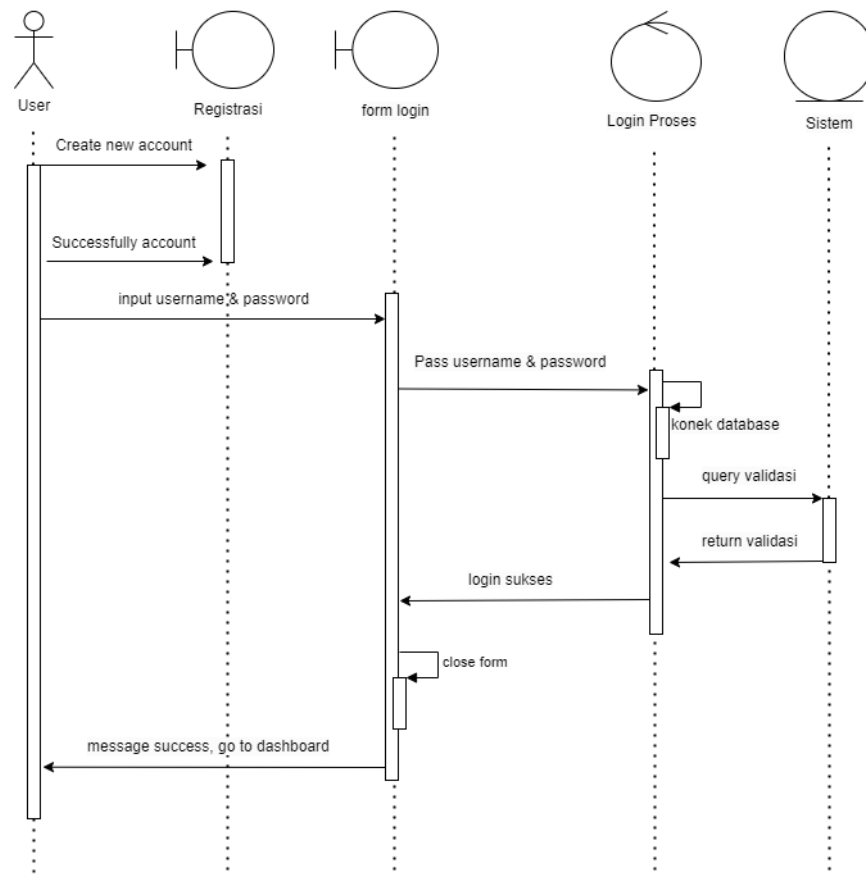
Gambar 4. 42 Rancangan Layar Transactions - Admin

Terakhir adalah halaman *transactions* dimana *superadmin (admin)* dapat melihat semua transaksi yang dilakukan oleh user.

## 4.7 Sequence Diagram

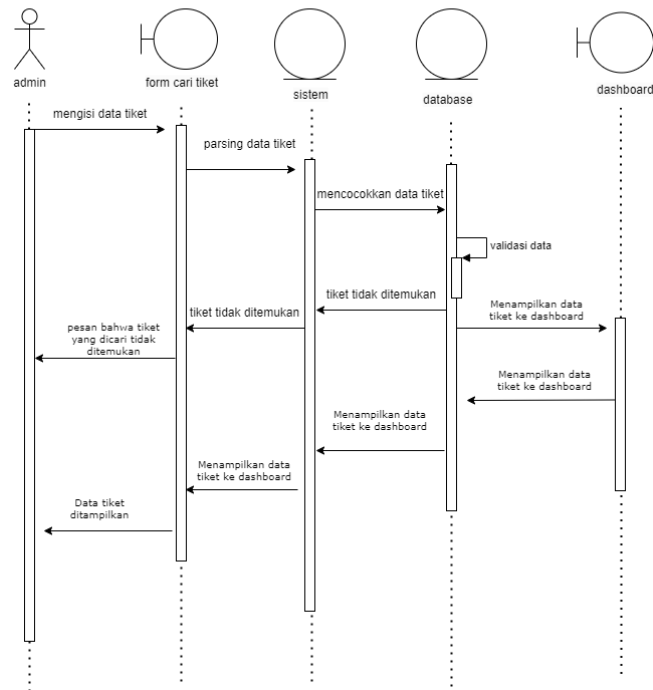
Use case diagram adalah satu dari berbagai jenis diagram UML (Unified Modelling Language) yang menggambarkan hubungan interaksi antara sistem dan aktor. Use Case dapat mendeskripsikan tipe interaksi antara si pengguna sistem dengan sistemnya [9]. Setelah dibuatnya *Use Case Diagram*, *Sequence Diagram* dibutuhkan untuk melihat skenario yang ada pada *Use Case Diagram*.

### Sequence Diagram Login Register GarudaNih



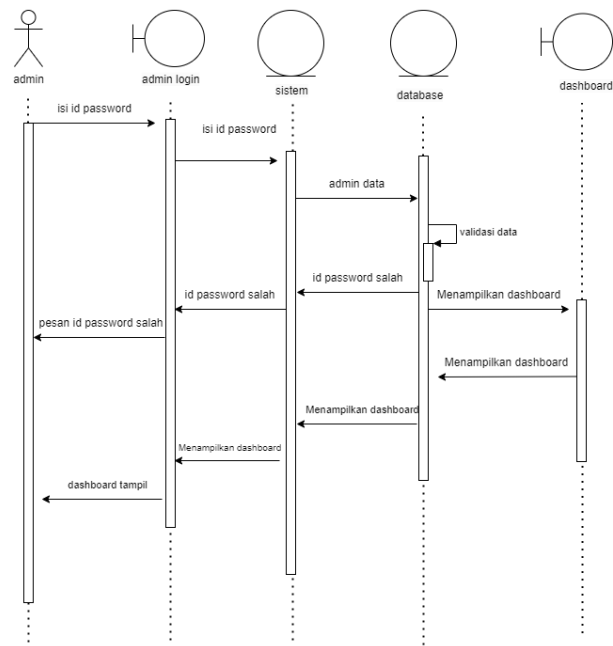
Gambar 4. 43 Sequence Diagram Login & Register - User

### Sequence Diagram Find Ticket User GarudaNih



Gambar 4. 44 Rancangan Layar Sequence Diagram Find Ticket - User

### Sequence Diagram Login Admin GarudaNih



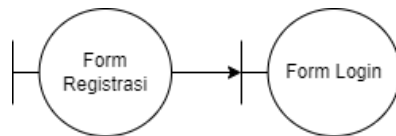
Gambar 4. 45 Sequence Diagram Login - Admin

## 4.8 Class Diagram

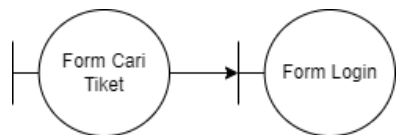
Berikut akan menjelaskan lebih mengenai *Boundary Class*, *Control Class* dan *Entity Class* dengan *Method (Class Diagram)* yang sudah ada sebelumnya.

### a. *Boundary Class*

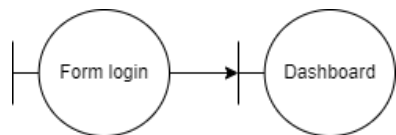
*Boundary Class* berfungsi untuk menangani komunikasi antara internal sistem dengan lingkungan sekitarnya sistem.



Gambar 4. 46 *Boundary Class User*



Gambar 4. 47 *Boundary Class User*



Gambar 4. 48 *Boundary Class Admin*

### b. *Control Class*

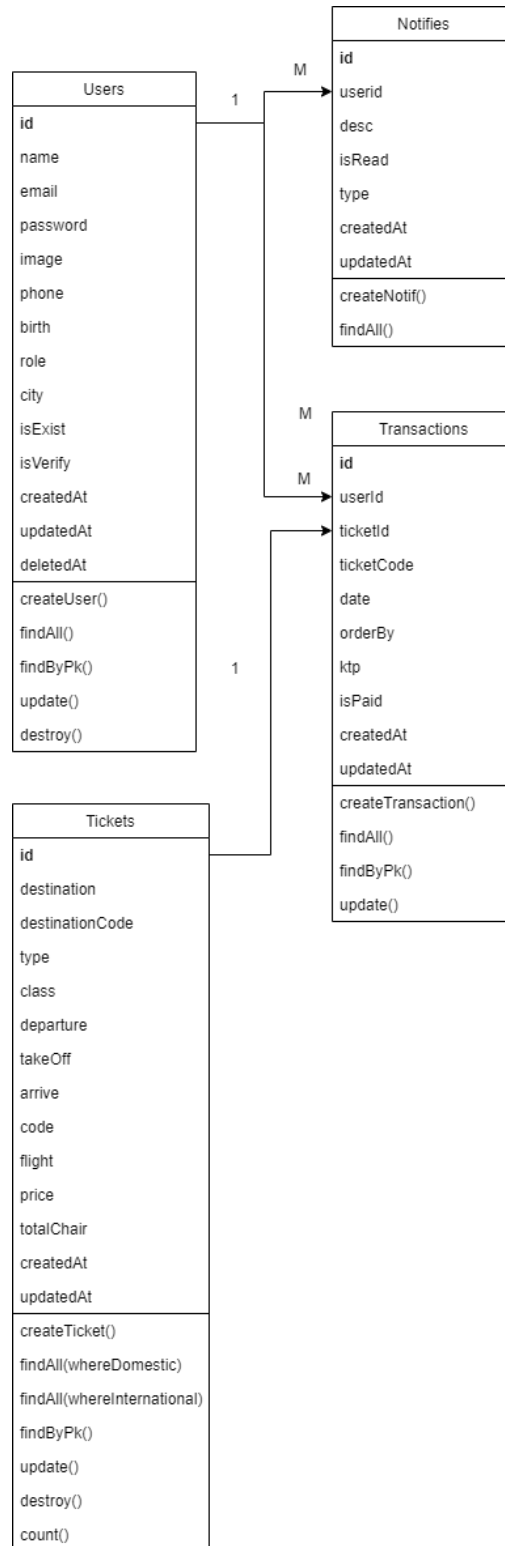
*Control Class* berfungsi untuk memodelkan urutan perilaku dalam satu atau lebih *use case diagram*.



Gambar 4. 49 *Control Class GarudaNih*

c. *Entity Class*

*Entity Class* berikut sama dengan *Class diagram* yang ada dalam rancangan basis data sebelumnya.



Gambar 4. 50 Entity Class GarudaNih

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Setelah melakukan tahap perancangan dan pembuatan sistem kemudian dilanjutkan dengan tahap implementasi, dapat disimpulkan bahwa aplikasi yang telah dibuat, *user requirement* dan *MVP (Minimum Viable Product)* yang ditargetkan sudah terpenuhi semuanya. Adapun *Minimum Viable Product* tersebut adalah:

1. Registrasi (email & password).
2. Login & logout.
3. Implementasi in app notification (notif lonceng).
4. Profile detail (foto, no. Hp, alamat tinggal).
5. Transaksi history (sebagai buyer).

#### **5.2 Saran**

Dengan kesimpulan di atas, maka saran untuk kelanjutan sistem ini adalah sebagai berikut:

1. Diharapkan menambahkan berbagai fitur tambahan yang cukup krusial, diantaranya *payment gateway*, barcode pada tiket.
2. Buat *responsive* berbagai device pada website.
3. Lebih memikirkan lagi dari sisi bisnisnya.
4. Pada tahap *post-production*, melakukan CI/CD, maintenance website.

## BAB VI

### REFLEKSI DIRI

Setelah mengikuti kegiatan magang di PT Lentera Bangsa Benderang (Binar Academy) ini selama kurang lebih 5 bulan, penulis mendapatkan banyak sekali pengalaman, baik *hard skills*, *soft skills* yang pastinya akan berguna bagi penulis di kemudian hari saat memasuki industri. Berikut beberapa hal positif serta pengalaman yang didapat penulis selama mengikuti magang:

- Penulis dapat pengalaman terkait *hard skills* dalam *web development*, khususnya menggunakan ekosistem *JavaScript* [10].
- Penulis dapat pengalaman terkait *soft skills*, khususnya di beberapa bagian seperti kepemimpinan, kerja sama dalam tim, komunikasi antar sesama teman dan *fasiliator*, *public speaking*.
- Dengan diberikannya *challenge* dan *final project*, penulis jadi terbiasa menghadapi banyak *bug* yang muncul dan *problem solving*.
- Dengan mengikuti *final project*, penulis jadi mengerti tanggung jawab seorang *programmer* yang tidak hanya sekedar memperbaiki *bug* dan *problem solving*, tetapi juga harus bisa memimpin anggota timnya.
- Penulis menerapkan dasar-dasar pemrograman serta *algoritma* yang dipelajari selama perkuliahan di Kampus Budi Luhur.
- Penulis juga sangat bersyukur mendapatkan kesempatan untuk mengikuti program ini, karena penulis bisa menambah relasi dan bertemu teman dengan berbagai fokus keahlian, sehingga membuka perspektif baru penulis dalam dunia pemrograman.

## DAFTAR PUSTAKA

- [1] P. S. Nasional, "CYBER EFFECT: PENGARUH INTERNET TERHADAP KEHIDUPAN MANUSIA," in *Prosiding Seminar Nasional Universitas Setia Budi*, 2018.
- [2] pcpiyush1106, "What is PERN Stack?," Sep. 25, 2020. <https://www.geeksforgeeks.org/what-is-pern-stack/> (accessed Jan. 30, 2023).
- [3] Dicoding Intern, "Apa Itu Kotlin? Kenapa Kita Harus Mempelajari Kotlin?," Jul. 22, 2020. <https://www.dicoding.com/blog/apa-itu-kotlin-kenapa-kita-harus-mempelajari-kotlin/> (accessed Jan. 30, 2023).
- [4] B. Frankston, "The javascript ecosystem," *IEEE Consumer Electronics Magazine*, vol. 9, no. 6. 2020. doi: 10.1109/MCE.2020.3009457.
- [5] Binar Academy, "Perbedaan Front End dan Back End Serta Cara Kerjanya." <https://www.binaracademy.com/blog/perbedaan-front-end-dan-back-end> (accessed Jan. 25, 2023).
- [6] "Mengenal Proses Bisnis, Tipe, Contoh, dan Manfaatnya untuk Perusahaan", Accessed: Jan. 25, 2023. [Online]. Available: <https://www.jurnal.id/id/blog/proses-bisnis-adalah/#:~:text=Proses%20bisnis%20adalah%20sekumpulan%20aktivitas,output%20yang%20bernilai%20bagi%20pelanggan.&text=Kemudian%2C%20proses%20bisnis%20adalah%20sekumpulan,untuk%20menghasilkan%20produk%20dan%20jasa.>
- [7] Ardiansyah, "ATURAN BISNIS DALAM KEBUTUHAN FUNGSIONAL PERANGKAT LUNAK," May 27, 2015. [http://ardiansyah.tif.uad.ac.id/aturan-bisnis-dalam-kebutuhan-fungsional-perangkat-lunak-2/#:~:text=Menurut%20Business%20Rules%20Group%20\(1993,mengendalikan%20atau%20mempengaruhi%20perilaku%20bisnis.](http://ardiansyah.tif.uad.ac.id/aturan-bisnis-dalam-kebutuhan-fungsional-perangkat-lunak-2/#:~:text=Menurut%20Business%20Rules%20Group%20(1993,mengendalikan%20atau%20mempengaruhi%20perilaku%20bisnis.) (accessed Jan. 25, 2023).
- [8] Dicoding Intern, "Contoh Use Case Diagram Lengkap dengan Penjelasannya," May 19, 2021. <https://www.dicoding.com/blog/contoh-use-case-diagram/> (accessed Jan. 25, 2023).
- [9] Sekar, "Mengenal Komponen-Komponen & Contoh Sequence Diagram," Oct. 19, 2022. Mengenal Komponen-Komponen & Contoh Sequence Diagram (accessed Jan. 25, 2023).
- [10] M. Doernhoefer, "JavaScript," *ACM SIGSOFT Software Engineering Notes*, vol. 31, no. 4, 2006, doi: 10.1145/1142958.1142972.

### Lampiran 1. Log Activity

Minggu / Tgl	Kegiatan	Hasil
1 / (01 – 05 Agustus 2022)	<ul style="list-style-type: none"> <li>Berkenalan dan belajar HTML.</li> <li>Self learning, mengerjakan task 1.</li> <li>Belajar CSS, live coding imitasi tampilan suatu web.</li> <li>Self learning, mengerti grid dalam css dengan memainkan game gridgarden.</li> <li>Mengerjakan task.</li> </ul>	Minggu ini mempelajari HTML dan CSS. Perkenalan diri di kelas dan mengenal masing-masing peserta lainnya. Cukup mengerti bagaimana HTML dan CSS ketika diimplementasikan.
2 / (08 – 12 Agustus 2022)	<ul style="list-style-type: none"> <li><i>Introduction</i> challenge, pemberian challenge dan bagaimana cara mengerjakannya.</li> <li>Self learning CSS dan mengerjakan challenge.</li> <li>Track progres challenge oleh fasilitator.</li> <li>Self learning CSS dan mengerjakan challenge.</li> <li>Review challenge dan feedback challenge</li> </ul>	Minggu ini diperkenalkan dengan sistem challenge. Bagaimana para peserta diberikan petunjuk pengerjaannya. Mengerjakan challenge menggunakan framework CSS, yaitu Bootstrap.
3 / (15 – 19 Agustus 2022)	<ul style="list-style-type: none"> <li>Belajar Git.</li> <li>Self learning Git.</li> <li>Melanjutkan belajar Git.</li> <li>Self learning materi Git, Branch, Merge, Commit.</li> <li>Pemberian materi responsive web dan review task hari kemarin.</li> </ul>	Minggu ini mempelajari Git beserta kegunaannya. Mengetahui apa itu Branch, Merge, Commit. Mengetahui GitHub, aplikasi Git versi web. Lalu di hari terakhir diberi materi mengenai responsive web.
4 / (22 – 26 Agustus 2022)	<ul style="list-style-type: none"> <li>Belajar membuat responsive web design dengan bootstrap.</li> <li>Self learning bootstrap.</li> <li>Mulai mengerjakan challenge.</li> <li>Self learning dan mengerjakan challenge.</li> <li>Mengerjakan challenge.</li> </ul>	Belajar membuat responsive web design dengan Bootstrap. Mulai mengimplementasikan materi tentang Git dengan cara membuat repository dan unggah challenge di dalamnya.
5 / (29 Agustus – 02)	<ul style="list-style-type: none"> <li>Belajar Javascript.</li> <li>Self learning Javascript.</li> <li>Javascript lanjutan.</li> </ul>	Minggu ini belajar mengenai Javascript. Dimulai dari dasar seperti

September 2022)	<ul style="list-style-type: none"> <li>• Self learning Javascript lanjutan.</li> <li>• Logika pemrograman dengan Javascript.</li> </ul>	pengenalan tipe data sampai Javascript lanjutan seperti pengulangan <i>for loop</i> .
6 / (05 – 09 September 2022)	<ul style="list-style-type: none"> <li>• Melanjutkan Javascript lanjutan</li> <li>• Self learning Javascript lanjutan.</li> <li>• Task Javascript lanjutan.</li> <li>• Self learning dan mengerjakan Task.</li> <li>• Feedback Javascript lanjutan dengan task</li> </ul>	Minggu ini melanjutkan pembelajaran minggu lalu, yaitu Javascript lanjutan. Seperti kondisi, perulangan, dan array.
7 / (12 – 16 September 2022)	<ul style="list-style-type: none"> <li>• OOP pada Javascript.</li> <li>• Self learning OOP Javascript.</li> <li>• Task OOP Javascript.</li> <li>• Self learning dan mengerjakan task.</li> <li>• Instalasi NodeJs</li> </ul>	Minggu ini melanjutkan Javascript, yaitu <i>Object Oriented Programming / OOP</i> . Di hari terakhir melakukan penginstalasian NodeJs.
8 / (19 – 23 September 2022)	<ul style="list-style-type: none"> <li>• HTTP server dengan NodeJs.</li> <li>• Self learning NodeJs.</li> <li>• Melanjutkan HTTP Server dengan NodeJs.</li> <li>• Self learning dan mengerjakan challenge.</li> <li>• Menyelesaikan challenge.</li> </ul>	Minggu ini belajar mengenai HTTP server dengan NodeJs. Challenge diberikan dan dikerjakan dengan cara berkolaborasi dengan teman sekelas.
9 / (26 – 30 September 2022)	<ul style="list-style-type: none"> <li>• Belajar ExpressJs.</li> <li>• Self learning Express Js.</li> <li>• Database <i>introduction</i>, SQL dan PostgreSQL.</li> <li>• Self learning database.</li> <li>• Belajar Restful API dan ORM.</li> </ul>	Minggu ini belajar menggunakan ExpressJs, mendesain RestfulAPI dan database dengan sql.
10 / (03 – 07 Oktober 2022)	<ul style="list-style-type: none"> <li>• Melanjutkan ORM.</li> <li>• Self learning ORM.</li> <li>• Membahas challenge.</li> <li>• Self learning dan pengerjaan challenge.</li> <li>• Menyelesaikan challenge</li> </ul>	Minggu ini sedikit melanjutkan materi ORM, lalu dilanjutkan dengan pengerjaan challenge secara berkelompok.
11 / (10 – 14 Oktober 2022)	<ul style="list-style-type: none"> <li>• Belajar design pattern.</li> <li>• Self learning design pattern.</li> <li>• Asynchronous Javascript.</li> <li>• Self learning asynchronous</li> <li>• Authentication &amp; Authorization</li> </ul>	Minggu ini belajar Design pattern seperti mvc. Asynchronous pada javascript, Autentikasi dan autorisasi

<b>12 / (17 – 21 Oktober 2022)</b>	<ul style="list-style-type: none"> <li>• Open API.</li> <li>• Self learning.</li> <li>• Pemberian dan mengerjakan challenge.</li> <li>• Self learning dan proses mengerjakan challenge.</li> <li>• Membahas dan menyelesaikan challenge.</li> </ul>	Minggu ini Belajar open api dan menyelesaikan challenge.
<b>13 / (24 – 28 Oktober 2022)</b>	<ul style="list-style-type: none"> <li>• ReactJs.</li> <li>• Self learning ReactJs.</li> <li>• React Router.</li> <li>• Self learning React Router.</li> <li>• Oauth google.</li> </ul>	Minggu ini mulai belajar ke tahap advance, yaitu kerangka kerja ReactJs. Ditambahkan pula dengan Oauth google.
<b>14 / (31 Oktober – 04 November 2022)</b>	<ul style="list-style-type: none"> <li>• React Redux.</li> <li>• Self learning React Redux.</li> <li>• Mengerjakan challenge.</li> <li>• Self learning dan melanjutkan challenge.</li> <li>• Membahas challenge</li> </ul>	Minggu ini belajar menggunakan redux pada react, menyelesaikan challenge.
<b>15 / (07 – 11 November 2022)</b>	<ul style="list-style-type: none"> <li>• Belajar cara menggunakan web socket.</li> <li>• Self learning web socket.</li> <li>• Belajar menggunakan NextJs.</li> <li>• Self learning NextJs.</li> <li>• Improvisasi syntax menggunakan eslint.</li> </ul>	Minggu ini belajar banyak sekali hal, mulai dari implementasi real time chat menggunakan web socket, next js, dan eslint.
<b>16 / (14 – 18 November 2022)</b>	<ul style="list-style-type: none"> <li>• Testing (tdd).</li> <li>• Self learning testing.</li> <li>• Deployment menggunakan railway.</li> <li>• Melanjutkan self learning testing.</li> <li>• Mengerjakan challenge.</li> </ul>	Minggu ini berfokus pada testing, deployment dan pengerjaan challenge.
<b>17 / (21 – 25 November 2022)</b>	<ul style="list-style-type: none"> <li>• Melakukan diskusi awal final project.</li> <li>• Self learning persiapan final project.</li> <li>• Melanjutkan diskusi final project, menentukan role.</li> <li>• Self learning.</li> <li>• Melakukan perancangan database.</li> </ul>	Minggu ini diperkenalkan final project. Menentukan role untuk masing-masing peserta kelompok.
<b>18 / (28 November –</b>	<ul style="list-style-type: none"> <li>• Melanjutkan diskusi, menentukan alur aplikasi.</li> </ul>	Minggu ini melanjutkan final project, khususnya

02 Desember 2022)	<ul style="list-style-type: none"> <li>• Self learning, melanjutkan hasil diskusi kemarin.</li> <li>• Melanjutkan menentukan alur program.</li> <li>• Self learning, melanjutkan pembuatan alur program.</li> <li>• Mendesain endpoint sesuai yang disepakati.</li> </ul>	di bagian perancangan alur aplikasi dan pembuatan dasar API.
19 / (05 – 09 Desember 2022)	<ul style="list-style-type: none"> <li>• Diskusi dengan kelompok, melakukan tracking progres pembuatan aplikasi.</li> <li>• Melanjutkan mendesain API.</li> <li>• Melakukan diskusi dengan kelompok terkait bug yang dialami.</li> <li>• Melanjutkan mendesain API.</li> <li>• Diskusi dengan teman sekelompok terkait progres dan masalah yang dialami saat mendesain API.</li> </ul>	Minggu ini digunakan untuk mendesain api dan berdiskusi terkait masalah atau error yang dialami saat mendesain api.
20 / (12 – 16 Desember 2022)	<ul style="list-style-type: none"> <li>• Melakukan tracking progres dan berdiskusi error.</li> <li>• Melanjutkan membuat API.</li> <li>• Melanjutkan diskusi terkait MVP final project.</li> <li>• Menyelesaikan fitur agar sesuai dengan MVP.</li> <li>• Membuat testing.</li> </ul>	Menyelesaikan api, menyelesaikan beberapa fitur agar sesuai dengan mvp. Mencoba melakukan testing.
21 / (19 – 23 Desember 2022)	<ul style="list-style-type: none"> <li>• Berdiskusi dan melakukan tracking progress.</li> <li>• Melanjutkan membuat testing untuk API.</li> <li>• Berdiskusi mengenai masalah yang dialami saat membuat testing.</li> <li>• Melanjutkan membuat testing.</li> <li>• Berdiskusi untuk menambahkan additional feature dalam final project</li> </ul>	Minggu ini mengerjakan testing dan mendebug beberapa error yang ditemukan saat mengerjakan final project.
22 / (26 – 30 Desember 2022)	<ul style="list-style-type: none"> <li>• Mengerjakan additional feature</li> <li>• Self learning, mengerjakan additional feature.</li> <li>• Testing manual fungsionalitas website.</li> <li>• Tahap akhir (penyelesaian) final project.</li> </ul>	Mengimplementasikan beberapa fitur tambahan dan menyelesaikan final project.

## Lampiran 2. Student Report, Sertifikat Mitra dan Surat Penilaian Instansi

### Student Report Halaman 1



### STUDENT REPORT FULLSTACK WEB DEVELOPMENT CLASS

This temporary report is belong to student with name :

**MUHAMMAD ALI AKBAR**

( aliakbar1850@gmail.com )

Who passed **Chapter 11** in Fullstack Web Development Class with the following result :

Chapters Overall Score	
<b>Chapter 0 : Memahami Pengantar Dunia Pemrograman</b> <i>1. Programming Introduction</i>	<b>84</b>
<b>Chapter 1 : Menguasai kemampuan dasar untuk membuat halaman web</b> <i>1. FSW (Pengantar) 2.HTML 3. CSS 4. CSS Framework</i>	<b>80</b>
<b>Chapter 2 : Mampu menjalankan code dan melakukan kolaborasi dengan developer lain dalam proses system development</b> <i>1. Terminal dan IDE 2. GIT 3. Web Layout 4. Responsive Design 5. Advance CSS Framework</i>	<b>87</b>
<b>Chapter 3 : Memahami logika-logika dasar Javascript</b> <i>1. Data Structure 2. Operator &amp; Expression 3. Basic Javascript Algorithm</i>	<b>90</b>
<b>Chapter 4 : Mampu menerapkan OOP dan DOM dalam pengembangan web</b> <i>1. OOP in Javascript 2. DOM 3. Node.JS 4. HTTP Server</i>	<b>90</b>
<b>Chapter 5 : Melakukan perancangan database</b> <i>1. Express JS 2. Restful API 3. Database</i>	<b>85</b>

## Student Report Halaman 2



**Kampus  
Merdeka**  
INDONESIA JAYA

**Binar Academy**

PT. Lentera Bangsa Benderang  
The Breeze BSD, Blok L30  
Jl. Grand Boulevard, BSD Green Office Park  
BSD City, Tangerang 15345

4. SQL 5. ORM (Object Relational Mapping)	
<b>Chapter 6 : Merancang arsitektur dan dokumentasi API</b> 1. Design Pattern 2. Asynchronous 3. Authentication 4. Open API	<b>87</b>
<b>Chapter 7 : Mampu membuat tampilan web dengan menggunakan React JS</b> 1. React JS 2. React Router & Back-end Integration 3. OAuth 4. Redux	<b>80</b>
<b>Chapter 8 : Melakukan unit testing dan deployment</b> 1. Web Socket 2. SSR (Server Side Rendering) pada React / With NEXTJS 3. ESLINT 4. Unit testing & TDD 5. Deployment & CI/CD	<b>90</b>
<b>Chapter 9-11 : Final Project</b> Melakukan kolaborasi tim pengembangan aplikasi dengan iterasi proses testing dan evaluasi melalui repository Github	<b>90</b>

**Chapters Average Score**

**88**

Sincerely,  
Binar Academy

## Student Report Halaman 3



**Kampus  
Merdeka**  
INDONESIA JAYA

**Binar Academy**

PT. Lentera Bangsa Benderang  
The Breeze BSD, Blok L30  
Jl. Grand Boulevard, BSD Green Office Park  
BSD City, Tangerang 15345

### Additional Information

Logbook	
<b>Total Approved Logbook</b>	<b>100.00</b>
Approved Logbook Chapter 11	%

Notes From Facilitator
<b>Chapter 1 :</b> <i>Di Challenge ini, siswa dapat mengimplementasikan materi dengan baik. Tampilan website sudah sesuai desain yang ada. Overall sudah Oke!</i>
<b>Chapter 2 :</b> <i>Pemahaman siswa mengenai Git sudah bagus sekali, siswa mampu menerapkan responsive desain sesuai dengan tampilan di figma. Excellent!</i>
<b>Chapter 3 :</b> <i>Siswa mampu memahami materi dengan baik dan mengerjakan challenge dengan maksimal. Good!</i>
<b>Chapter 4 :</b> <i>Pada chapter ini siswa mampu mengimplementasikan DOM dan menggunakan HTTP Server dengan baik dan benar. Siswa aktif dikelas dan membantu teman yang kesulitan. Excellent!</i>
<b>Chapter 5 :</b> <i>Pada chapter ini website sudah bagus dan semua fungsi berjalan dengan baik. Implementasi Rest API menggunakan view engine sudah bagus. Lanjutkan!</i>
<b>Chapter 6 :</b> <i>Siswa mampu menerapkan materi di chapter in dengan baik dan mengerjakan challenge dengan maksimal. Good Job!</i>
<b>Chapter 7 :</b> <i>Siswa selalu mengikuti pelajaran dengan baik, meski kurang aktif dan komunikatif saat di kelas, namun mampu menyelesaikan challenge dengan baik.</i>
<b>Chapter 8 :</b> <i>Siswa mampu melakukan unit testing dan deployment dengan baik. Good Job!</i>

## Student Report Halaman 4



**Kampus  
Merdeka**  
INDONESIA JAYA

**Binar Academy**

PT. Lentera Bangsa Benderang  
The Breeze BSD, Blok L30  
Jl. Grand Boulevard, BSD Green Office Park  
BSD City, Tangerang 15345

Chapters Overall Soft Skill Score	
<b>Learning Drive</b> <i>Motivasi yang menjadi landasan dan dorongan untuk terus belajar dan mengembangkan diri</i>	<b>90</b>
<b>Growth Mindset</b> <i>Persepsi/kepercayaan akan kemampuan diri sendiri bahwa kemampuan dirinya masih dapat terus dikembangkan dengan usaha daripada menilai dirinya sudah tidak dapat diubah lagi</i>	<b>90</b>
<b>Grit</b> <i>Motivasi yang menjadi landasan dan dorongan untuk terus belajar dan mengembangkan diri</i>	<b>90</b>
<b>Self-regulated learning</b> <i>Persepsi/kepercayaan akan kemampuan diri sendiri bahwa kemampuan dirinya masih dapat terus dikembangkan dengan usaha daripada menilai dirinya sudah tidak dapat diubah lagi</i>	<b>90</b>
<b>Critical Thinking</b> <i>Kemampuan mengidentifikasi tingkat pemahaman, mengevaluasi berbagai informasi hingga mampu membuat pola pemahaman/kesimpulan, dan menyampaikan pemahamannya dengan efektif</i>	<b>90</b>
<b>Problem Solving</b> <i>Kemampuan memahami masalah yang diberikan hingga mampu menentukan jawaban/solusi paling sesuai pada situasi tersebut</i>	<b>90</b>
<b>Creative Thinking</b> <i>Kemampuan menemukan peluang inovasi hingga mampu menginisiasi ide/inisiatif baru yang relevan</i>	<b>90</b>
<b>Learning Agility</b> <i>Kemampuan menemukan peluang inovasi hingga mampu menginisiasi ide/inisiatif baru yang relevan</i>	<b>90</b>
<b>Communication skill</b> <i>Kemampuan menyampaikan tujuan/ide/pemikiran secara efektif dan meyakinkan pihak lain</i>	<b>85</b>

## Student Report Halaman 5



**Kampus  
Merdeka**  
INDONESIA JAYA

**Binar Academy**

PT. Lentera Bangsa Benderang  
The Breeze BSD, Blok L30  
Jl. Grand Boulevard, BSD Green Office Park  
BSD City, Tangerang 15345

<b>Management skill</b> <i>Kemampuan mengelola proses kerja untuk menghasilkan target yang diharapkan</i>	<b>95</b>
<b>Leadership skill</b> <i>Kemampuan memotivasi, mengarahkan, dan mengembangkan diri sendiri maupun orang lain untuk dapat mencapai tujuan bersama</i>	<b>95</b>

Soft Skill Score
<b>91</b>

Sincerely,  
Binar Academy

## Sertifikat Mitra



## Penilaian Instansi Halaman 1



Jakarta, 26 September 2022

Nomor : H/UBL/KKP/220082/09/22  
Perihal : Permohonan Penilaian Riset KKP

Kepada Yth:  
**Bapak/ Ibu Pembimbing Riset KKP**  
**PT LENTERA BANGSA BENDERANG**  
di  
Tempat

Dengan hormat,  
Sesuai dengan kegiatan akademik yang dilaksanakan di Fakultas Teknologi Informasi Universitas Budi Luhur Jakarta, kepada mahasiswa yang telah memenuhi persyaratan akademik wajib melakukan kegiatan riset untuk keperluan Kuliah Kerja Praktek (KKP). sehubungan dengan hal tersebut, dengan ini kami mohon kesediaan Bapak/ Ibu Pembimbing untuk memberikan penilaian kepada mahasiswa tersebut pada form Penilaian terlampir.

Atas kesediaan Bapak/Ibu untuk memberikan pengarahan, bimbingan serta penilaian kepada para mahasiswa kami selama melakukan riset KKP di instansi/ perusahaan Bapak/Ibu, kami haturkan terima kasih.

Hormat kami,  
Dekan Fakultas Teknologi Informasi  
Universitas Budi Luhur

**Dr. Deni Mahdiana, S.Kom., M.M., M.Kom.**

## Penilaian Instansi Halaman 2

### FORM PENILAIAN TINGKAT KEPUASAN PENGGUNA

NIM : 1911502217

Nama : Muhammad Ali Akbar

No.	Jenis Kemampuan	Nilai (0-100)
1	Etika	88
2	Keahlian pada bidang ilmu (kompetensi utama)	90
3	Kemampuan berbahasa asing	86
4	Penggunaan teknologi informasi	90
5	Kemampuan berkomunikasi	85
6	Kerjasama tim	85
7	Pengembangan diri	89
8	Disiplin	87

Tangerang Selatan, 26 Januari 2023

Pembimbing Lapangan



PT. LENTERA BANGSA  
BENDERANG

Bagus Prakoso Gunawan

### Lampiran 3. Lampiran Logbook Kegiatan Harian

Fakultas Teknologi Informasi – Universitas Budi Luhur

Nama Instansi KKP	PT. Lentera Bangsa Benderang (Binar Academy)
Alamat Instansi	Binar Academy bertempat di The Breeze (Batik 3 - L.30, Jl. BSD Grand Boulevard, BSD City, Kec. Cisauk, Kabupaten Tangerang, Banten 15345.
Dosen Pembimbing	Ikhsan Rahdiana S.Kom., M.Kom.

No.	NIM	Nama
1	1911502217	Muhammad Ali Akbar

No.	Tanggal	Materi yang dikonsultasikan	Paraf Dosen
1	13 Des/2022	Membahas Cara Pengisian Trap - trap kab	
2	09 Jan/2023	Bimbingan bab 1-3 & Cara Pengisian bab 4	
3	15 Jan/2023	Struktur tampilan aplikasi	
4	17 Jan/2023	Membahas Aliran bisnis aplikasi	
5	19 Jan/2023	Bimbingan revisi bab 1-4	
6	21 Jan/2023	Revisi wireframe, menambahkan android & admin	
7	23 Jan/2023	Revisi struktur tampilan, spesifikasi basis data	
8	25 Jan/2023	Bimbingan bab 1-6, mengganti judul per-tan	
9			
10			

Mahasiswa di atas telah melakukan bimbingan dengan jumlah materi yang telah mencukupi untuk diseminarkan.

Jakarta, 25 Januari 2023  
Dosen Pembimbing

Ikhsan Rahdiana S.Kom., M.Kom.

## Lampiran 4. Laporan Persetujuan Kuliah Kerja Praktek



FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BUDI LUHUR

---

### PERSETUJUAN LAPORAN KULIAH KERJA PRAKTEK

Program Studi : Teknik Informatika  
Bidang Peminatan : Programming Expert  
Jenjang Studi : Strata 1  
Judul : Studi Independen Pembuatan Aplikasi Maskapai  
Penerbangan Berbasis Website dan Android Di PT Lentera  
Bangsa Banderang

NIM	Nama
1. 1911502217	Muhammad Ali Akbar

Disetujui untuk dipertahankan dalam seminar KKP periode semester Gasal  
tahun ajaran 2022/2023

Jakarta, 25 Januari 2023

Dosen Pembimbing

Ikhsan Rahdiana S.Kom., M.Kom.

# Lampiran 5. Materi Matkul MBKM

SILVER				GOLD				PLATINUM			
Chapter 0	Chapter 1	Chapter 2	Chapter 3	Chapter 4	Chapter 5	Chapter 6	Chapter 7	Chapter 8	Chapter 9	Chapter 10	Chapter 11
<p><b>Mengikuti Program Studi Pengembangan</b></p> <p>Programming Introduction</p> <ul style="list-style-type: none"> <li>- Segitiga Abstrak</li> <li>- Konsep Produk</li> <li>- Non Stack</li> <li>- Bahasa Pemrograman</li> </ul>	<p><b>Menguasai kemampuan dasar untuk membuat halaman web</b></p> <p>FSW (Frontend)</p> <ul style="list-style-type: none"> <li>- Pengetahuan tentang web developer</li> <li>- FSW developer job scope</li> <li>- HTML dan CSS</li> <li>- Tools introduction</li> </ul>	<p><b>Memahami logika logika dasar JavaScript</b></p> <p>Data Structure</p> <ul style="list-style-type: none"> <li>- Variable</li> <li>- Array</li> </ul>	<p><b>Mampu menjelaskan code dalam proses system</b></p> <p>Terminal dan IDE</p> <ul style="list-style-type: none"> <li>- Terminal dan IDE</li> </ul>	<p><b>Mampu menerapkan OOP dan DOM dalam pengembangan web</b></p> <p>OOP in Javascript</p> <ul style="list-style-type: none"> <li>- OOP introduction</li> <li>- Polymorphism</li> <li>- Encapsulation</li> <li>- Abstraction</li> </ul>	<p><b>Melakukan perencanaan database</b></p> <p>Express JS</p> <ul style="list-style-type: none"> <li>- Express JS introduction and installation</li> <li>- Middleware (authentication, authorization)</li> <li>- View Engine</li> </ul>	<p><b>Mencoreng arsitektur dan dokumentasi API</b></p> <p>Design Pattern</p> <ul style="list-style-type: none"> <li>- Design pattern introduction</li> <li>- Microservice vs Monolith</li> <li>- API Gateway</li> <li>- Service Repository Pattern</li> </ul>	<p><b>Mampu membuat tampilan web dengan menggunakan React JS</b></p> <p>React JS (SPA)</p> <ul style="list-style-type: none"> <li>- React JS (SPA)</li> <li>- Component, State dan Properti</li> <li>- Styling di ReactJS</li> <li>- Menggunakan UI framework</li> </ul>	<p><b>Melakukan unit testing dan deployment</b></p> <p>Web Socket</p> <ul style="list-style-type: none"> <li>- Web socket introduction</li> <li>- Socket IO</li> </ul>	<p>Self Learning, Live Forum Discussion, Individual Project</p>	<p>Self Learning, Live Forum Discussion, Collaboration Project</p>	<p>Self Learning, Live Forum Discussion, Collaboration Project</p>
<p><b>HTML</b></p> <ul style="list-style-type: none"> <li>- Introduction HTML dan web design</li> <li>- Meta data formatting</li> <li>- HTML dan CSS</li> <li>- Docs and structure</li> <li>- Debugging HTML</li> </ul>	<p><b>Operasi &amp; Expression</b></p> <ul style="list-style-type: none"> <li>- Operators</li> <li>- Logical operators</li> </ul>	<p><b>Git</b></p> <ul style="list-style-type: none"> <li>- Mengetahui GIT</li> <li>- Pengunaan GIT</li> <li>- Remote repository</li> <li>- Cloning</li> </ul>	<p><b>Web Layout</b></p> <ul style="list-style-type: none"> <li>- Memahami Web Layout</li> <li>- Teknik spacing</li> <li>- Mempraktikkan CSS</li> </ul>	<p><b>DOM</b></p> <ul style="list-style-type: none"> <li>- DOM introduction</li> <li>- Selector</li> <li>- DOM manipulation</li> </ul>	<p><b>Database</b></p> <ul style="list-style-type: none"> <li>- Mengenal database</li> <li>- DBMS (Database Management System)</li> <li>- DDL (Data Definition Language)</li> <li>- DML (Data Manipulation Language)</li> <li>- NoSQL (MongoDB)</li> <li>- ERD (Entity Relationship Diagram)</li> </ul>	<p><b>Authentikasi</b></p> <ul style="list-style-type: none"> <li>- Authentication</li> <li>- JWT strategy</li> <li>- Encryption</li> </ul>	<p><b>OAuth</b></p> <ul style="list-style-type: none"> <li>- OAuth</li> <li>- Google &amp; Facebook oauth</li> </ul>	<p><b>Media handling</b></p> <ul style="list-style-type: none"> <li>- Video</li> <li>- Image</li> <li>- File</li> </ul>	<p>Self Learning, Live Forum Discussion, Individual Project</p>	<p>Self Learning, Live Forum Discussion, Collaboration Project</p>	<p>Self Learning, Live Forum Discussion, Collaboration Project</p>
<p><b>CSS</b></p> <ul style="list-style-type: none"> <li>- CSS introduction</li> <li>- CSS syntax</li> <li>- CSS Values and Units</li> <li>- The box models</li> <li>- Debugging CSS</li> </ul>	<p><b>Basic Javascript Algorithm</b></p> <ul style="list-style-type: none"> <li>- Algorithm, Pseudocode</li> <li>- Loop</li> <li>- Function</li> </ul>	<p><b>Responsive Design</b></p> <ul style="list-style-type: none"> <li>- Mengetahui Responsive Design</li> <li>- Media query</li> <li>- Teknik Responsive Design</li> </ul>	<p><b>NodeJS</b></p> <ul style="list-style-type: none"> <li>- Runtime environment</li> <li>- Module (Core Module, Third Party)</li> <li>- Package manager</li> <li>- Node JS vs Browser</li> <li>- Read and write file</li> </ul>	<p><b>ORM (Object Relational Mapping)</b></p> <ul style="list-style-type: none"> <li>- Mengenal ORM dan instalasi</li> <li>- Definis Model</li> <li>- Implementasi CRUD</li> </ul>	<p><b>Open API</b></p> <ul style="list-style-type: none"> <li>- Mengenal Open API</li> <li>- Struktur dan Klien Dokumentasi API</li> <li>- Swagger Tool</li> <li>- Swagger Swagger</li> <li>- Integrasi Dokumentasi API</li> </ul>	<p><b>Redux</b></p> <ul style="list-style-type: none"> <li>- Redux Management</li> <li>- Implementasi React + Redux</li> <li>- Redux thnik (Middleware)</li> </ul>	<p><b>ESLINT</b></p> <ul style="list-style-type: none"> <li>- Setup ESLint</li> </ul>	<p><b>Unit testing &amp; TDD</b></p> <ul style="list-style-type: none"> <li>- Unit Testing</li> <li>- TDD (Test Driven Development)</li> <li>- Unit Testing</li> <li>- Integration Testing</li> </ul>	<p>Self Learning, Live Forum Discussion, Individual Project</p>	<p>Self Learning, Live Forum Discussion, Collaboration Project</p>	<p>Self Learning, Live Forum Discussion, Collaboration Project</p>
<p><b>CSS Framework</b></p> <ul style="list-style-type: none"> <li>- Foundation</li> <li>- Semantic UI</li> </ul>									<p>Self Learning, Live Forum Discussion, Individual Project</p>	<p>Self Learning, Live Forum Discussion, Collaboration Project</p>	<p>Self Learning, Live Forum Discussion, Collaboration Project</p>

## Lampiran 6. Foto Kegiatan MBKM

